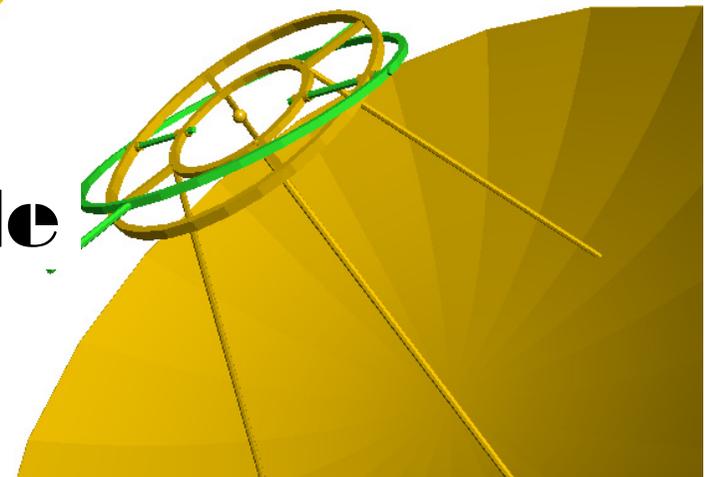


Guide des Choses, Machins, Trucs et autres sortes de Bidules à fabriquer



Avec le concours de



Bienvenue

BlocksCAD est un logiciel qui permet de faire des dessins en volumes. Ce logiciel permet de manipuler diverses formes de bases (cylindres, parallélépipèdes, cônes, sphères) pour faire de nombreux objets en 3D pour utiliser un mot à la mode. En plus, BlocksCAD permet de lancer les objets en impression 3D.

Ce livre est un voyage d'exploration pour s'initier à la construction d'objets avec les fonctions de base mais surtout en explorant toutes les autres fonctions disponibles c'est à dire :

- la capacité de programmation avec des fonctions informatiques telles que les boucles et les tests,
- la présence de nombreuses fonctions mathématiques.

Ce livre va devenir un prétexte à parler de mathématiques, de géométrie et de technologie en essayant de rester compréhensible et simple même pour les jeunes qui n'ont pas encore appris ces domaines en classe mais aussi pour les plus grands qui ont des souvenirs assez anciens sur ces notions.

Ce livre s'adresse aussi à ceux qui aiment bricoler le dimanche et passer un peu de temps tout en essayant de comprendre ce qu'ils font.

Ce livre vous est présenté par Cyril Vignet, ingénieur, assisté de Quitrie (CM2) et de Xavière (CM2). La licence du livre est Creative Commons Attribution 3.0 France. Pour les blocks proposés, aucun besoin de respecter l'attribution. Janvier 2017.

Premiers contacts

BLOCKSCAD est un logiciel disponible à partir d'un navigateur qui permet de modéliser des volumes à partir de fonctions géométriques et informatiques. Nous utilisons l'adresse des concepteurs du logiciel <https://www.blockscad3d.com/editor/#> Il existe aussi d'autres sites, en particulier en français, mais les quelques mots en anglais ne posent guère de soucis.

Le bureau de travail

The screenshot shows the BlocksCAD web editor interface. On the left is a vertical menu with categories like 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. At the top is a horizontal menu with Project, Options, Help, and Examples. The main workspace is a 3D grid. On the right is a preview window showing a rendered object. Three callout boxes provide explanations: an orange box at the top points to the horizontal menu, a light blue box in the center points to the main workspace, and a pink box on the right points to the preview window and its controls.

Un menu horizontal en haut de l'écran pour gérer son profil et sélectionner ses différents projets en cours.

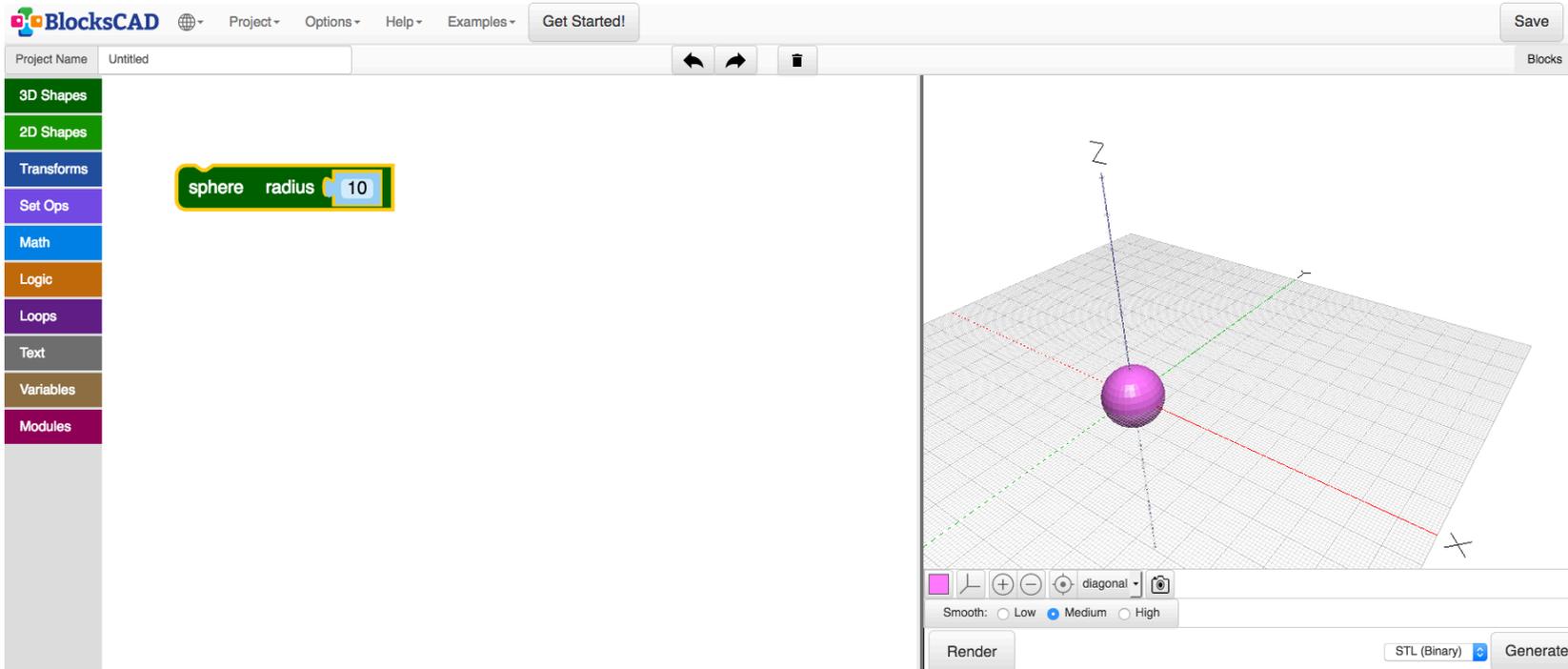
Un espace blanc au milieu pour placer les fonctions géométriques et informatiques décrivant sous forme de formule un objet.

Un menu vertical à gauche permettant un accès facile aux fonctions géométriques et informatiques.

Une fenêtre à droite visualisant l'objet décrit par les formules, une fois que l'on clique sur le bouton RENDER

Une boule avec SPHERE

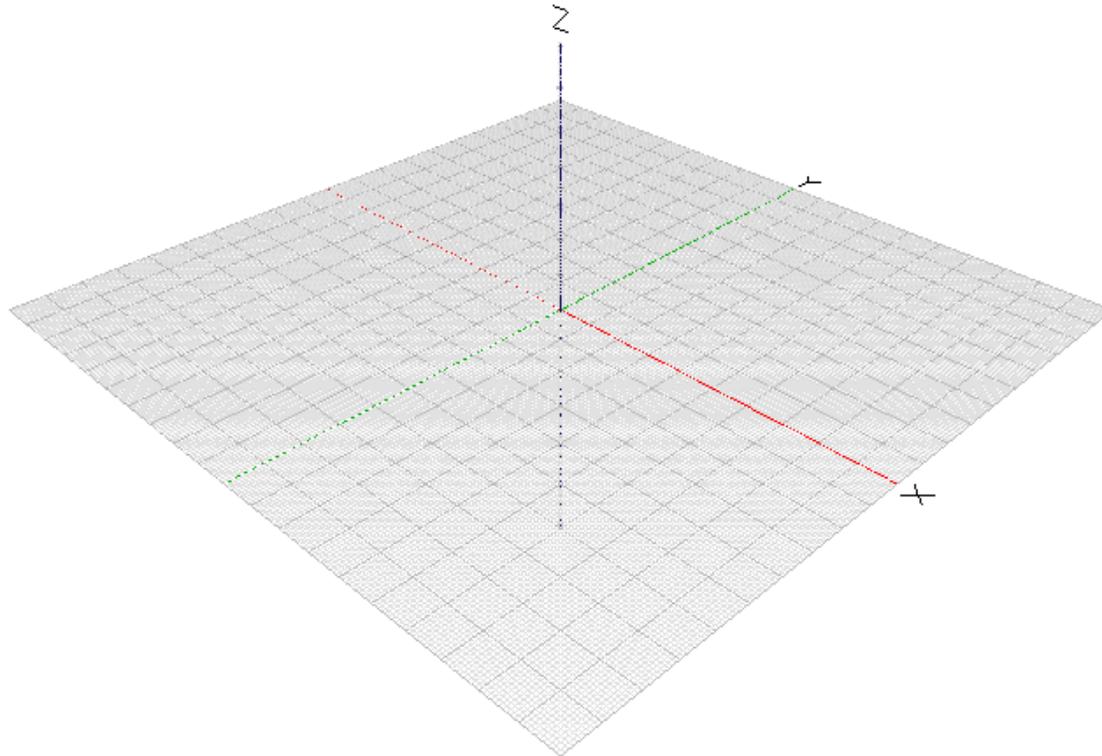
En sélectionnant, dans le menu 3D Shape, [SPHERE RADIUS 10], un bloc de programme s'affiche sur le bureau de travail. En cliquant sur RENDER, une boule de rayon 10 s'affiche.



Et voilà, d'un seul coup, vous venez de faire votre premier programme et votre premier objet.

La représentation graphique

La représentation graphique utilise les notations habituelles de la géométrie. Trois axes, X, Y et Z, se coupant suivant des angles droits (on dit « orthogonaux » en mathématique) et se coupant en un même point que l'on appelle l'origine.

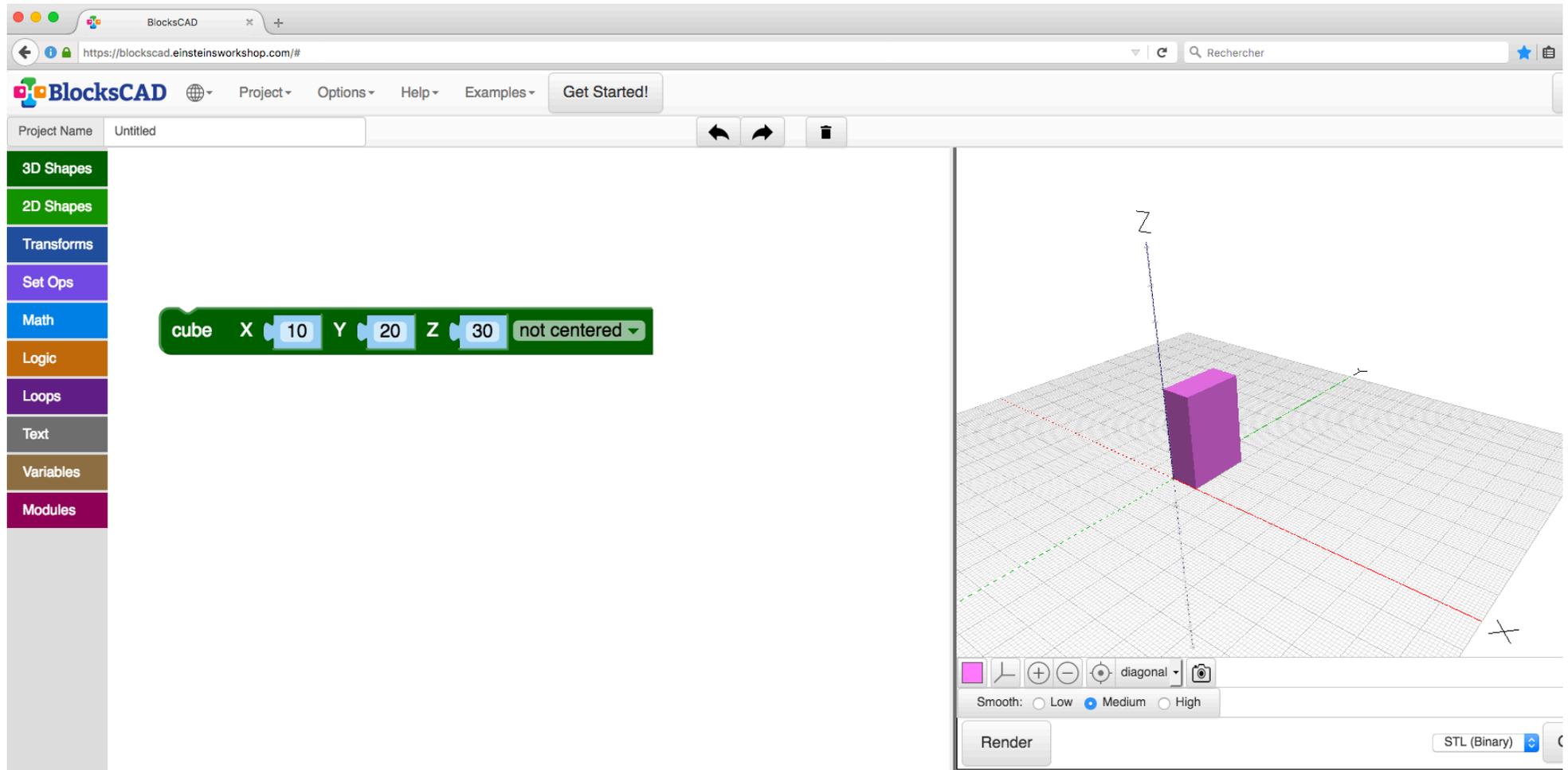


Dans une telle représentation, un point est représenté par trois nombres : l'abscisse pour l'axe X, l'ordonnée pour l'axe Y et la hauteur pour l'axe Z.

La représentation graphique de BlocksCAD s'appelle la représentation cartésienne, du nom du fameux mathématicien et philosophe Descartes qui vécut à l'époque de d'Artagnan et des trois mousquetaires. On peut le voir sur l'écran de modélisation (RENDER), une telle représentation découpe l'espace en petits carrés.

La fonction CUBE

La fonction « Cube » disponible depuis le menu de gauche « 3D SHAPE» permet la saisie de 3 nombres : l'abscisse, l'ordonnée et la hauteur. Cette fonction CUBE permet de créer tout type de parallélépipède rectangle, nom savant pour remplacer le mot « brique ».

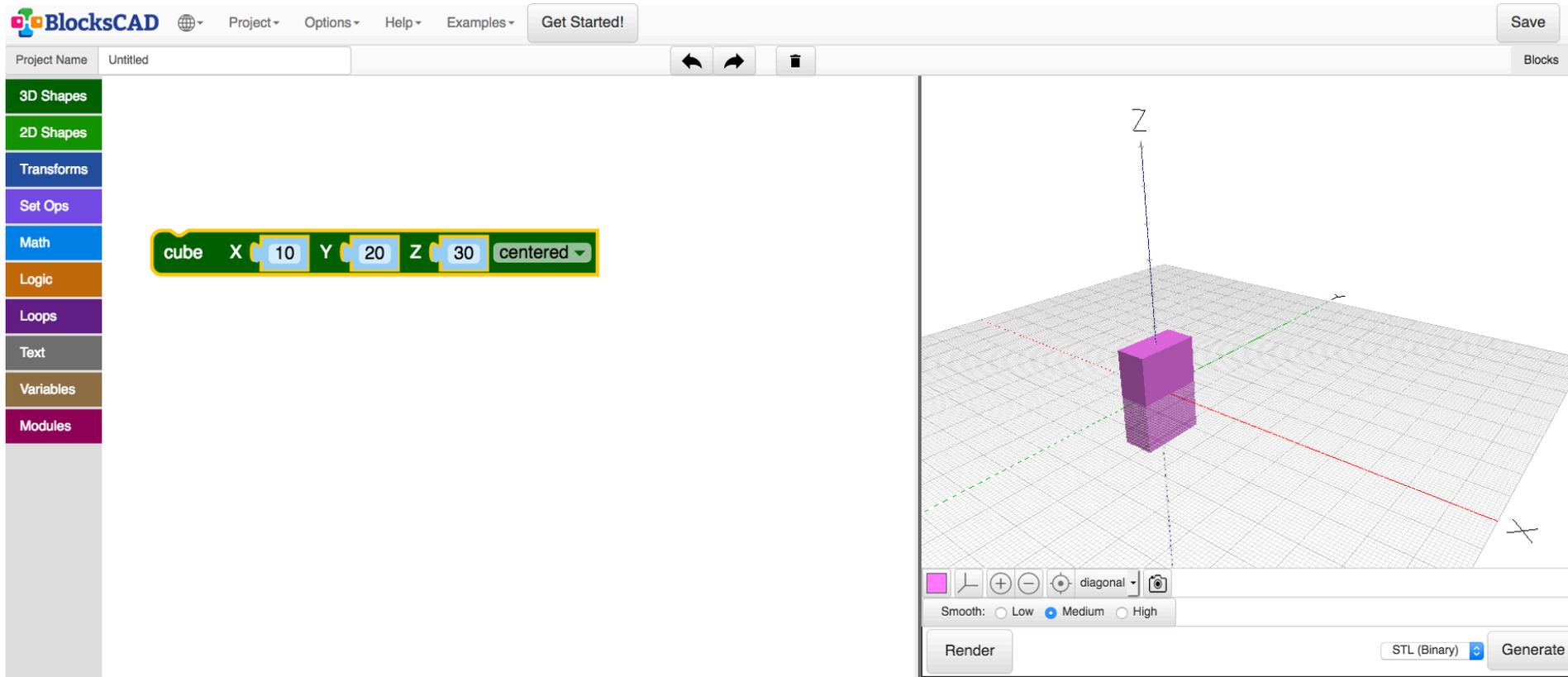


Et voici un parallélépipède rectangle de 10 (X), 20 (Y) et 30 (Z). A ce stade, les objets n'ont pas d'unité. Celle-ci pourra être choisie pendant l'impression numérique : millimètres, centimètres, décimètres, mètres ? Dans ce dernier cas, il va falloir une grande imprimante. Pour les exemples que nous donnons, nous prenons toujours l'unité pour des millimètres.

L'option CENTERED

Cette option permet de préciser la position du parallélépipède. Si l'option est sur NOT CENTERED, alors l'origine est positionné sur le coté et en bas de l'objet. Dans ce cas, les dimensions de l'objet se lisent directement sur les axes.

Si CENTERED est positionné, l'origine est au centre de l'objet.

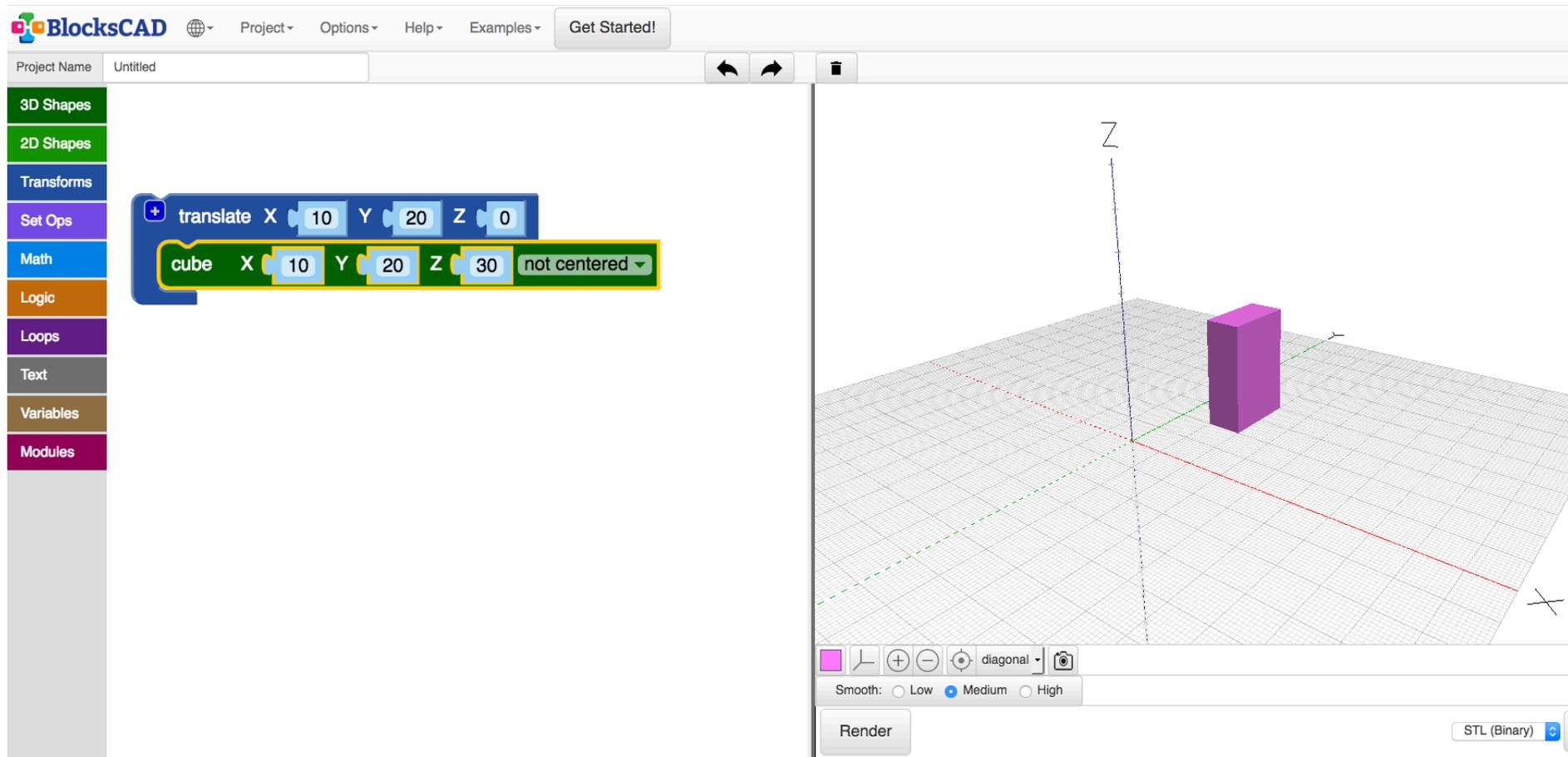


Dans ce cas, les mesures que l'on peut lire sur les axes sont égales à la moitié des mesures saisies dans la fonction CUBE.

La fonction TRANSLATE

BlocksCAD crée les objets à l'origine ou sur le coté de l'origine lorsque l'option NOT CENTERED est disponible.

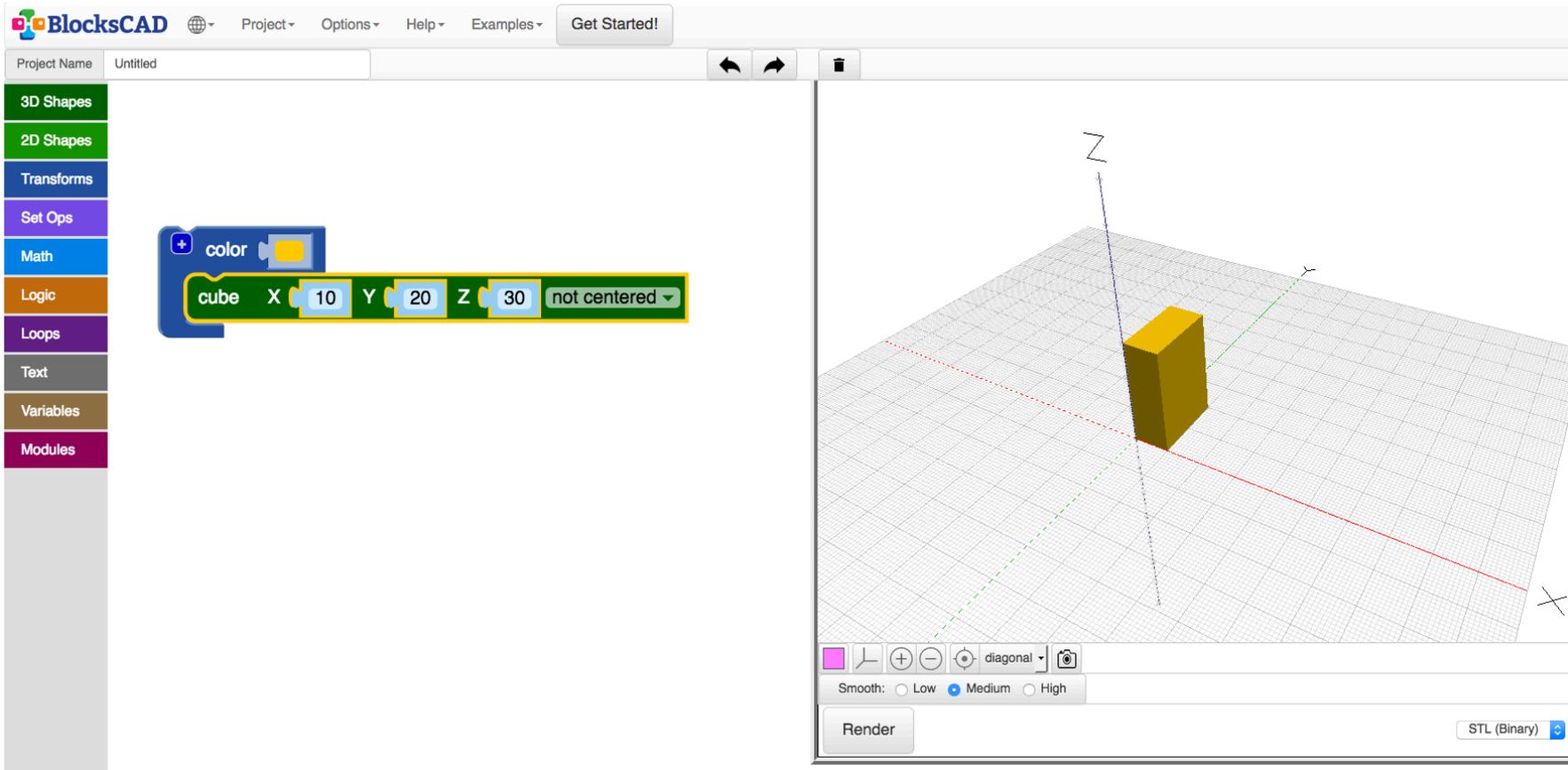
La fonction TRANSLATE disponible dans le menu de gauche « TRANSFORMS » permet de déplacer l'objet ainsi créé.



Cette fonction est très simple, mais elle est la pierre angulaire de beaucoup de réalisation que nous verrons plus tard.

Un peu de COLOR

Encore mieux que votre fidèle BIC quatre couleurs, BlocksCAD permet de coloriser le dessin de vos objets. La fonction COLOR se trouve dans le menu gauche TRANSFORM et permet de changer la couleur de l'objet entouré.

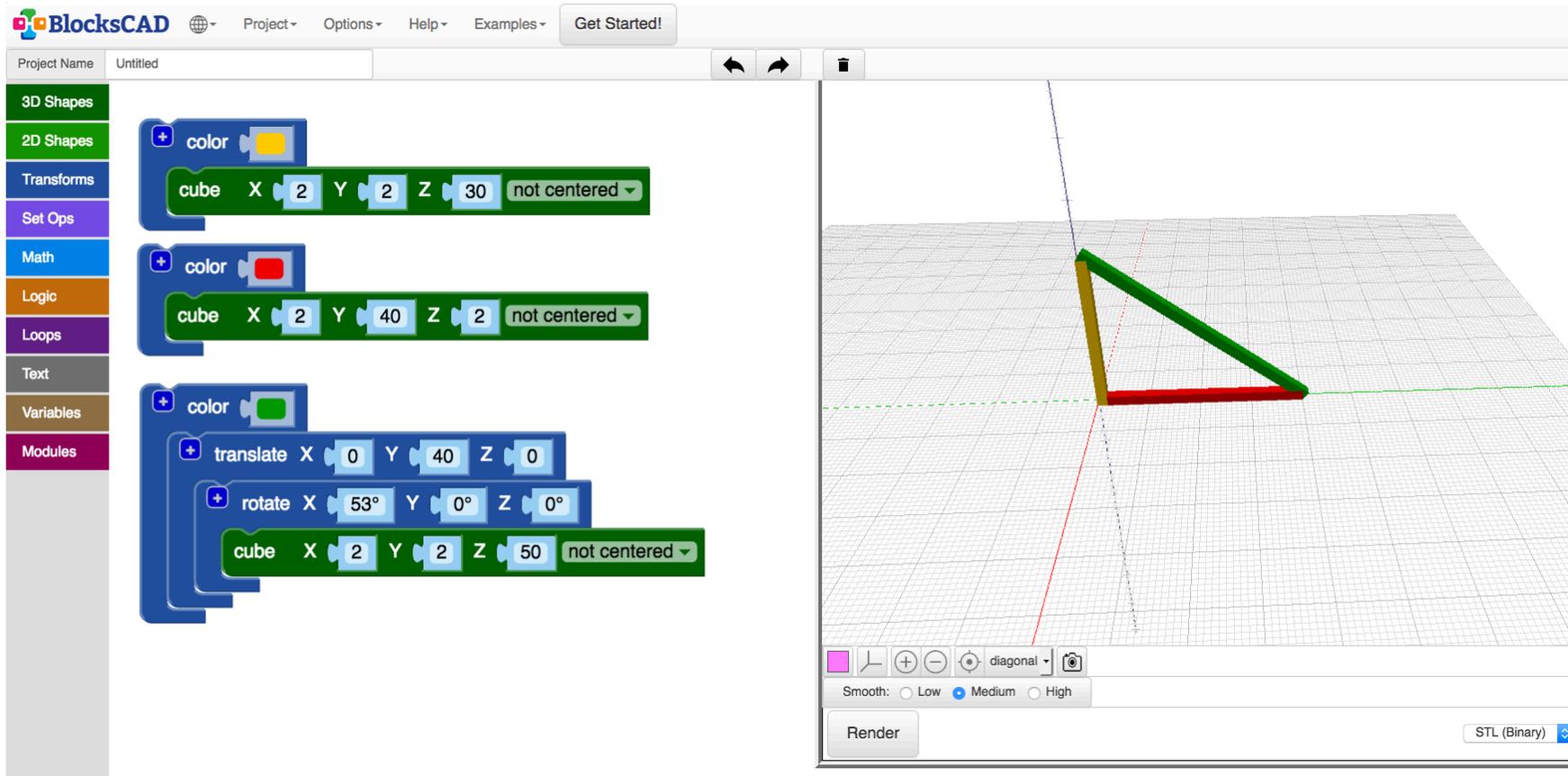


L'option COLOR sera beaucoup utilisée dans les réalisations pour faire facilement le lien entre le dessin modelé (RENDER) et les blocks des programmes.

Un triangle rectangle avec l'option ROTATE

Un triangle rectangle est un triangle qui comporte un angle droit. Nous allons le créer avec :

- une barre verticale passant par l'origine, comme dans l'exemple précédent,
- une barre horizontale, passant aussi par l'origine mais avec des dimensions différentes,
- une barre inclinée pour finir le triangle.



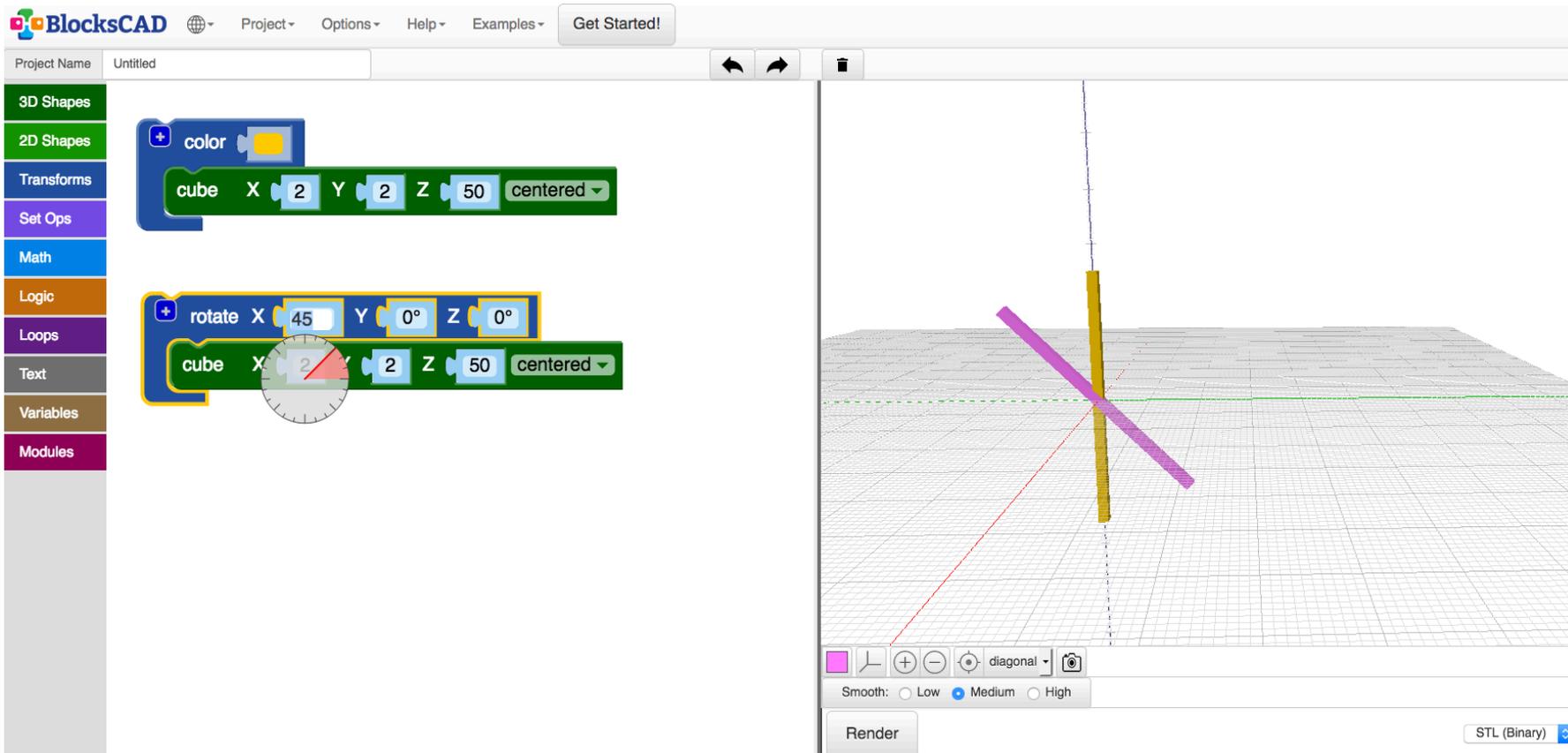
Les 2 premières barres sont des bouts de programmes déjà vus. Le suivant fait appel à une nouvelle fonction, disponible dans le menu TRANSFORM : ROTATE. Couplée avec la fonction TRANSLATE, elle permet de positionner n'importe quel objet n'importe où dans l'espace et suivant n'importe quelle orientation.

ROTATE et TRANSLATE

La fonction TRANSLATE fait glisser l'objet selon les axes comme si ceux-ci étaient des rails.

La fonction ROTATE fait tourner (et non glisser) autour de l'axe défini par le X-Y-Z de la fonction.

Regardons ce que cela fait pour l'axe X, c'est à dire que seulement le nombre sur X va être modifié.



En jaune, la barre initiale. En violet la barre qui tourne autour de l'axe X de 45° (ce qui se lit 45 degré), ce qui est un demi angle droit. L'angle droit lui 90° .

La mesure des angles : degré, radian ou grade

Les angles doivent être donnés en degré (noté $^{\circ}$) qui est une unité classique d'angle et qui ne doit pas avoir grand-chose à voir avec les degrés des températures. Dans l'unité des degrés, un tour complet vaut 360° . On en déduit qu'un angle droit vaut 90° et un angle plat 180° . Jusqu'ici tout paraît simple. Cependant, les degrés ne se divisent pas en centièmes mais en soixantièmes. Ceux-ci sont appelés des minutes d'arc. Ces dernières se subdivisent encore en soixante : les secondes d'arc. Plus simple, tu ne peux pas !

Le degré est très pratique par les navigateurs qui compte les longueurs en miles marins (Nautical Miles en anglais). Hors il se trouve que le tour de la terre à l'équateur vaut 21600 miles marins, c'est-à-dire 360×60 . Donc un mile marin représente la distance sur l'équateur pour un angle d'une minute. Donc cela tombe assez juste ce qui est pratique pour graduer les cartes, les règles marines et faire les calculs.

Il existe d'autres unités pour mesurer un angle. Les mathématiciens utilisent le radian qui est l'unité du système international . Par convention un tour complet vaut 2π . La mesure de l'angle en radian est l'arc de l'angle pour un cercle de 1 de rayon.

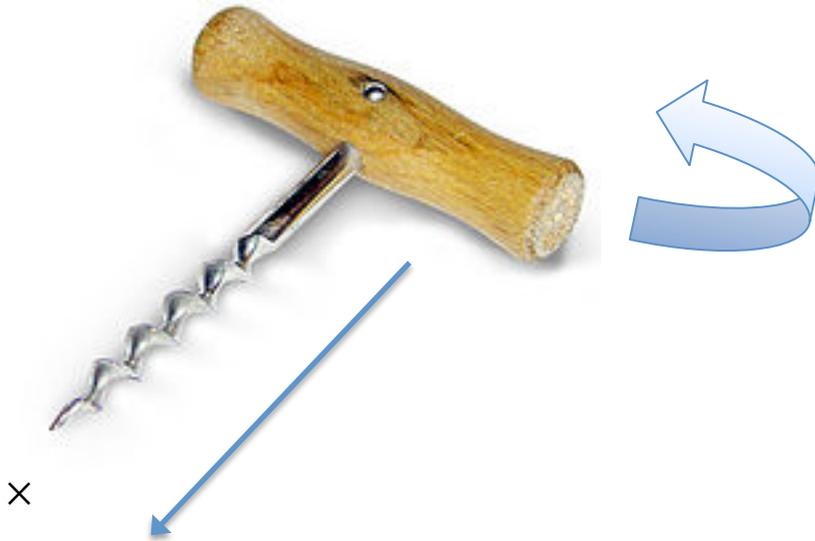
Comme nous sommes en France, notons aussi l'existence des grades avec la convention : un tour complet vaut 400 grades. Il faut rapprocher cette convention du lancement du système métrique juste après la révolution française. Et les subdivisions sont en dixièmes et centièmes. Fini, les soixantièmes ! Avec le Kilomètre, le tour de la terre vaut 40 000 km. Mais bon, à part quelques nostalgiques de la révolution, personne n'utilise le grade.

Donc finalement nous travaillons toujours en degrés aussi pour les angles dans BlocksCAD, à un petit détail : après la virgule c'est en dixièmes et centièmes. C'est quand même plus simple que les soixantièmes.

Le Tire-bouchon de Maxwell

Mais comment prévoir dans quel sens va tourner la barre violette ? Une méthode pragmatique, c'est de cliquer sur RENDER et de voir. Cependant, nous allons vous présenter une astuce de physiciens nommée le « Tire-bouchon de Maxwell » :

- ❑ imaginons un tire bouchon le long de l'axe X avec sa pointe dirigée vers l'axe X
- ❑ tournons le tire-bouchon comme si nous voulions l'enfoncer dans le bouchon
- ❑ le bord du manche indique le sens dans lequel la rotation va se faire.

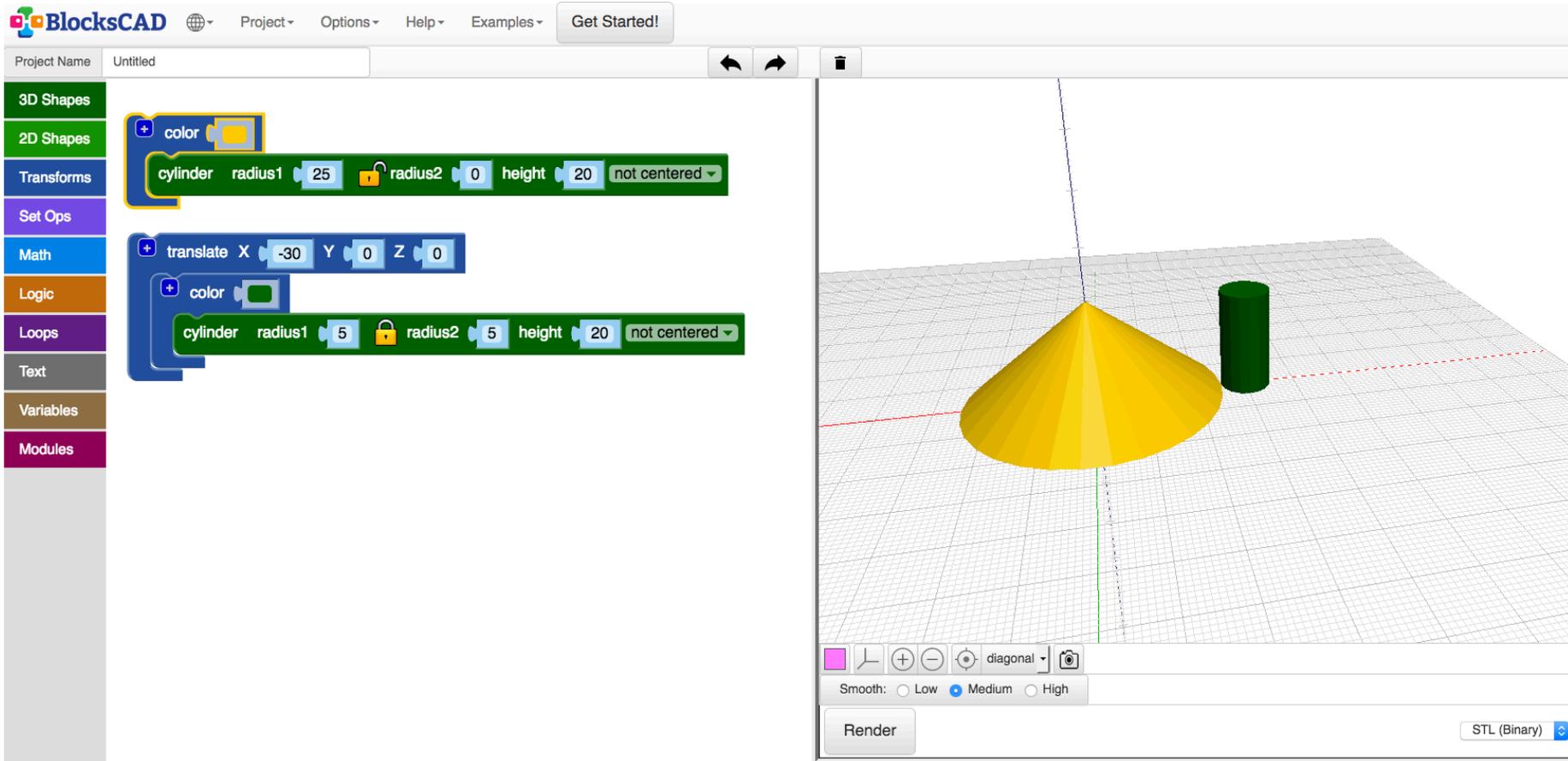


Bien sûr, il faut savoir dans quel sens tourne un tire-bouchon ! Nous comptons sur vous pour apprendre ce dernier point par vous-même.

Cylindres et cônes

Au même titre que la fonction CUBE permet de faire tout parallépipède rectangle, la fonction CYLINDER permet de faire tout type de cylindres, de cônes (le rayon du haut est à zéro, comme dans l'objet en jaune) et de tronc de cônes (les 2 rayons sont différents).

Deux petits exemples de cette famille de formes.



N'oubliez pas de cliquer pour ouvrir le cadenas si vous voulez que les 2 rayons (RADIUS) soient différents.

Les tores avec TORUS

Rien à voir avec le dieu germanique : le tore est une forme annulaire, c'est à dire un cylindre qui se retourne sur lui-même.

The screenshot shows the BlocksCAD interface. The 'torus' block is selected in the 'Math' category. The parameters are: radius1: 4, radius2: 1, sides: 8, faces: 16. A tooltip diagram shows a torus cross-section with radii r_1 and r_2 , and labels for 'sides' and 'faces'. The 3D view area shows a grid with a red dashed line and a green dashed line, and a 'Render' button with an error message 'Error: Nothing to Render'.

Pour vous aider à comprendre les différentes options dans la fonction TORUS, le petit schéma à fond jaune pâle, très instructif, apparaît lorsque l'on laisse la souris sur le bloc de fonction.

Les options SIDES et FACES

Comme pour les autres fonctions du menu de gauche 3D SHAPE, la fonction TORUS propose plusieurs options telles que « sides » et « faces ».

The screenshot shows the BlocksCAD web interface. On the left is a sidebar with categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains two blocks:

- A top block with a 'rotate' block (X: 90°, Y: 0°, Z: 0°), a 'color' block (yellow), and a 'torus' block (radius1: 10, radius2: 2, sides: 25, faces: 25).
- A bottom block with a 'translate' block (X: -10, Y: 0, Z: 0) and a 'torus' block (radius1: 10, radius2: 2, sides: 3, faces: 4).

The 3D view on the right shows a yellow torus with a purple coordinate system. Below the view are buttons for 'Render', 'STL (Binary)', and 'Generate STL'. The interface also includes a 'Project Name' field with 'LeLivre', navigation arrows, a trash icon, and a search bar.

Et voici deux exemples de tores imbriqués avec des paramètres différents. A vous de changer les valeurs pour voir la différence.

La réunion de différentes formes avec UNION

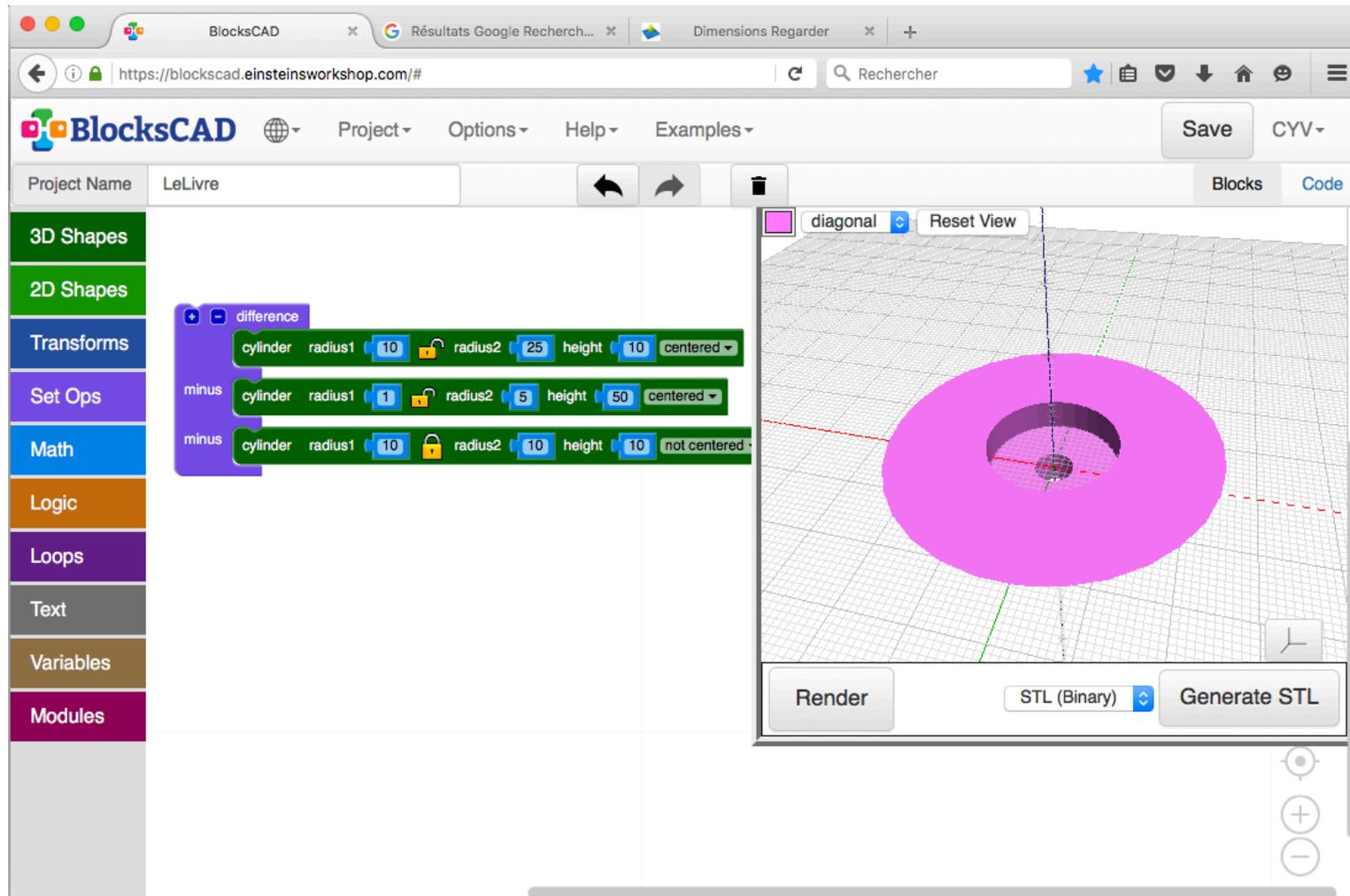
Visualisons un petit objet composé d'un cône et d'un cylindre. Ces deux objets sont imbriqués dans la fonction UNION issue du menu gauche « SET OPS ».

The screenshot shows the BlocksCAD web interface. The browser address bar displays <https://blockscad.einsteinsworkshop.com/#>. The page title is "BlocksCAD". The sidebar on the left contains the following categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The "Set Ops" category is selected, and a "union" block is visible in the workspace. The "union" block contains three "cylinder" blocks. The first cylinder has radius1: 10, radius2: 25, height: 10, and is centered. The second cylinder has radius1: 1, radius2: 5, height: 50, and is centered. The third cylinder has radius1: 10, radius2: 10, height: 10, and is not centered. The 3D view shows a purple object composed of a cylinder and a cone. The "Render" and "Generate STL" buttons are visible at the bottom of the 3D view.

Cette fonction regroupe les 2 objets en un seul. Cela permet de plus facilement gérer un objet composé de plusieurs formes de bases. En appuyant sur le pictogramme (+) de la fonction UNION on peut rajouter autant de formes et de blocs de programmes que l'on veut.

La différence de différents formes avec DIFFERENCE

Toujours dans le menu SET OPS, l'opération DIFFERENCE, permet d'enlever facilement de la matière dans un objet. En reprenant les mêmes objets de base, le résultat devient bien différent.



Bien gérer la DIFFERENCE

L'opération DIFFERENCE prend le premier objet et tous les objets suivants sont enlevés. Si nous désirons que seul le troisième soit ôté des 2 premiers, il est nécessaire d'utiliser l'opération UNION avant d'appliquer l'opération DIFFERENCE.

The screenshot shows the BlocksCAD interface. The top bar includes the logo, navigation menus (Project, Options, Help, Examples), and buttons for Save and CYV. The Project Name is 'LeLivre'. The left sidebar contains a list of categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace displays a 3D model of a purple object, which is a cylinder with a smaller cylinder removed from its top and a larger cylinder removed from its bottom. The code blocks are as follows:

- A 'difference' block containing:
 - A 'union' block containing:
 - A 'cylinder' block with radius1: 10, radius2: 25, height: 10, centered.
 - A 'plus' block containing:
 - A 'cylinder' block with radius1: 1, radius2: 5, height: 50, centered.
 - A 'minus' block containing:
 - A 'cylinder' block with radius1: 10, radius2: 10, height: 10, not centered.

The 3D view shows the object in a diagonal perspective. The bottom bar includes a 'Render' button, a dropdown menu set to 'STL (Binary)', and a 'Generate STL' button.

L'ordre des opérations a donc son importance.

Un machin planétaire

Avant d'aller plus loin dans les opérations (SET OPS) et autres transformations (TRANSFORMS) de BlocksCAD, voici un petit exemple proposé par Quitrie que vous pouvez facilement faire dès à présent.

The screenshot displays the BlocksCAD software interface. On the left, a sidebar lists various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace is filled with a complex assembly of blocks. These blocks include 'difference' operations (purple), 'minus' operations (green), 'rotate' (blue), 'color' (orange), 'cylinder' (green), 'sphere' (green), and 'translate' (blue). The blocks are arranged in a hierarchical structure, representing the construction of a 3D model. On the right side of the interface, a 3D view shows the resulting model: a brown planet with a yellow ring system, set against a grey grid background. Below the 3D view, there are controls for rendering, including a 'Render' button, a 'Smooth' dropdown menu (set to 'Medium'), and a 'Generate STL' button. The project name 'MACHIN' is visible at the top left of the workspace.

De quoi aborder avec sérénité vos prochaines lectures sur les étoiles.

L'intersection de formes avec INTERSECTION

L'avant dernière opération, INTERSECTION, ne prend que la matière qui est commune à tous les objets concernés. Toujours avec les mêmes objets des images précédentes, le résultat d'INTERSECTION.

The screenshot displays the BlocksCAD web application interface. The browser address bar shows the URL <https://blockscad.einsteinsworkshop.com/#>. The page title is "BlocksCAD". The navigation menu includes "Project", "Options", "Help", and "Examples". The "Project Name" field contains "LeLivre". The "Save" button is visible, along with a "CYV" dropdown. The left sidebar lists various categories: "3D Shapes", "2D Shapes", "Transforms", "Set Ops", "Math", "Logic", "Loops", "Text", "Variables", and "Modules". The main workspace shows a 3D grid with a purple cylinder and a hole. The top bar of the workspace includes a "diagonal" dropdown and a "Reset View" button. The bottom bar of the workspace has a "Render" button, a "STL (Binary)" dropdown, and a "Generate STL" button. The workspace also features a "Blocks" and "Code" tab.

HULL

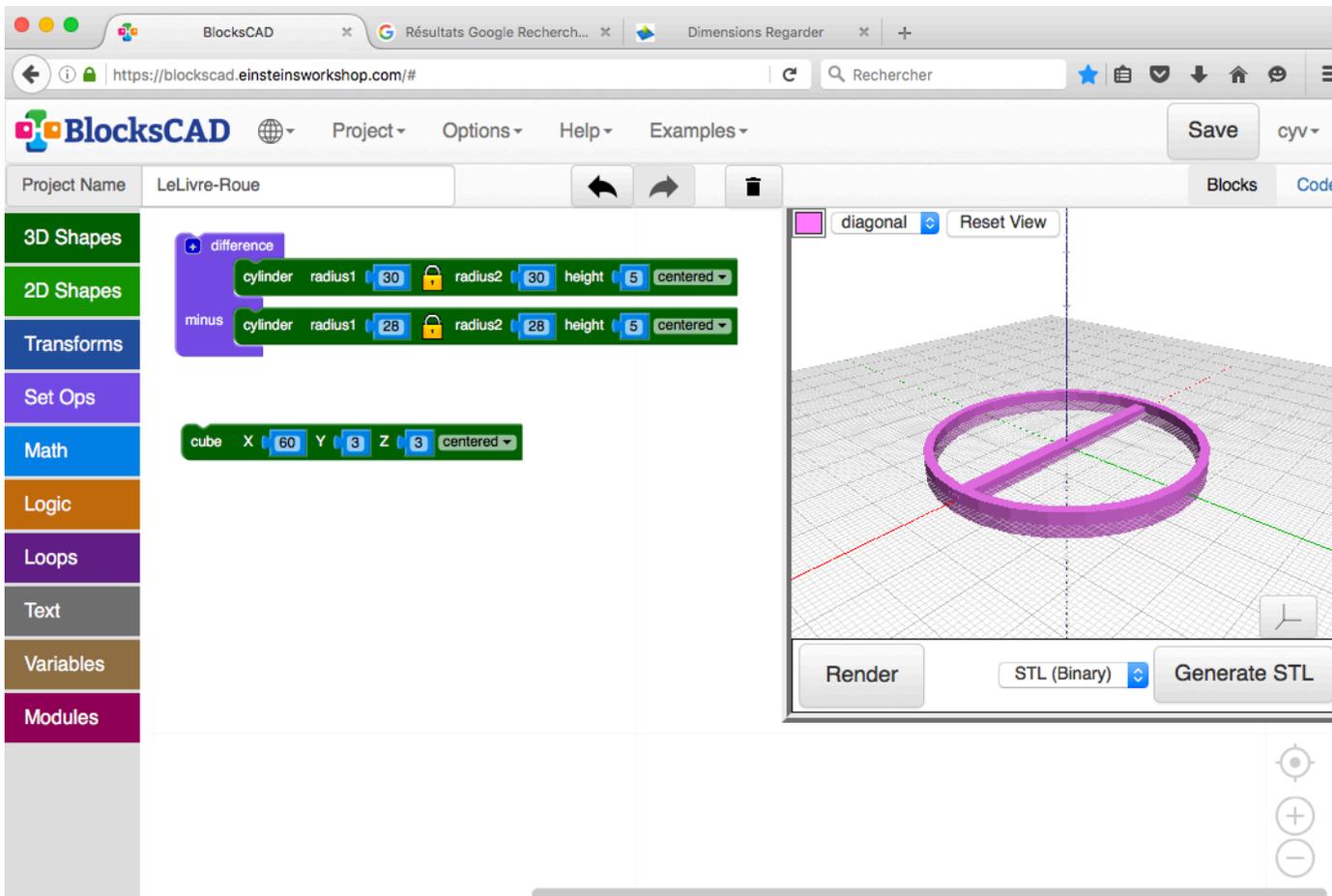
HULL signifie « coque » en anglais. Cette opération est très pratique et crée une forme entourant tous les objets concernés. Toujours avec les mêmes objets, le résultat est le suivant.

The screenshot displays the BlocksCAD web application interface. The browser address bar shows the URL <https://blockscad.einsteinsworkshop.com/#>. The page title is "BlocksCAD". The navigation menu includes "Project", "Options", "Help", and "Examples". The project name is "LeLivre". The interface is divided into a left sidebar with categories: "3D Shapes", "2D Shapes", "Transforms", "Set Ops", "Math", "Logic", "Loops", "Text", "Variables", and "Modules". The main workspace shows a 3D view of a purple hull shape on a grid. The code editor on the left contains a "hull" block with three "with" sub-blocks, each containing a "cylinder" block with specific parameters: radius1, radius2, height, and centered/not centered options. The 3D view includes a "diagonal" view selector, a "Reset View" button, and a "Render" button. Below the 3D view, there is a "STL (Binary)" dropdown menu and a "Generate STL" button. The bottom right corner of the interface has zoom controls (+, -) and a reset view icon.

Une roue (simple) de charrette

Maintenant que nous connaissons la plupart des opérations et transformations pour manipuler des formes, réalisons une roue de charrette. Elle pourra se complexifier plus tard et servir de base à d'autres types de roue.

Commençons par l'extérieur de la roue : une DIFFERENCE entre deux cylindres. Le plus grand ayant 30 de rayon. Rajoutons le premier diamètre, sous forme d'un parallélépipède rectangle qui doit donc être de 2 fois le rayon soit 60.



Rappelez vous le tire-bouchon de Maxwell. Il faut l'orienter vers le haut donc vers l'axe Z et mettre le manche en bois dans en superposition du diamètre.

Réalisons les autres rayons, avec le même objet CUBE que le précédent et la transformation ROTATE.
Le nouveau diamètre est coloré en jaune pour bien voir la différence avec le premier.

The screenshot shows the BlocksCAD web interface. The browser address bar displays <https://blockscad.einsteinsworkshop.com/#>. The page title is "BlocksCAD" and the project name is "LeLivre-Roue". The interface includes a sidebar with categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains a 3D model of a ring with a diagonal and a rotated diameter. The diagonal is colored pink, and the rotated diameter is colored yellow. The 3D view is labeled "diagonal" and "Reset View". The interface also features a "Render" button, a "STL (Binary)" dropdown, and a "Generate STL" button.

Project Name: LeLivre-Roue

3D Shapes

2D Shapes

Transforms

Set Ops

Math

Logic

Loops

Text

Variables

Modules

difference

- cylinder radius1: 30 radius2: 30 height: 5 centered
- minus cylinder radius1: 28 radius2: 28 height: 5 centered

cube X: 60 Y: 3 Z: 3 centered

color

- rotate X: 0° Y: 0° Z: 30°
- cube X: 60 Y: 3 Z: 3 centered

diagonal Reset View

Render STL (Binary) Generate STL

Et maintenant, terminons les autres diamètres.

The screenshot shows the BlocksCAD web application interface. At the top, there is a browser window with the URL `https://blockscad.einsteinsworkshop.com/#`. Below the browser, the BlocksCAD logo and navigation menu are visible, including 'Project', 'Options', 'Help', and 'Examples'. A 'Save' button is present with the text 'cyv'. The main interface is divided into a left sidebar and a central 3D view area.

The left sidebar contains a 'Project Name' field with the text 'LeLivre-Roue'. Below this is a vertical menu with categories: '3D Shapes', '2D Shapes', 'Transforms', 'Set Ops', 'Math', 'Logic', 'Loops', 'Text', 'Variables', and 'Modules'. Each category has a corresponding block palette with various geometric and transformation blocks. For example, under '3D Shapes', there are blocks for 'difference', 'cylinder', and 'minus'. Under 'Transforms', there are blocks for 'cube' with X, Y, and Z coordinates.

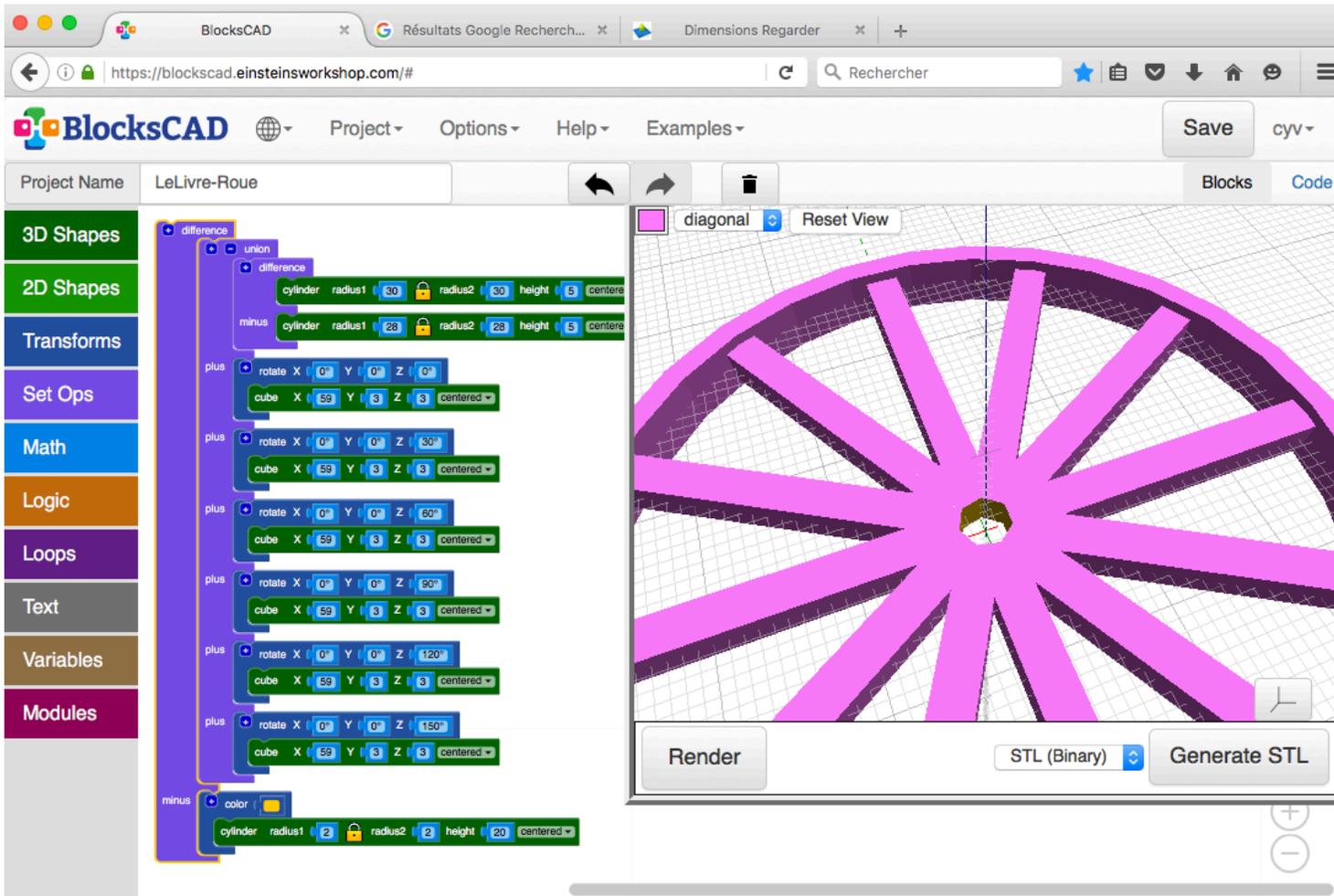
The central 3D view area displays a 3D model of a wheel. The wheel has a purple rim and several yellow spokes. The view is set to 'diagonal' and includes a 'Reset View' button. At the bottom of the 3D view area, there is a 'Render' button and a 'Generate STL' button with a dropdown menu set to 'STL (Binary)'. The 3D view area also features a grid and coordinate axes.

Et pour finir, terminons le moyeu. Passons néanmoins par une étape intermédiaire en regroupant la roue (UNION) et les diamètres et en jaune le futur trou du moyeu.

The screenshot displays the BlocksCAD web application interface. The browser address bar shows the URL `https://blockscad.einsteinsworkshop.com/#`. The page title is "BlocksCAD" and the project name is "LeLivre-Roue". The interface includes a sidebar with categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D model of a wheel with a central hub, rendered in purple and yellow. The code editor on the left contains the following blocks:

- union
 - difference
 - cylinder radius1: 30, radius2: 30, height: 6, centered
 - minus
 - cylinder radius1: 28, radius2: 28, height: 6, centered
 - plus
 - cube X: 59, Y: 3, Z: 3, centered
 - plus
 - rotate X: 0°, Y: 0°, Z: 30°
 - cube X: 59, Y: 3, Z: 3, centered
 - plus
 - rotate X: 0°, Y: 0°, Z: 60°
 - cube X: 59, Y: 3, Z: 3, centered
 - plus
 - rotate X: 0°, Y: 0°, Z: 90°
 - cube X: 59, Y: 3, Z: 3, centered
 - plus
 - rotate X: 0°, Y: 0°, Z: 120°
 - cube X: 59, Y: 3, Z: 3, centered
 - plus
 - rotate X: 0°, Y: 0°, Z: 150°
 - cube X: 59, Y: 3, Z: 3, centered
- color
 - cylinder radius1: 2, radius2: 2, height: 20, centered

Terminons donc avec l'opération DIFFERENCE entre l'objet violet et l'objet jaune.

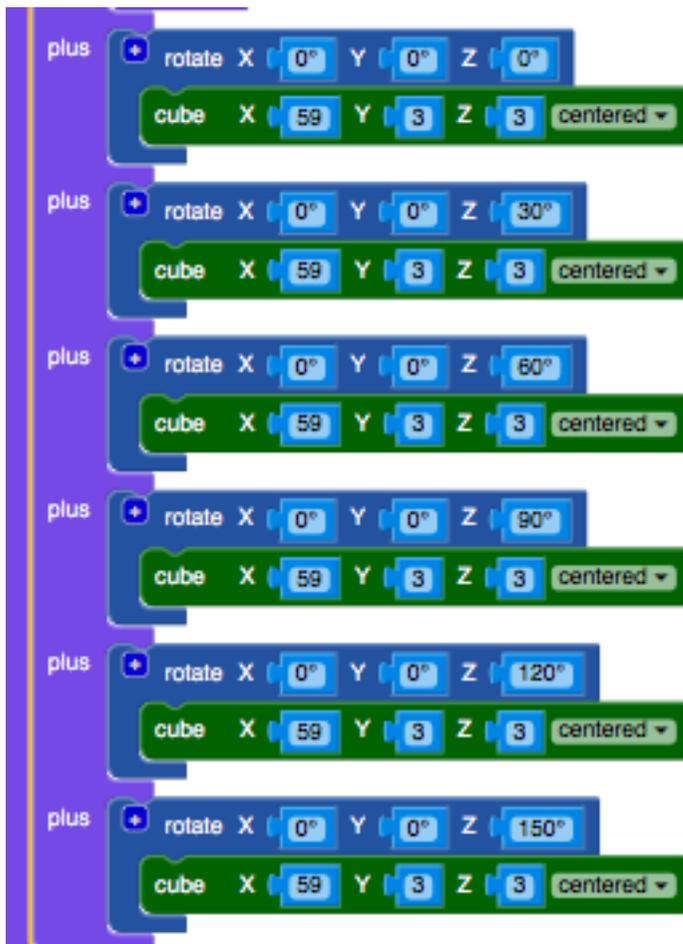


Et nous voici avec notre roue et un trou pour passer le moyeu. Remarquez au passage que nous avons rajouté une transformation ROTATE sur le premier diamètre, tout en laissant les angles de rotation à zéro. Ceci ne change rien à ce diamètre mais ce bloc de programme ressemble maintenant à tous les autres.

LOOP, une boucle informatique

Après toutes les fonctions géométriques de formes et de transformations, il est temps de voir notre première fonction informatique : LOOP, la boucle.

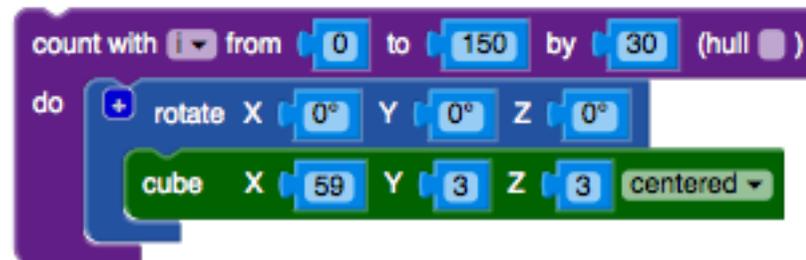
Regardez ci-dessous un zoom du code de l'image précédente. C'est toujours le même code, avec une simple différence à chaque fois : Z=0 pour le premier, Z = 30 , Z = 60 ... jusqu'à Z=150.



Malgré leur absence d'humour, les ordinateurs ont deux vraies forces : ils savent facilement faire des calculs (même plus nécessaire d'apprendre ses tables de multiplications ? non je rigole) et ils savent répéter la même chose sans s'ennuyer.

Nous allons donc utiliser **la fonction LOOP** comme ceci.

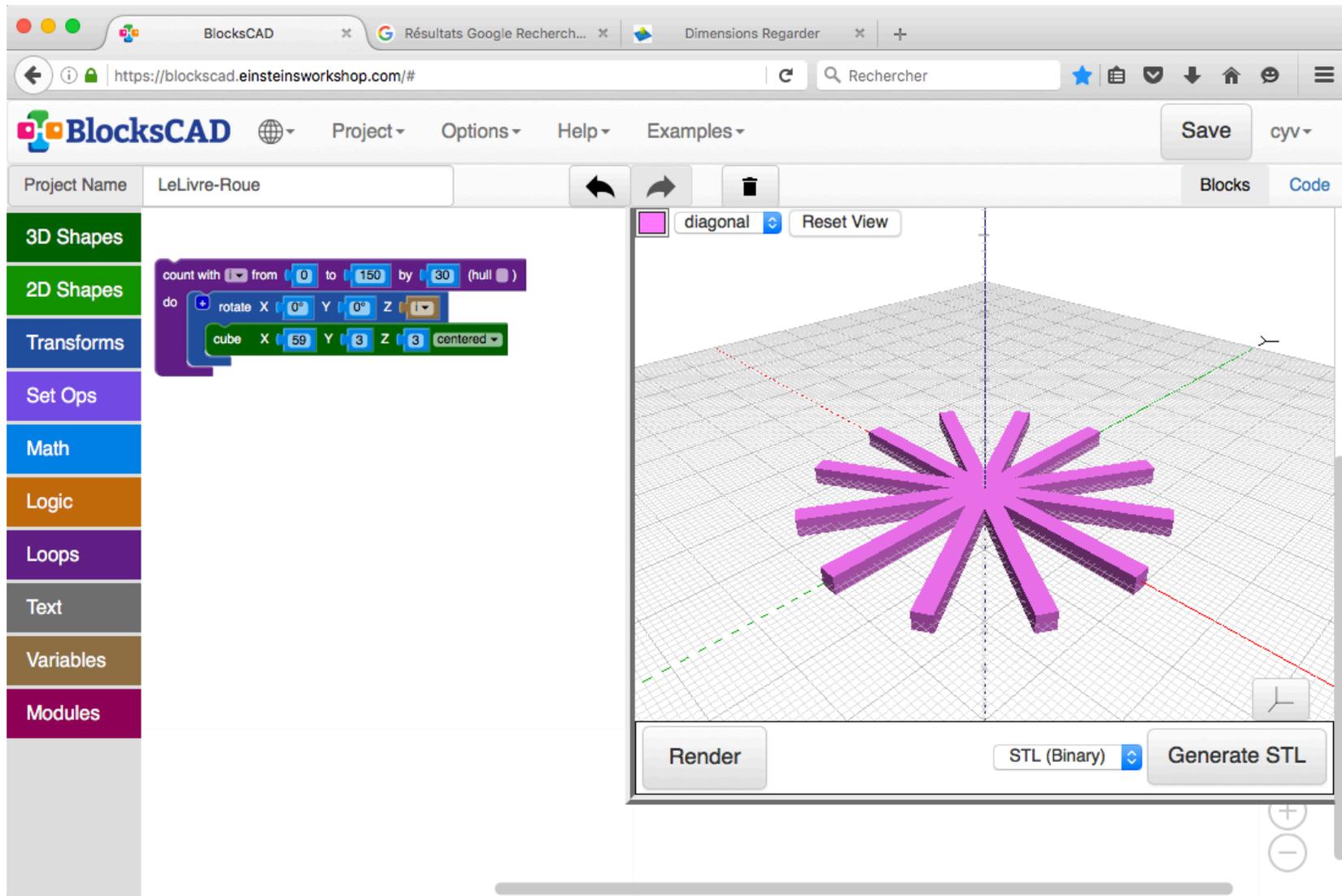
Nous avons besoin de 6 diamètres, donc nous allons compter 6 fois (**COUNT**), de (**FROM**) 0 à (**TO**) 150 avec des sauts (**BY**) de 30. La variable (i) sera donc successivement égal à : 0, 30, 60, 90, 120, 150.



« i » est appelé une variable en mathématique, tout simplement car sa valeur varie à chaque boucle. La petite case « i » est disponible dans le menu de gauche « Variables ».

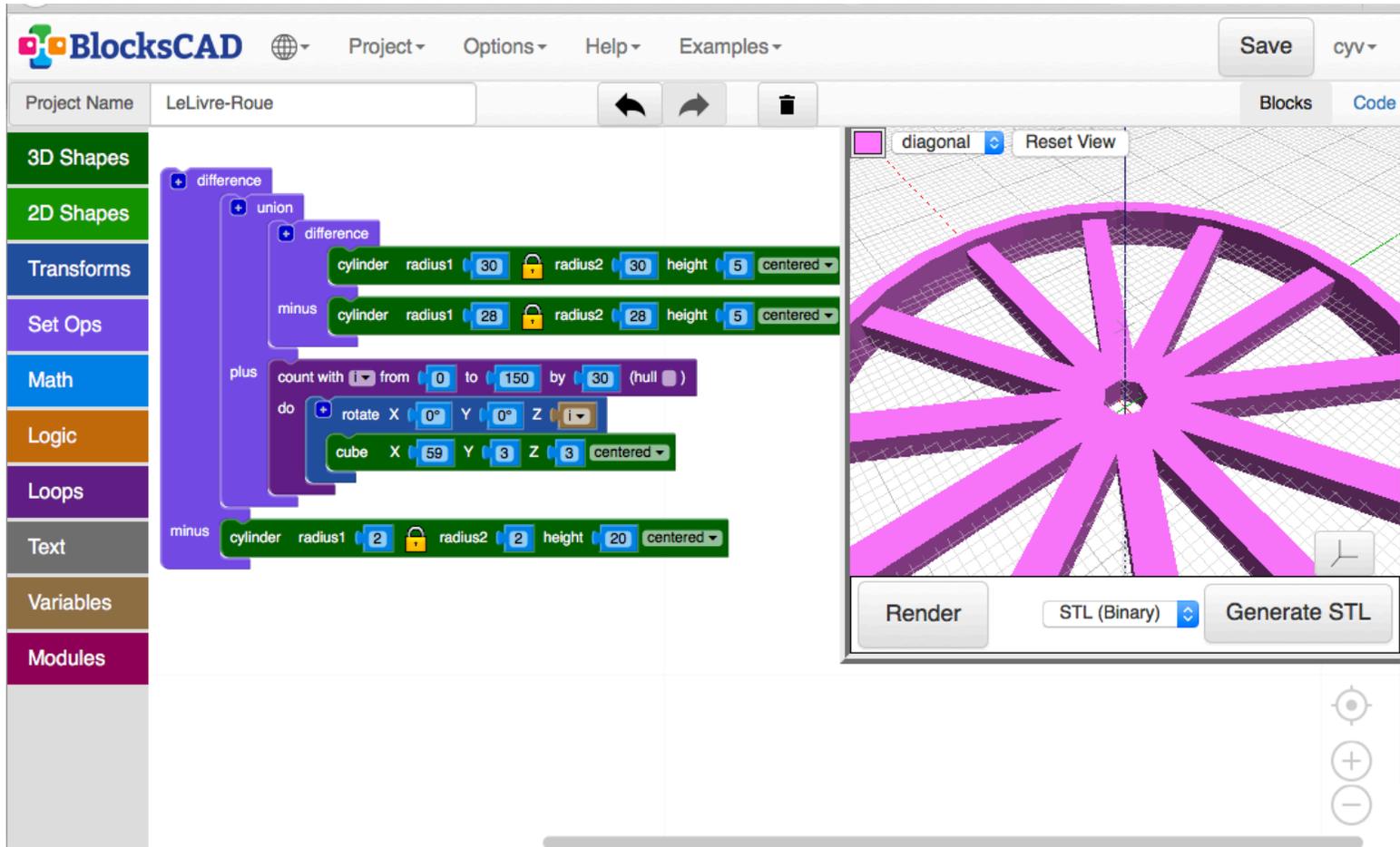


Et voilà notre programme avec sa boucle.



Il prend beaucoup moins de place, il est plus facile à lire et surtout plus facile à écrire.

Pour conclure, voyons avec le programme complet.



Pourquoi utiliser cette fonction LOOP ? Pour plusieurs raisons :

- le programme est plus simple, donc plus lisible
- le programme peut être plus facilement changé (on dit aussi « maintenir » dans le clan des informaticiens !) : on peut facilement augmenter le nombre de rayon, changer la forme du diamètre, par exemple.

Par exemple, nous voulons changer l'épaisseur des diamètres. Il suffit de modifier (Y=2, Z=2) dans CUBE une seule fois.

The screenshot displays the BlocksCAD web application interface. The browser address bar shows the URL `https://blockscad.einsteinsworkshop.com/#`. The page title is "BlocksCAD" and the project name is "LeLivre-Roue". The interface includes a sidebar with categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D model of a wheel with pink spokes and a purple rim, viewed from a diagonal perspective. The code blocks are as follows:

- A **difference** block containing:
 - A **union** block containing:
 - A **difference** block containing:
 - A **cylinder** block with `radius1` 30, `radius2` 30, and `height` 5.
 - A **minus** block containing a **cylinder** block with `radius1` 28, `radius2` 28, and `height` 5.
 - A **plus** block containing:
 - A **count with** block with `from` 0, `to` 150, and `by` 30.
 - A **do** loop containing:
 - A **rotate** block with `X` 0°, `Y` 0°, and `Z` i.
 - A **cube** block with `X` 59, `Y` 2, and `Z` 2.
 - A **minus** block containing a **cylinder** block with `radius1` 2, `radius2` 2, and `height` 20.

Le module « faire une roue »

Ensuite, il faut pouvoir facilement utiliser cette roue. Pour cela le menu de gauche nous donne la possibilité de créer un MODULE. Changez le nom du module et insérez le programme dedans. Retournez dans le menu MODULES et sélectionnez l'appel du module « faire une roue ». En effet, une fois le module créé, il est nécessaire de donner l'ordre à l'ordinateur de l'utiliser.

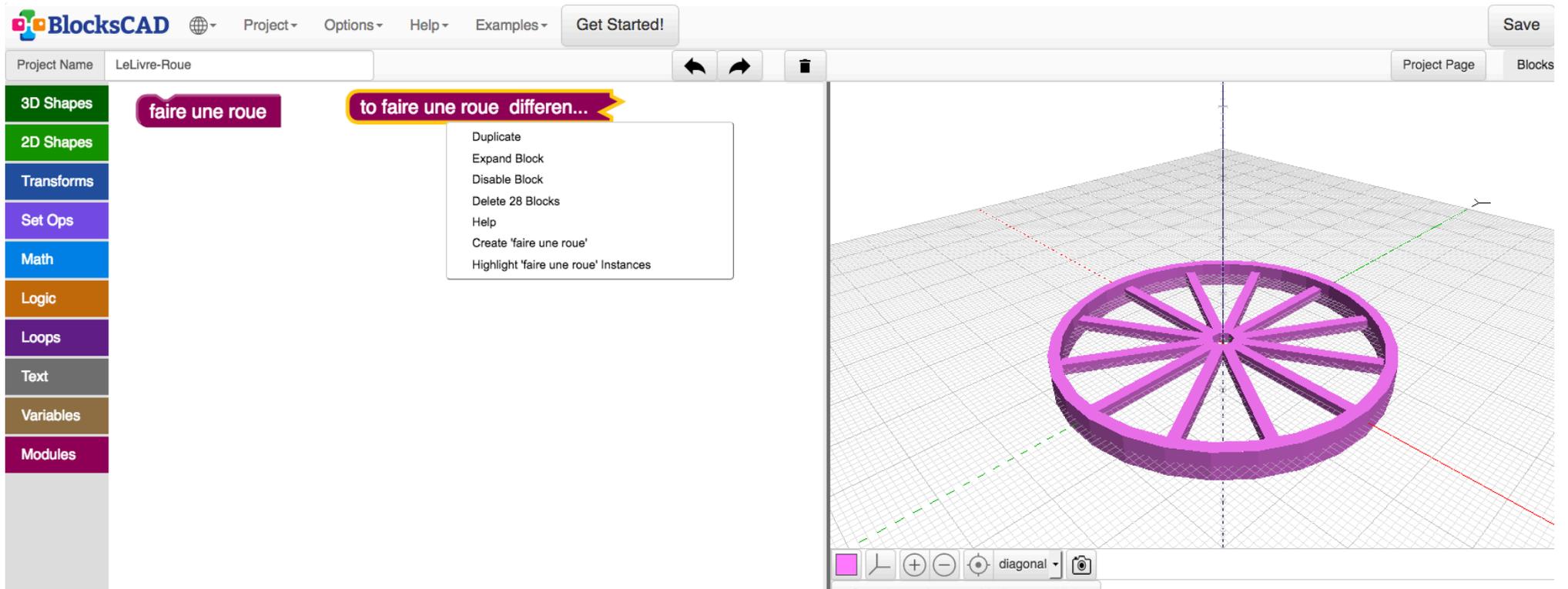
The screenshot shows the BlocksCAD web interface. The browser address bar displays <https://blockscad.einsteinsworkshop.com/#>. The page title is "BlocksCAD" and the project name is "LeLivre-Roue". The interface includes a sidebar with categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D model of a wheel with a purple rim and spokes. The code blocks are as follows:

```
to faire une roue
  difference
    union
      difference
        cylinder radius1 30 radius2 30 height 5 centered
      minus
        cylinder radius1 28 radius2 28 height 5 centered
    plus
      count with from 0 to 150 by 30 (hull)
      do
        rotate X 0° Y 0° Z
        cube X 59 Y 2 Z 2 centered
  minus
    cylinder radius1 2 radius2 2 height 20 centered
```

The 3D view includes a "diagonal" view selector and a "Reset View" button. At the bottom of the 3D view, there are buttons for "Render", "STL (Binary)", and "Generate STL".

L'appel du module peut être utilisé autant de fois que vous voulez, à l'intérieur d'un programme ou hors comme dans notre exemple, dans une boucle, etc.

Et enfin pour terminer, un clic-droit sur le module « faire une roue » et dans le menu, on choisit « collapse block » pour simplement ranger le programme. Ne pas s'inquiéter, rien n'est perdu. Pour voir le programme complet, il suffira de faire clic-droit sur le bloc puis de sélectionner, dans le menu contextuel qui apparaît, la ligne EXPAND BLOCK.



Et puis, faire SAVE, en haut à gauche pour tout enregistrer.

Un champignon

Petit intermède proposé par Xavière sous la forme d'un champignon. Il vous permettra de décorer soit votre jardin, soit votre arbre de Noël soit votre planète ou votre machin planétaire si vous vous appelez le Petit Prince.

The screenshot shows the BlocksCAD software interface. On the left, the 'Project Name' is 'Mon Champignon'. The code editor contains the following blocks:

- translate X 0 Y 0 Z 40
- color [red]
- difference
 - sphere radius 30
 - cylinder radius1 8 radius2 10 height 50 not centered
- minus
 - translate X 0 Y 0 Z -20
 - sphere radius 35
- count with from 60 to 360 by 120 (hull)
 - do
 - rotate X 0° Y 0° Z [arrow]
 - translate X 0 Y 25 Z 10
 - color [white]
 - sphere radius 5
- count with from 0 to 360 by 120 (hull)
 - do
 - rotate X 0° Y 0° Z [arrow]
 - translate X 0 Y 20 Z 20
 - color [white]
 - sphere radius 5
- translate X 0 Y 0 Z 28
- color [white]
- sphere radius 5

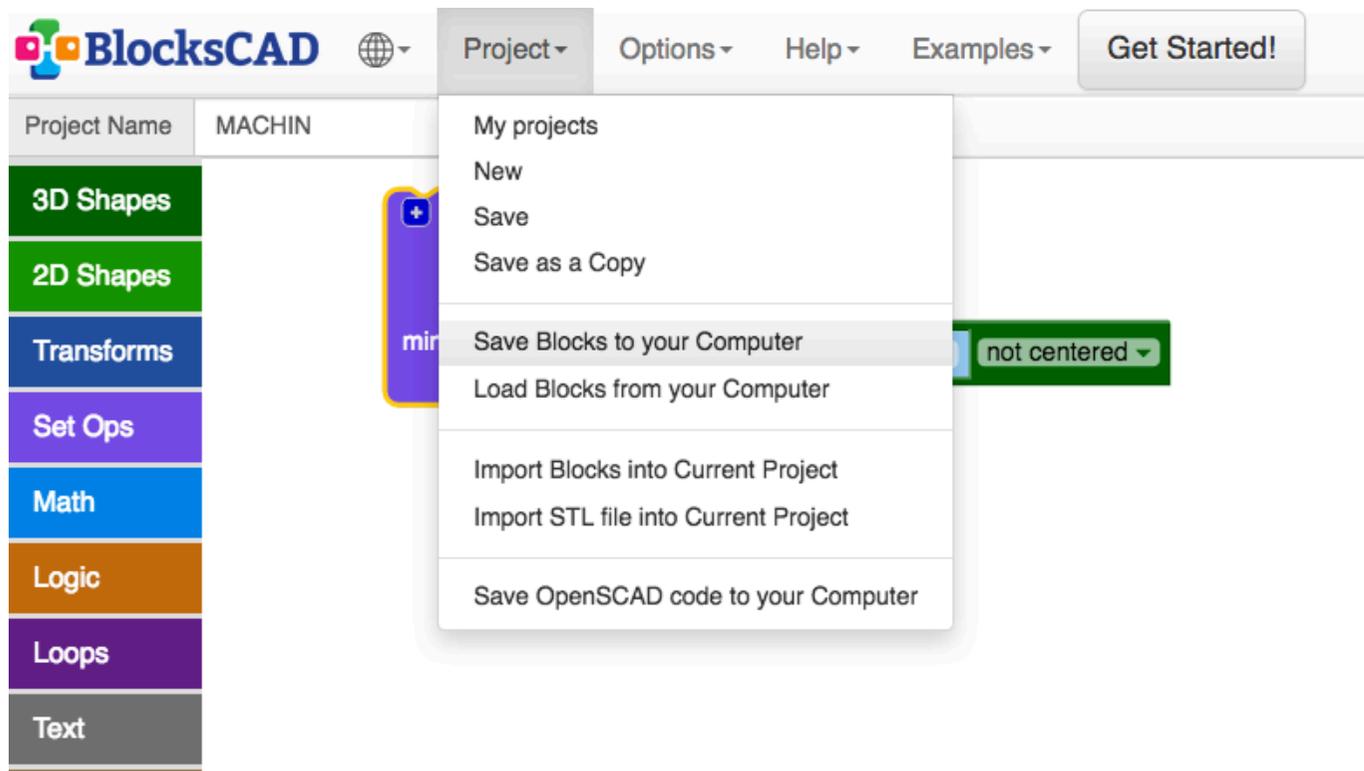
The 3D view shows a red mushroom with a grey stem and five white spots on its cap. The interface includes a 'Render' button and a 'Generate STL' button.

Evitez tout de même de le manger !

Un peu d'organisation

Avant d'aller plus loin, parlons un peu organisation. Dès que l'on réalise des programmes, ceux-ci s'accumulent et puis assez vite, cela devient compliqué de les retrouver, de les gérer et de les modifier.

Pour commencer, nous avons ouvert un compte dans « BlocksCAD ». Cela permet de sauvegarder tous les programmes mais surtout de faire SAVE (en haut à droite) régulièrement. Une fois un programme terminé, il faut aussi le sauvegarder sur votre ordinateur en faisant « **Save Block to your computer** ».



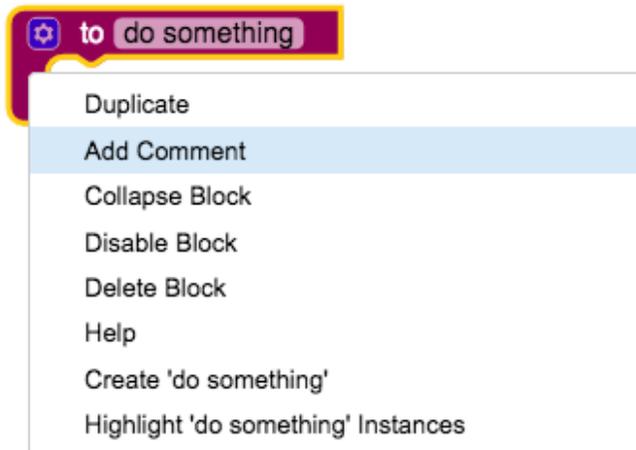
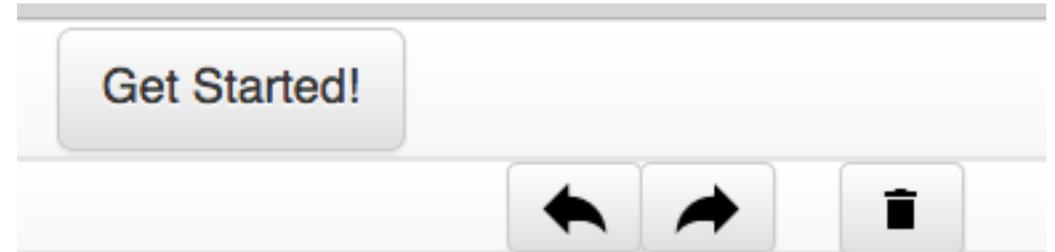
Mettre tous vos programmes dans un dossier spécifique, cela permettra de les récupérer ensuite. En effet, quand on veut utiliser un ancien block dans un programme en cours, il est facile de faire « **IMPORT Blocks into current project** », de choisir le block désiré et de le faire apparaître avec le programme.

Programmez en COLOR

Pour réaliser un block, n'hésitez pas à utiliser la fonction COLOR : cela permet de plus facilement s'y retrouver plus tard lorsque l'on essaye de relire le block une semaine plus tard.

Ensuite, il est utile d'utiliser des modules. C'est bien plus pratique pour s'y retrouver.

N'oubliez pas ces 2 flèches : elles permettent de revenir en arrière et en avant en cas de problèmes. De temps en temps cela peut sauver un certain temps de travail.



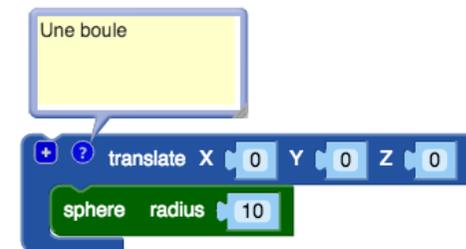
Une autre astuce : avec la souris du Mac, le zoom de l'écran des programmes change continuellement. Pour éviter cela, nous avons pris une vieille souris à molette.

Commentez vos programmes

Il est possible de mettre des commentaires dans tous les modules mais aussi sur les blocks. Un clic-droit sur une fonction et on peut faire « **Add Comment** ».

Voici en exemple, le commentaires dans une petite bulle jaune, accessible en cliquant sur le point d'interrogation.

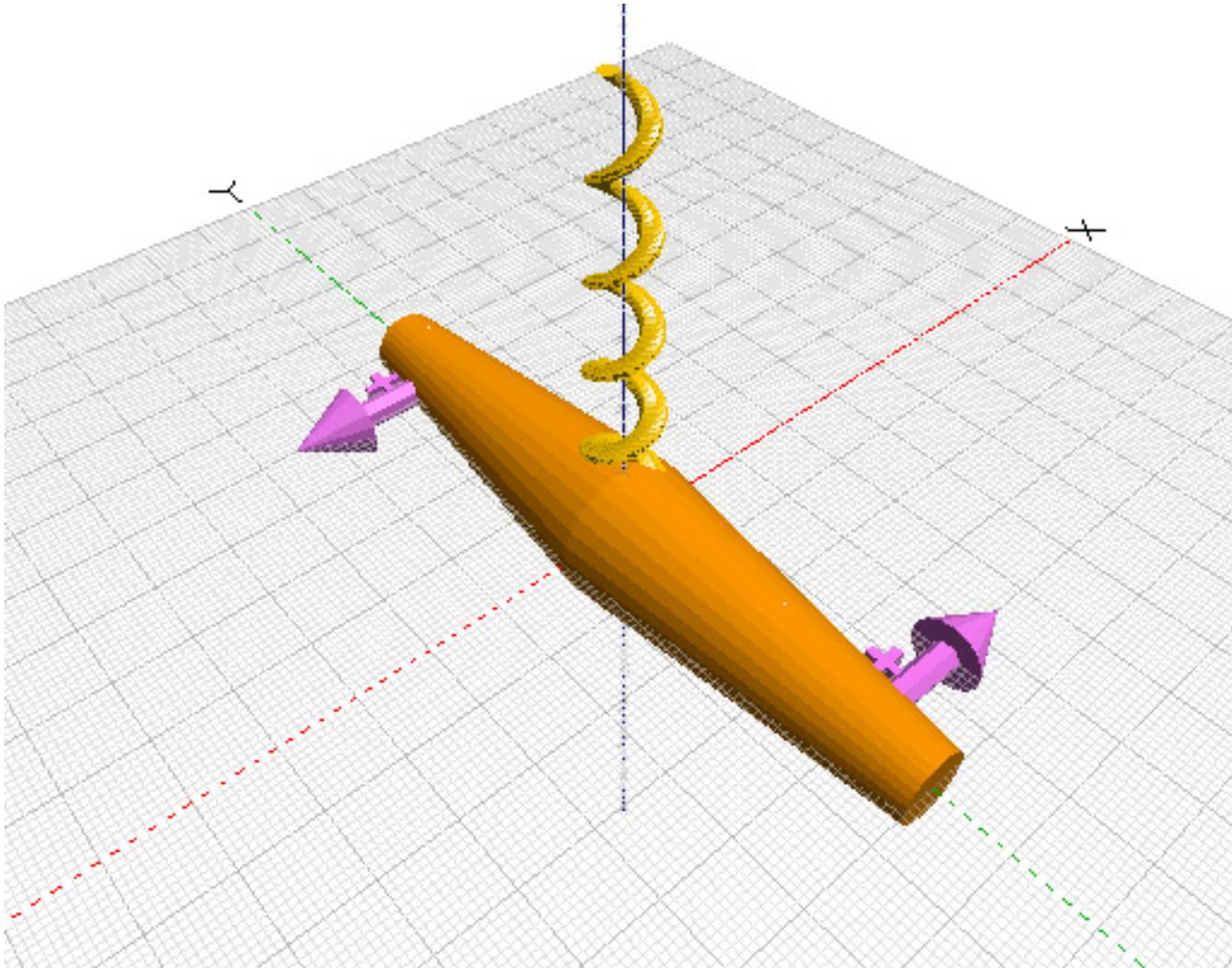
Une copie du block avec son commentaire, une fois l'option « Collapse Block » activée. Le début du commentaire apparaît ce qui permet une facilité de lecture du programme.



Une boule - translate X 0 Y 0 ...

Le tire bouchon de Maxwell

Puisque nous avons parlé de ce fameux tire bouchon, il devient opportun de le fabriquer plutôt que de prendre des images sur l'internet. Voici sa version terminée avec la vis, le manche et même des flèches pour indiquer le sens de rotation.



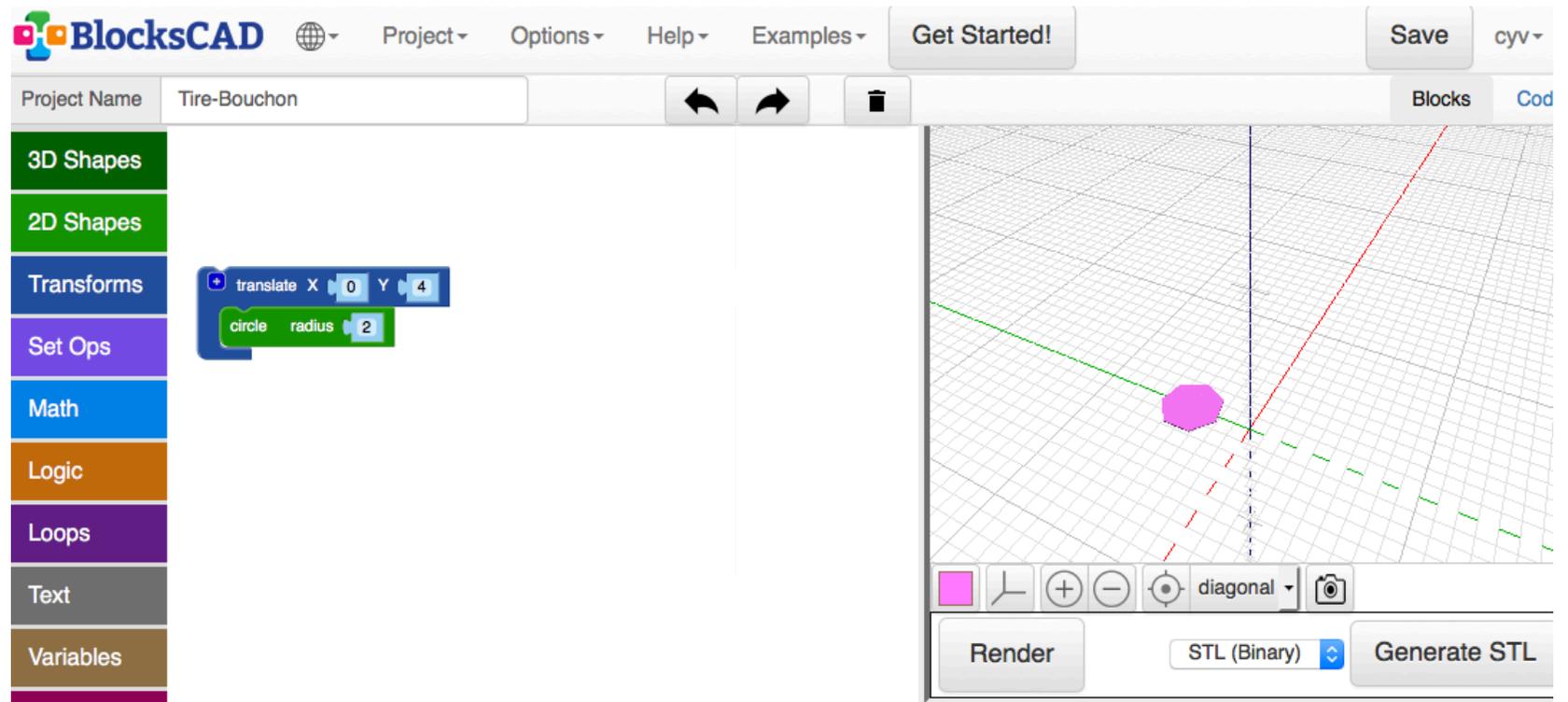
Dans les paragraphes suivants, nous allons détailler sa fabrication pas à pas, ou presque.

Réalisons un profil pour une extrusion

Pour expliquer ce que fait la transformation LINEAR EXTRUDE, commençons par fermer les yeux et se remémorer une glace à l'italienne, comme la photo ci-dessous. La machine à glace réalise une extrusion de la crème glacée en utilisant un profil, la forme que va prendre le ruban de crème.



Maintenant que nous avons une compréhension gustative de l'extrusion, commençons par faire un cercle légèrement excentré, c'est à dire décalé par rapport à l'origine. Pour cela nous utilisons dans le menu gauche 2D SHAPES, l'ordre CIRCLE. « 2D » signifie que c'est à plat dans le plan [X,Y] accompagné d'un TRANSLATE pour le décaler.



Cette forme plate va nous servir de profil pour l'extrusion. Notre profil n'est pas un cercle mais un heptagone car l'option SIDES n'a pas été utilisée.

La fonction EXTRUDE sur un profil 2D

Appliquons la transformation EXTRUDE et regardons ce qui se passe. Notre forme a été extrudée ce qui nous donne une barre de 60 de haut, puisque nous avons demandé 60 dans la case HEIGHT.

The screenshot displays the BlocksCAD interface. On the left, a sidebar lists various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The '2D Shapes' category is active, showing a block editor with three steps: a 'circle' block with a radius of 2, a 'translate' block with X=0 and Y=4, and a 'linear extrude' block with height=60, twist=0°, scale x=1, scale y=1, and 'not centered' selected. The main workspace shows a 3D grid with a purple cylindrical bar extruded from the origin. The bar's height is 60 units. The bottom right of the workspace contains a toolbar with a pink square, a camera icon, zoom in (+) and zoom out (-) buttons, a 'diagonal' view selector, and a camera icon. Below the toolbar are 'Render', 'STL (Binary)', and 'Generate STL' buttons.

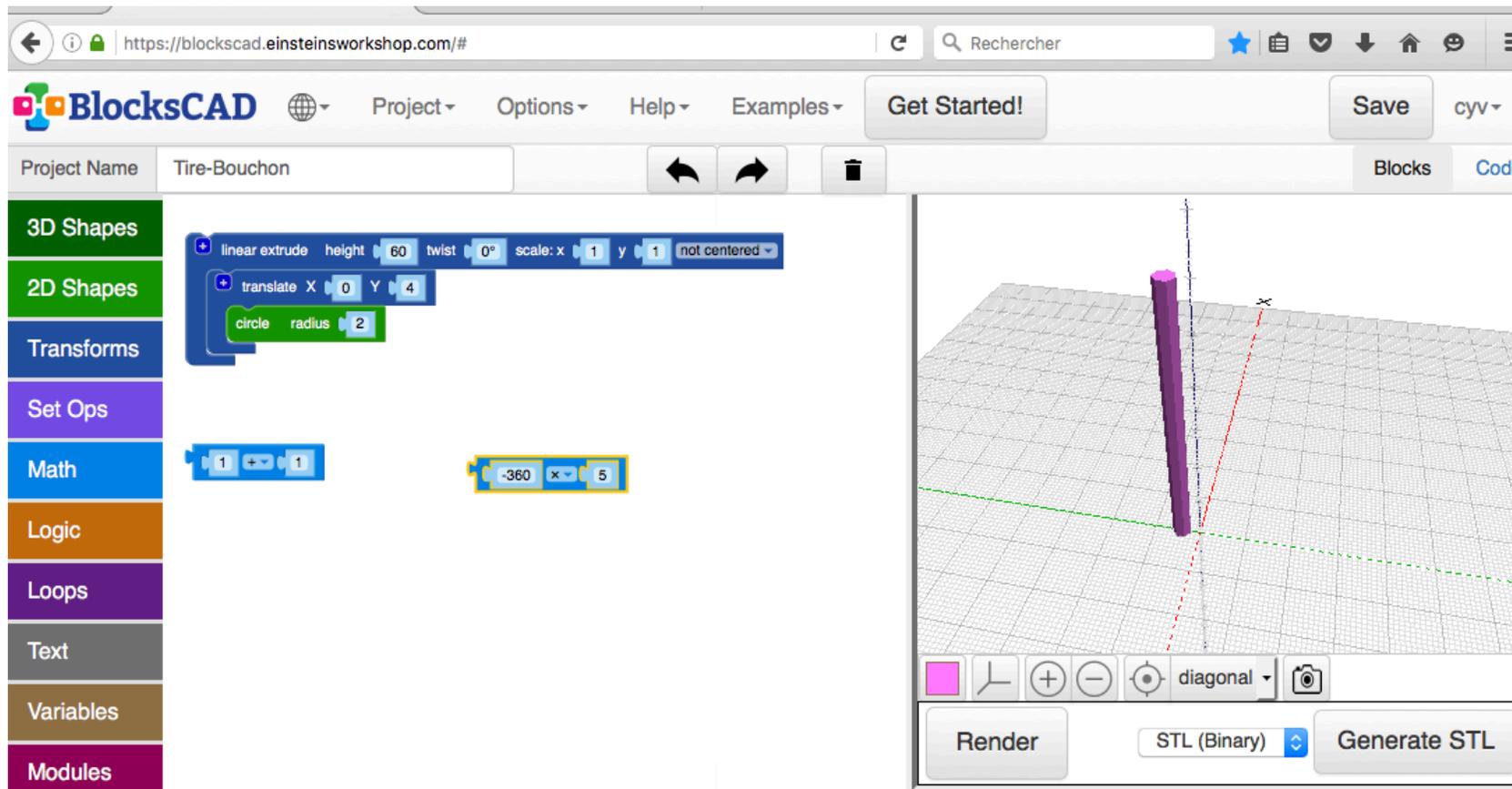
Au passage, indiquons que dans le monde de la métallurgie, l'extrusion est utilisée pour réaliser des barre avec de multiples profils différents en utilisant un métal à haute température.

L'option TWIST de la fonction EXTRUDE

La transformation LINEAR EXTRUDE comporte 2 options de plus. Commençons par TWIST. Cette option permet de faire tourner le profil en même temps qu'il monte. Exactement comme le glacier tourne le cornet pour faire une jolie forme.

Pour notre tire-bouchon, nous voulons 5 tours. Cette option ne fonctionne pas avec des « tours » mais avec des degrés. Les degrés sont calculés pour qu'un tour fasse 360° . Il faut donc $360 \times 5 = 1800$ degrés dans l'option TWIST. Hélas dans la version actuelle de BLOCKSCAD, impossible d'inscrire 1800 dans la case.

Pour éviter cet ennui, nous allons utiliser une formule présente dans le menu de gauche MATH : la formule 1+1.

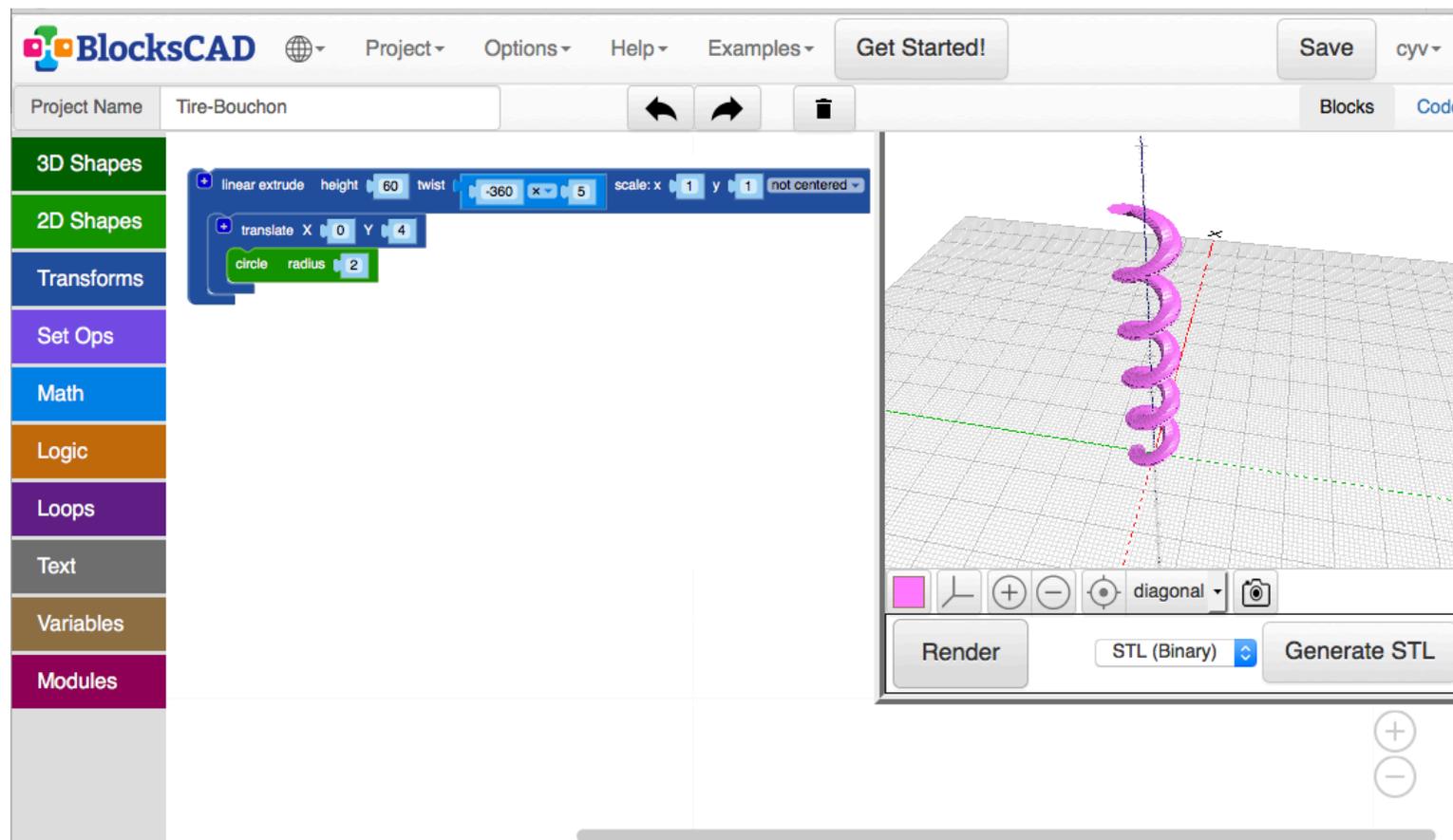


Notre première formule mathématique

Nous allons modifier cette formule :

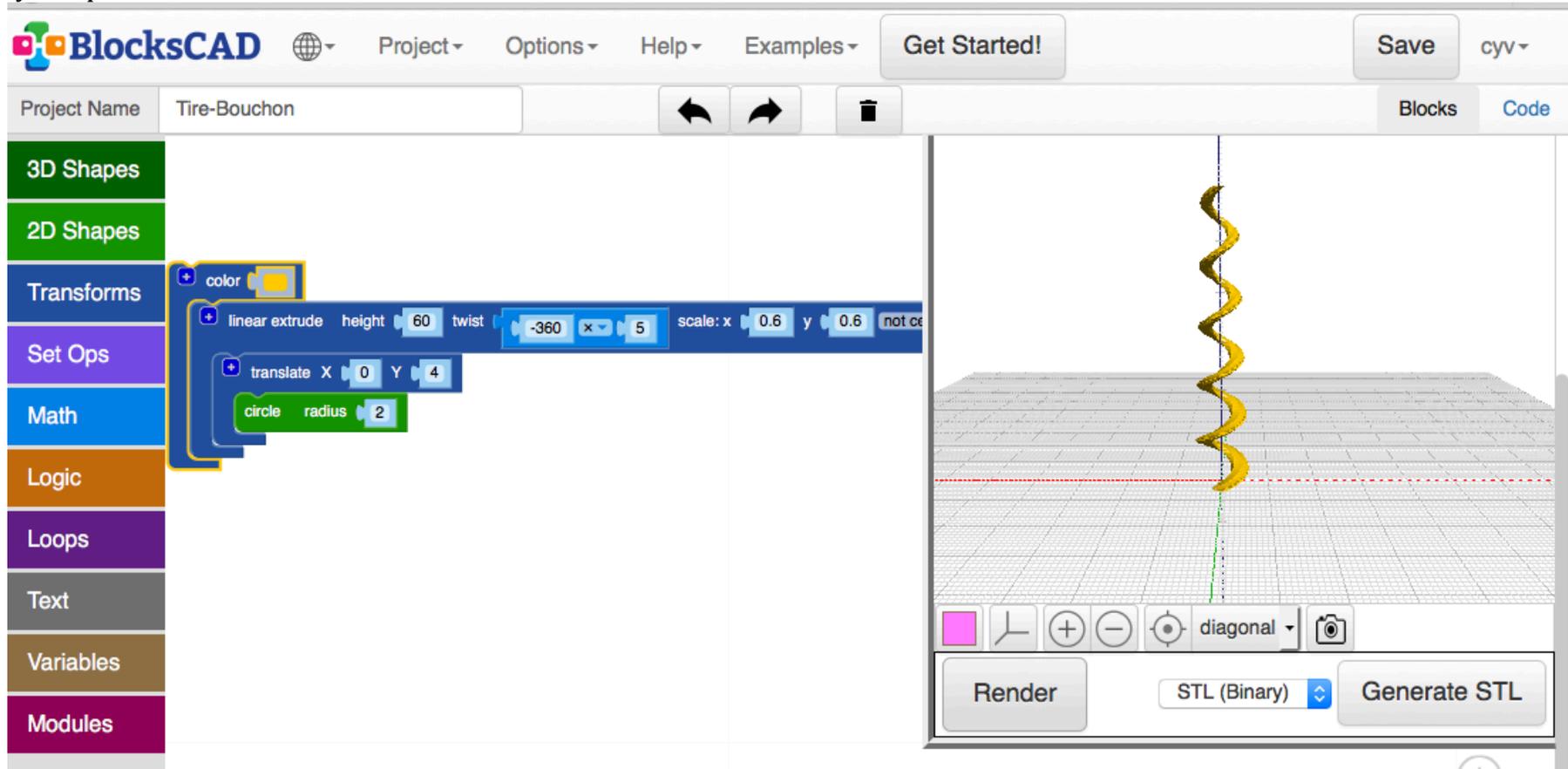
- changer le premier « 1 » en « 360 », c'est à dire un tour,
- changer le « + » en « x » pour avoir une multiplication,
- changer le « 1 » de droite en « 5 » pour ainsi avoir nos 5 tours.

Inutile de connaître le résultat, l'ordinateur le fera très bien tout seul. Ensuite on place cette formule dans l'option TWIST et notre vis commence à prendre forme.



L'option SCALE de la fonction EXTRUDE

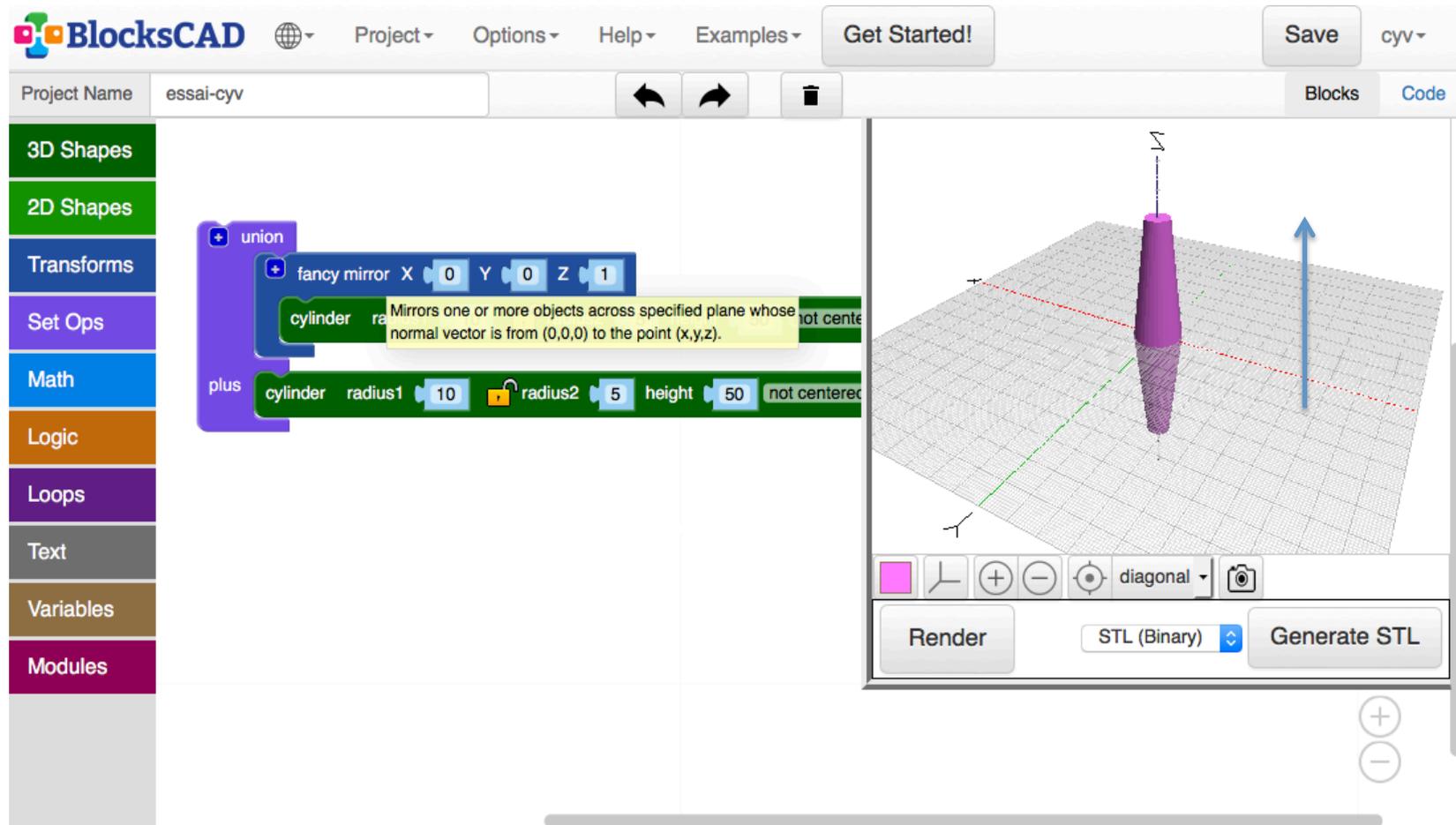
L'option SCALE, c'est à dire échelle en anglais, va nous permettre de changer les proportions selon l'axe X ou selon l'axe Y ou les 2. Plus la valeur sera petite plus la proportion suivant cet axe sera faible. Nous allons indiquer 0.6 mais à vous faire quelques essais en changeant les valeurs de SCALE y compris de voir avec les valeurs « 0 ».



Et voici notre vis terminée en trois lignes de programmation et quelques lignes d'explications.

Un manche conique avec FANCY MIRROR

Souvent appelé « Balto » car c'est l'enseigne de nombreux cafés, le manche est constitué de cônes accolés. Le programme comprend un cône réalisé avec la forme CYLINDER et son symétrique avec la transformation FANCY MIRROR qui simule un grand miroir plan, qui dans notre exemple est le plan XY.



Cette transformation se fait en indiquant dans les cases [X,Y,Z] le vecteur normal (normal vector en anglais). Pour faire simple, le vecteur normal d'un plan est la droite qui est orthogonale, c'est à dire forme un angle droit, avec ce plan. Le vecteur normal au plan XY est indiqué par la flèche bleue : c'est l'axe donc : X=0, Y= 0, Z=1.

FANCY MIRROR ou MIROR

Nous pouvons aussi utiliser la Transformation MIRROR qui est plus simple quand il s'agit du plans horizontal [XY] ou d'un des deux plans verticaux [XZ] ou [ZY]. En effet, on indique dans l'option simplement le plan qui représente le miroir. Regardez les blocks ci-dessous, ils sont équivalents.

The screenshot displays the BlocksCAD interface with two different block configurations for creating a mirrored cylinder. The left configuration uses the 'FANCY MIRROR' block, and the right configuration uses the 'MIRROR ACROSS' block. Both configurations use a 'plus' block to combine two cylinders. The 3D preview window shows a purple cylinder and its yellow mirror image on a grid.

Left Configuration (FANCY MIRROR):

- union
 - fancy mirror X: 0 Y: 0 Z: 1
 - cylinder radius1: 10 radius2: 5 height: 50 not centered
 - plus
 - cylinder radius1: 10 radius2: 5 height: 50 not centered
- color (yellow)
- translate X: 0 Y: 50 Z: 0
 - union
 - mirror across XY
 - cylinder radius1: 10 radius2: 5 height: 50 not centered
 - plus
 - cylinder radius1: 10 radius2: 5 height: 50 not centered

Right Configuration (MIRROR ACROSS):

- union
 - mirror across XY
 - cylinder radius1: 10 radius2: 5 height: 50 not centered
 - plus
 - cylinder radius1: 10 radius2: 5 height: 50 not centered
- color (yellow)
- translate X: 0 Y: 50 Z: 0
 - union
 - mirror across XY
 - cylinder radius1: 10 radius2: 5 height: 50 not centered
 - plus
 - cylinder radius1: 10 radius2: 5 height: 50 not centered

Nous avons présenté FANCY MIRROR car il permet beaucoup plus de possibilité en représentant des miroirs obliques en compléments des verticaux et horizontaux.

La poignée du tire-bouchon

Il suffit maintenant de mettre le manche à l'horizontal et de le colorer pour la suite.

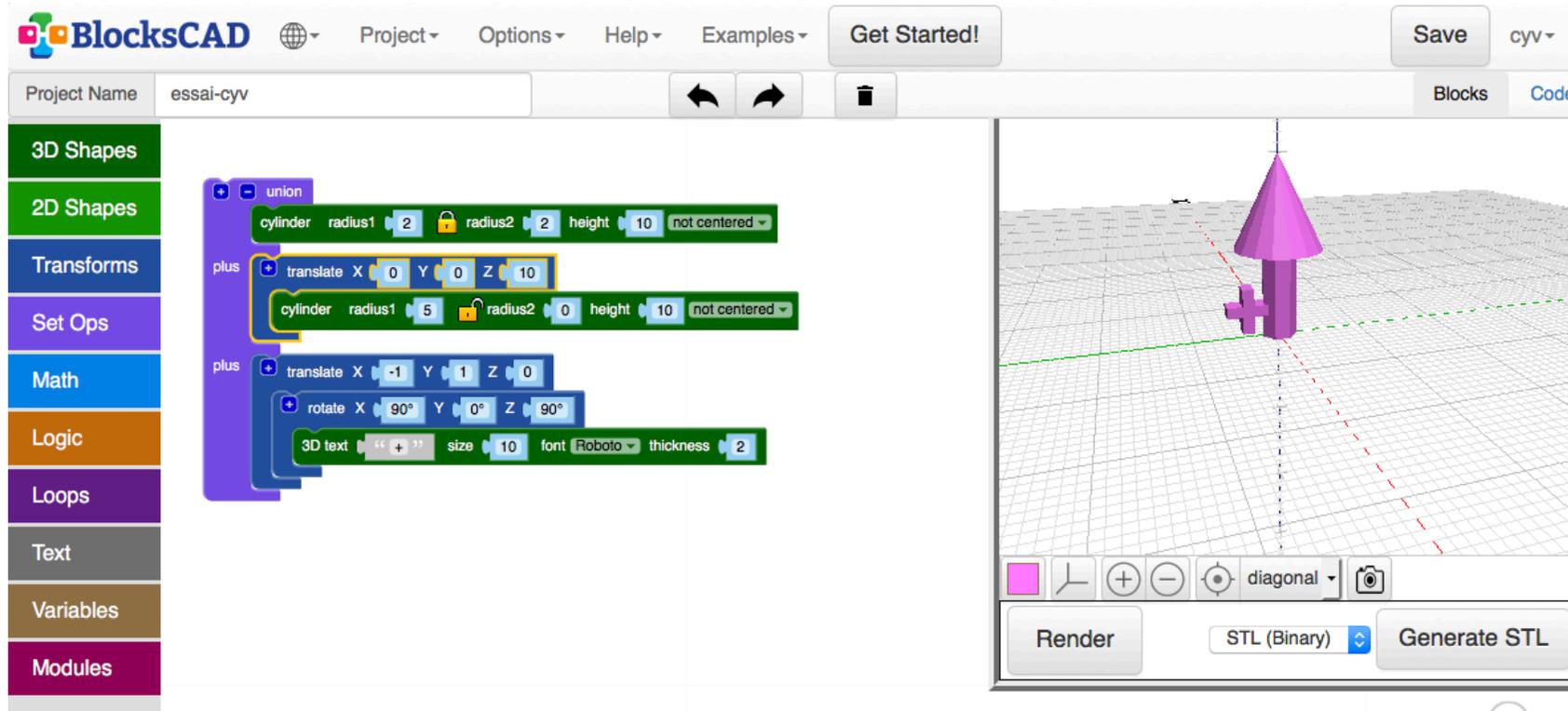
The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', 'Examples', and 'Get Started!'. The project name is 'essai-cyv'. The left sidebar contains a list of categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D view of a yellow, tapered handle on a grid. The code blocks are as follows:

- color** block with a yellow color swatch.
- rotate** block with X: 90°, Y: 90°, Z: 0°.
- union** block containing:
 - fancy mirror** block with X: 0, Y: 0, Z: 1.
 - cylinder** block with radius1: 10, radius2: 5, height: 50, and 'not centered' selected.
- plus** block containing:
 - cylinder** block with radius1: 10, radius2: 5, height: 50, and 'not centered' selected.

The 3D view includes a coordinate system with X, Y, and Z axes. Below the 3D view is a toolbar with a color selection tool, navigation controls (pan, zoom, rotate), and a 'diagonal' view selector. At the bottom of the interface are buttons for 'Render', 'STL (Binary)', and 'Generate STL'.

Une flèche positive avec 3D TEXT

Maintenant nous devons dessiner une flèche avec le signe « + » pour bien indiquer dans quel sens doit tourner le tire-bouchon.



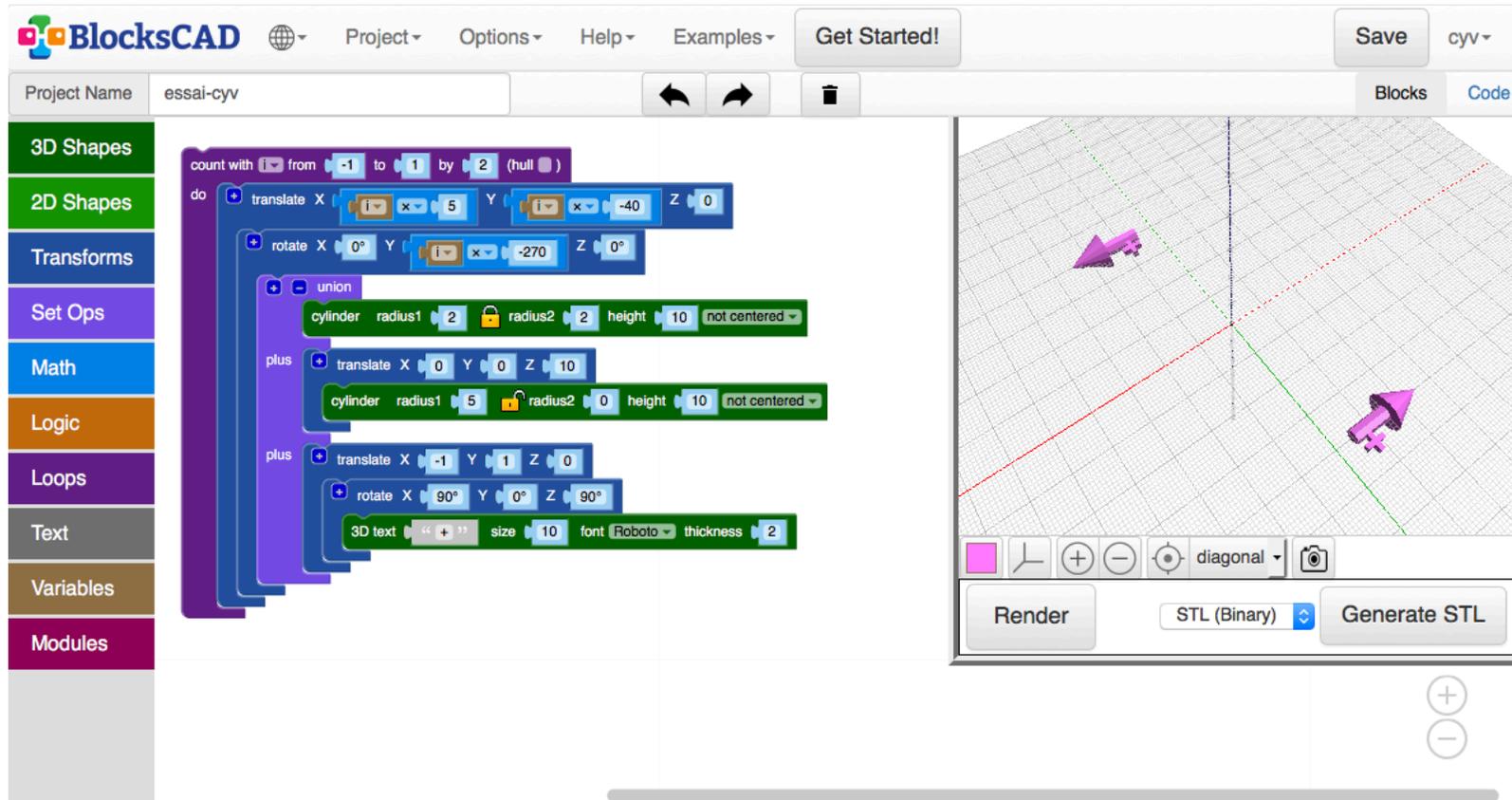
Le programme comprend déjà :

- un cylindre,
- un cylindre de forme conique, décalé (TRANSLATE) vers le haut (axe Z) pour couvrir le cylindre précédent.

Dans le troisième bloc, ce programme comprend une nouvelle fonction issue du menu gauche TEXT. Cette fonction permet d'écrire un texte en choisissant la taille (SIZE), la police de caractère (FONT), l'épaisseur (THICKNESS). Dans la première partie de cette fonction nous avons mis le texte « + ».

Enfin, nous rajoutons un TRANSLATE et un ROTATE pour bien positionner cet objet « + » avec la flèche conique.

Pour terminer, nous devons créer deux flèches et bien les positionner.



Pour cela nous avons utilisé une boucle informatique qui va faire 2 passages : $i=-1$ puis $i=1$.

Ensuite nous utilisons un ordre ROTATE qui va mettre les flèches à l'horizontal (il suffit de 90° pour passer de la vertical à l'horizontal) et l'ordre TRANSLATE avec une formule pour décaler de la longueur du manche.

Le programme complet du tire-bouchon de Maxwell

Et donc pour finir, le programme complet avec la vis, le manche et les flèches.

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', 'Examples', and 'Get Started!'. The project name is 'Tire-Bouchon'. The left sidebar contains a category list: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace is divided into two panels. The left panel shows a block-based programming tree for a 3D model. The right panel shows a 3D view of the model, which is a screwdriver with a yellow handle and a silver shaft. The handle is composed of several parts: a top section with a yellow spiral, a middle section with a yellow cylinder, and a bottom section with a yellow cylinder and a yellow circle. The shaft is a long yellow cylinder. The programming blocks are color-coded: purple for union, blue for translate, rotate, and linear extrude, green for cylinder, and yellow for color. The 3D view includes a grid, a camera icon, and a 'Render' button. The bottom right corner of the 3D view has a 'Generate STL' button with a dropdown menu set to 'STL (Binary)'.

Dégustons une glace cassis-vanille

Ci dessous, le programme utilisant deux fois la fonction LINEAR EXTRUDE :

- une première fois pour faire un cône vrillé (SCALE à zéro pour terminer en pointe),
- une deuxième fois en conservant l'échelle pour avoir une glace plus grosse à manger 😊.

Complétons avec un cône (CYLINDER) que nous renversons avec la fonction MIRROR. Et si vous n'aimez pas le cassis, utilisez la transformation COLOR pour changer de parfum. Bonne dégustation !

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', 'Examples', and 'Get Started!'. The project name is 'glace à l'italienne'. The left sidebar shows a library of blocks categorized by '3D Shapes', '2D Shapes', 'Transforms', 'Set Ops', 'Math', 'Logic', 'Loops', 'Text', 'Variables', and 'Modules'. The main workspace contains a 3D model of a swirl ice cream cone with a purple swirl and a yellow cone. The code blocks are as follows:

- union** block containing:
 - translate** block: X=0, Y=0, Z=20
 - linear extrude** block: height=30, twist=180°, scale: x=0, y=0, not centered
 - count with** block: from=1, to=6, by=1, (hull)
 - do** loop containing:
 - rotate** block: X=0°, Y=0°, Z=60°
 - square** block: X=10, Y=10, not centered
- plus** block containing:
 - color** block: yellow
 - linear extrude** block: height=20, twist=180°, scale: x=1, y=1, not centered
 - count with** block: from=1, to=6, by=1, (hull)
 - do** loop containing:
 - rotate** block: X=0°, Y=0°, Z=60°
 - square** block: X=10, Y=10, not centered
- color** block: yellow
- mirror across** block: XY
- cylinder** block: radius1=13, radius2=2, height=30, not centered

The bottom right of the interface shows a 'Render' button and a 'Generate' button with a dropdown menu set to 'STL (Binary)'. The 'Smooth' options are 'Low', 'Medium' (selected), and 'High'.

Retour sur la fonction MIROR

La fonction MIROR comporte 3 options : XY, XZ, YZ. Vous l'avez surement reconnu, ces options reprennent les lettres des axes. En fait, ces options décrivent les 3 plans :

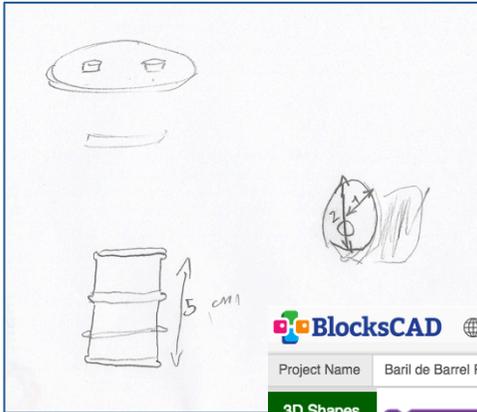
- XY pour le plan qui contient l'axe X et l'axe Y, plan qui est orthogonal à l'axe Z,
- XZ pour le plan qui contient l'axe X et l'axe Z, plan qui est orthogonal à l'axe Y,
- YZ pour le plan qui contient l'axe Y et l'axe Z, plan qui est orthogonal à l'axe X.

Imaginez que nous posions un miroir, la plaque grise, sur le plan YZ. L'option MIROR-YZ réalise l'objet symétrique par rapport à ce miroir imaginaire. Le même principe pour les autres plans.

The screenshot displays the BlocksCAD software interface. On the left, a vertical sidebar lists various block categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace is filled with a sequence of blocks: a 'translate' block (X: 60, Y: 0, Z: 0) followed by a 'sphere' block (radius: 10). Below these, a 'color' block (yellow) is followed by a 'mirror across' block (set to YZ). This is followed by another 'translate' block (X: 40, Y: 0, Z: 0) and a second 'sphere' block (radius: 10). At the bottom, a 'color' block (grey) is followed by a 'cube' block (X: 1, Y: 100, Z: 100, centered). The right side of the interface shows a 3D view of the model. A vertical grey plane (the mirror) is positioned in the YZ plane. A yellow sphere is located to the left of the plane, and a purple sphere is its mirror image to the right. A red dashed line indicates the X-axis. The bottom right corner features a rendering toolbar with options for 'Smooth' (Low, Medium, High) and a 'Generate' button.

Un Baril pour Barrel Race

Proposé par Xavière, un baril utilisé dans les Barrel Race, épreuve pour chevaux d'origine western. Pour ceux qui n'auraient pas bien vu le rapport, 90% des objets que nous réalisons servent de décors aux innombrables figurines de chevaux de Xavière et Quitrie.



Pour commencer, le plan initial de Xavière.

Effectivement, il faut se projeter un peu, mais bon cela peut suffire, le vrai plan sera le programme BlocksCAD.

Ce baril va vous faire réviser les LOOP, les TEXT les cylindres et les tores. La solution vous est fournie par Xavière. Les dimensions sont compatibles avec les figurines de chevaux de 2 grandes marques.

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', 'Examples', and 'Get Started!'. The project name is 'Baril de Barrel Race'. The left sidebar shows various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D model of a blue barrel with the text 'Barrel Race' on its top. The code editor on the left contains the following blocks:

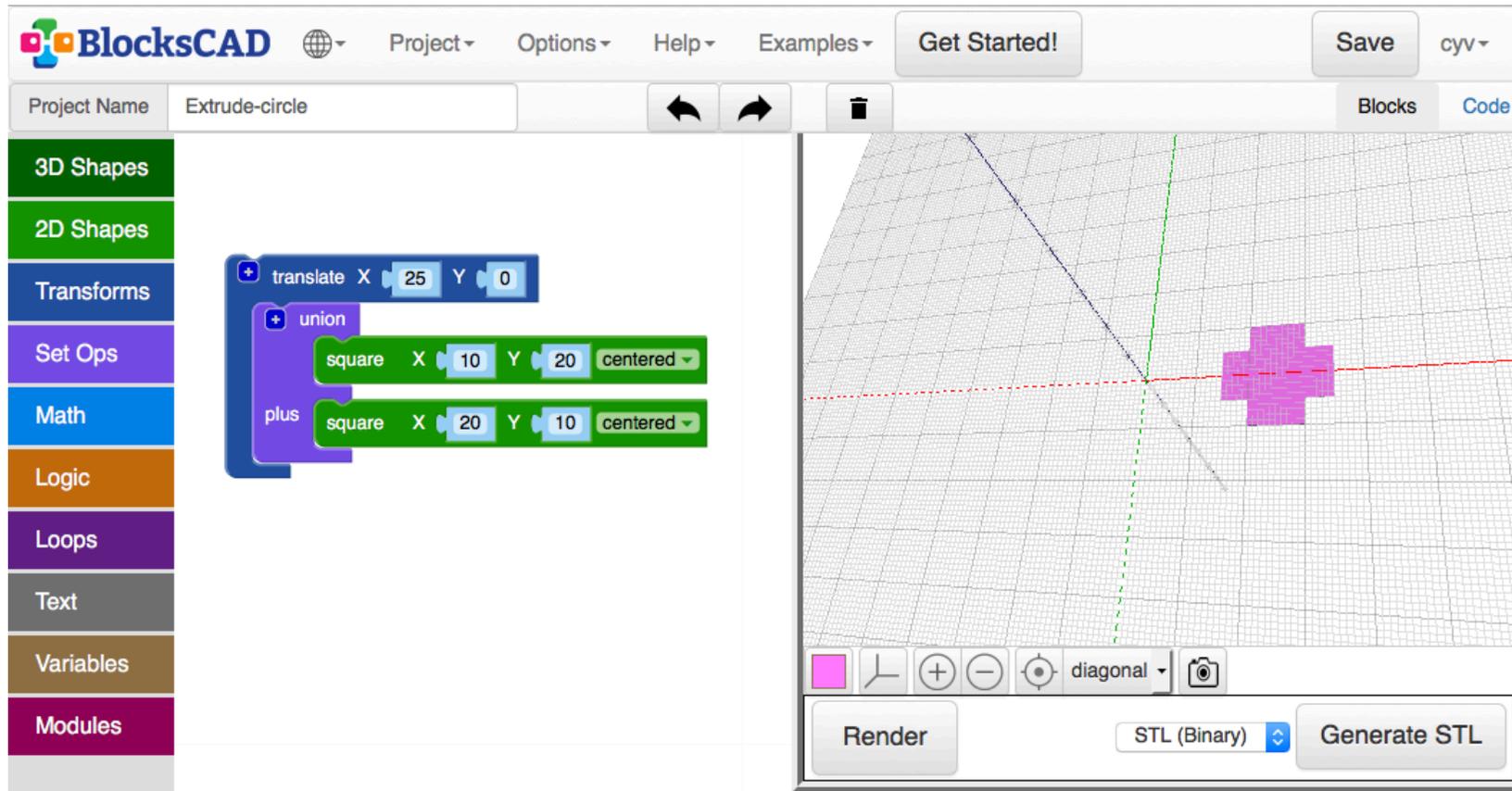
- count with** (Loop): from 0 to 50 by 16.66 (hull)
- do** (Loop):
 - translate X** (0) Y (0) Z (0)
 - torus** (radius1: 15, radius2: 1, sides: 100, faces: 30)
- color** (red)
- translate X** (9) Y (0) Z (50)
- cylinder** (radius1: 3, radius2: 3, height: 1, not centered)
- sides** (100)
- difference** (Loop):
 - cylinder** (radius1: 15, radius2: 15, height: 50, not centered)
 - minus** (Loop):
 - cylinder** (radius1: 12, radius2: 12, height: 40, not centered)
 - translate X** (0) Y (-13) Z (0)
 - rotate X** (0°) Y (0°) Z (90°)
 - color** (red)
 - translate X** (0) Y (0) Z (49)
 - 3D text** (" Barrel Race ", size: 4, font: Roboto, thickness: 2)

The 3D view shows the barrel on a grid. The bottom right of the interface includes a 'Render' button, a 'Smooth' dropdown (Low, Medium, High), and a 'Generate STL' button.

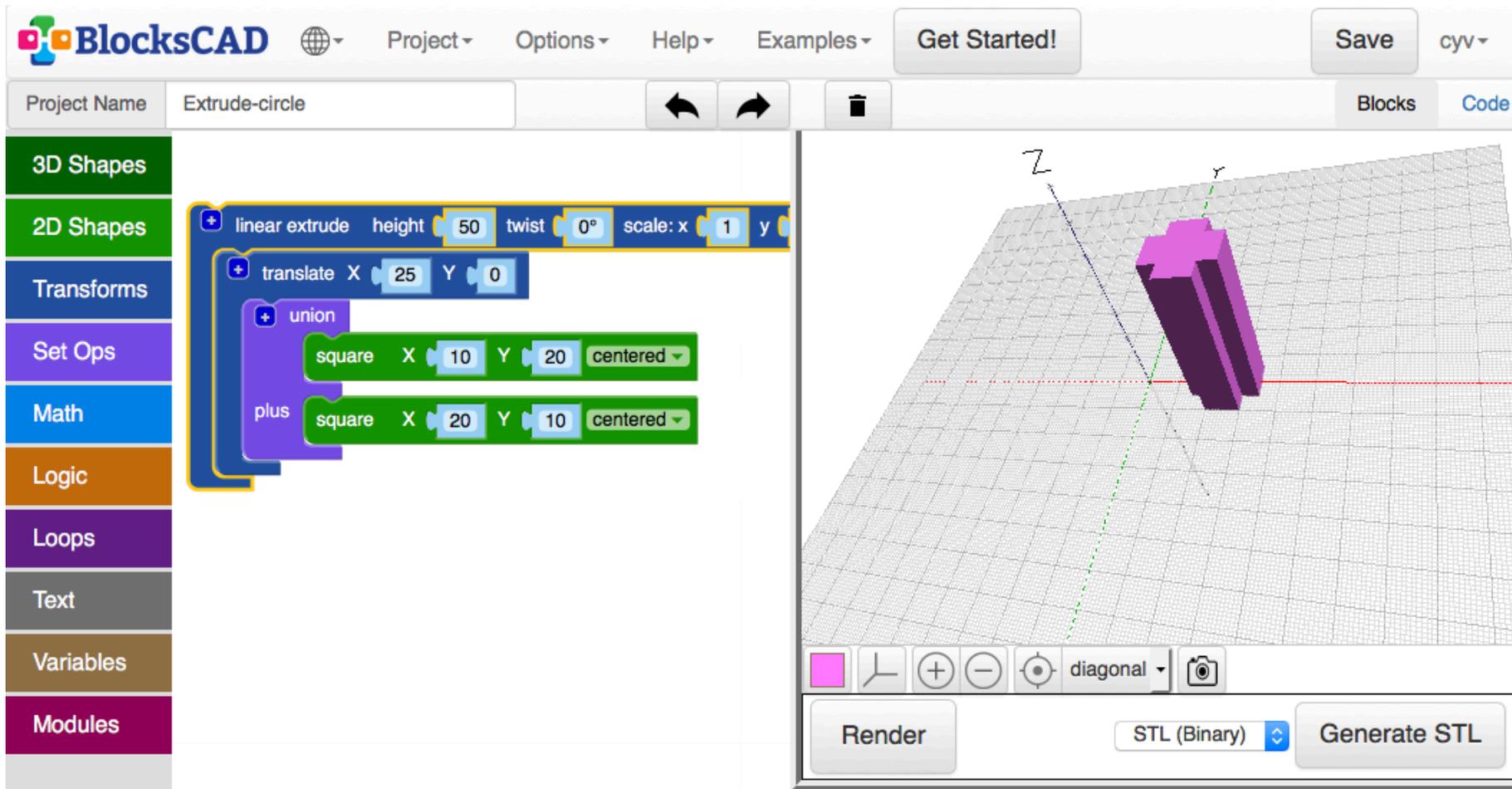
Profils pour les fonctions LINEAR EXTRUDE et ROTATE EXTRUDE

ROTATE EXTRUDE

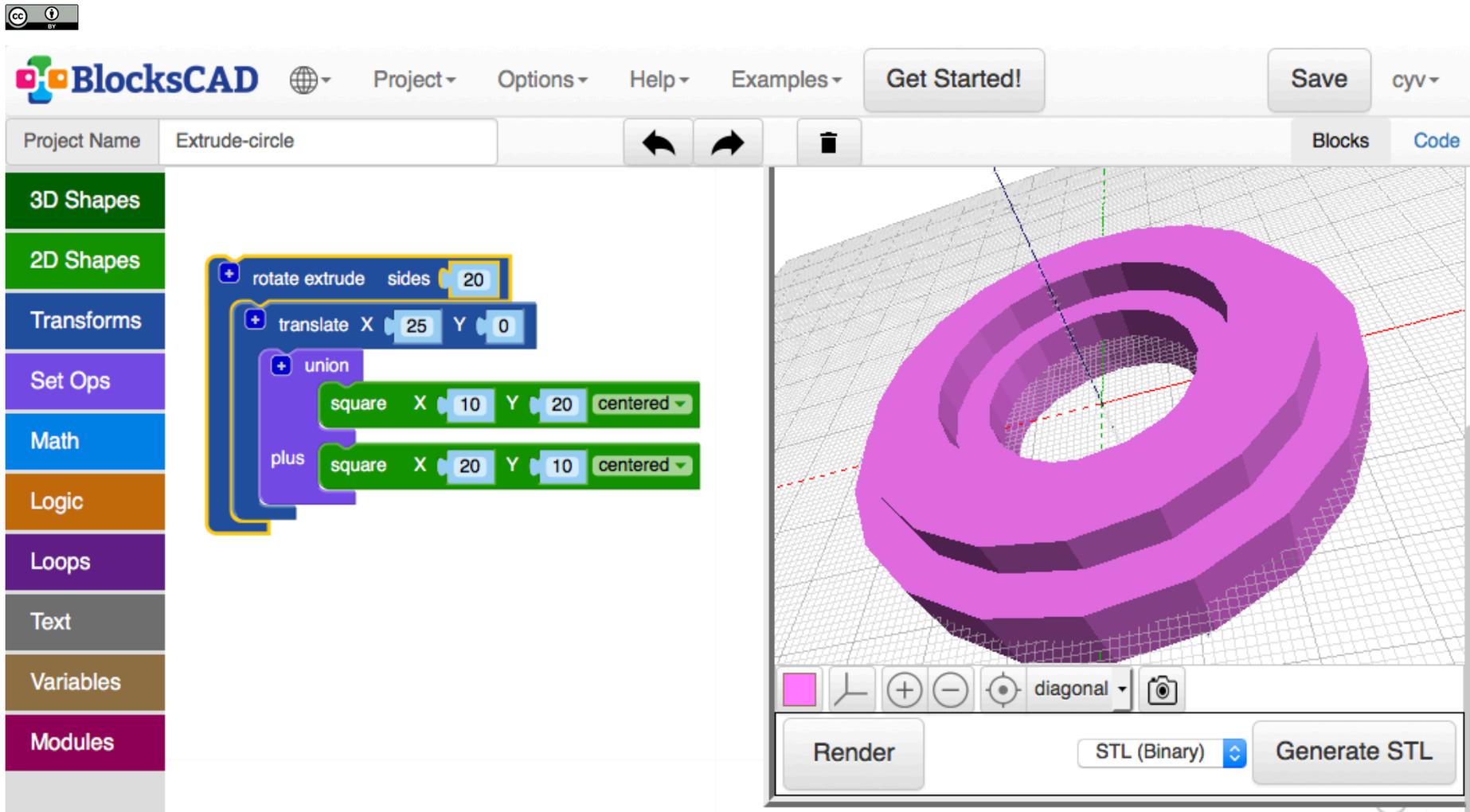
Réalisons un petit profil simple en utilisant deux rectangles avec la fonction SQUARE.



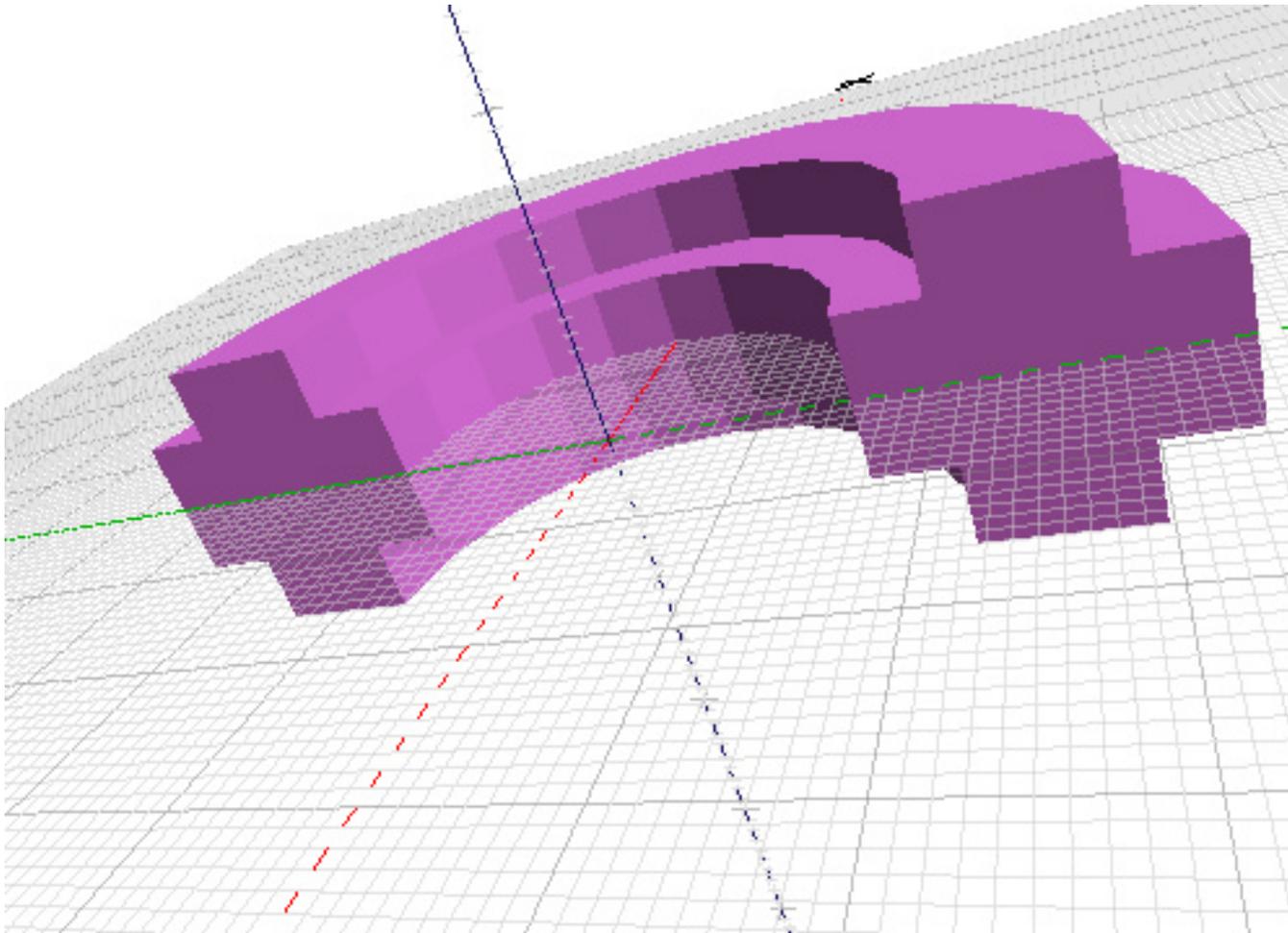
Nous avons déjà vu la transformation LINEAR EXTRUDE qui nous permet de réaliser des barres plus ou moins vrillées sur la base d'un profil initial. Et voici ce que cela donne.



Remplaçons la transformation LINEAR EXTRUDE par ROTATE EXTRUDE.



Cette transformation prend le profil, le remet vertical et le faire tourner autour de l'axe Z.
Ce que nous voyons plus facilement sur le dessin suivant.



Bien sûr, me direz vous, le profil dessiné était horizontal et pourtant l'objet à un profil vertical. Eh bien c'est comme ça ! Cela à l'avantage de dessiner normalement sur le plan $[X,Y]$ avec les fonctions de 2D SHAPES et de ne pas de soucier du reste.

L'option SIDE de ROTATE-EXTRUDE

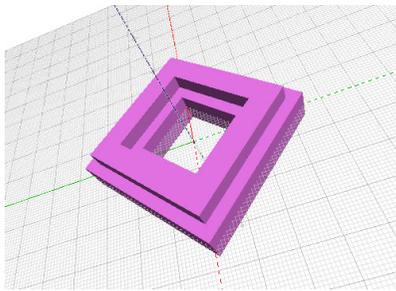
Cette option permet d'indiquer le nombre de cotés, au minimum 3.

The screenshot displays the BlocksCAD software interface. The top navigation bar includes the logo, a globe icon, and menu items: Project, Options, Help, Examples, Get Started!, Save, and cyv. Below this, the Project Name is set to "Extrude-circle". The left sidebar contains a vertical menu with categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace is divided into two panels. The left panel shows a block-based code editor with the following structure:

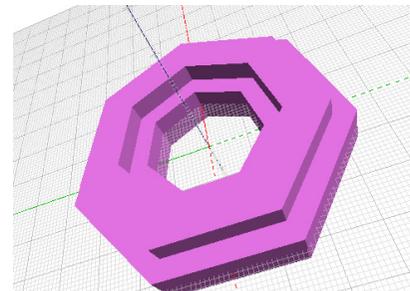
- rotate extrude sides 3
- translate X 25 Y 0
- union
 - square X 10 Y 20 centered
 - plus
 - square X 20 Y 10 centered

The right panel shows a 3D perspective view of a purple triangular prism (a triangular extrusion) on a grid. The prism is oriented diagonally. Below the 3D view is a toolbar with icons for color selection, rotation, zoom in (+), zoom out (-), and a dropdown menu currently set to "diagonal". At the bottom of the interface, there are buttons for "Render", "STL (Binary)", and "Generate STL".

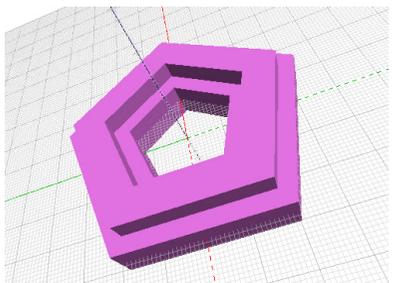
Avec l'option égale à 3, cela nous donne un triangle.



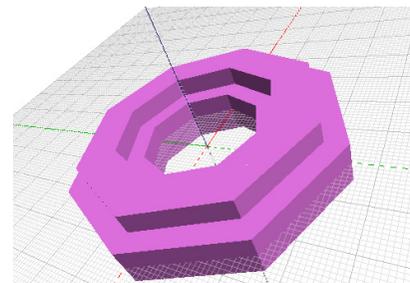
Avec, l'option égale à 4 c'est un carré.



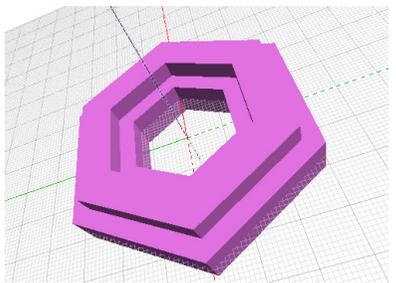
Avec, l'option égale à 7 c'est un heptagone



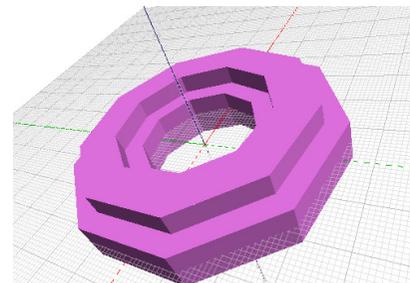
Avec, l'option égale à 5 c'est un pentagone



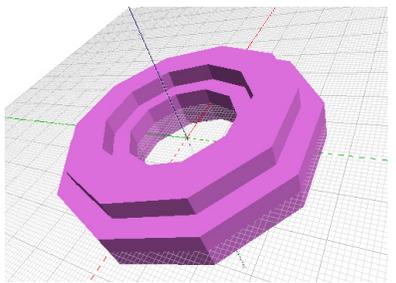
Avec, l'option égale à 8 c'est un octogone



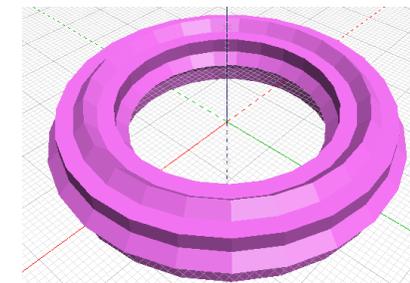
Avec, l'option égale à 6 c'est un hexagone



Avec, l'option égale à 9 c'est un ennéagone



Avec, l'option égale à 10 c'est un dodécagone



Et ainsi de suite dans la galaxie de truc-gone. Et si vous voulez que cela ressemble à une forme circulaire, il faut mettre l'option à 25 ou plus

Les profils

Avec LINEAR EXTRUDE et ROTATE EXTRUDE, nous avons donc des transformations qui nous permettent de faire une infinité de formes, respectivement linéaire ou circulaire, à base de profils.

Vous pouvez explorer comment faire des profils avec les 3 fonctions de 2D SHAPES : SQUARE et CIRCLE et 2D TEXT. Cette dernière fonction se trouve dans le menu de gauche TEXT.

En utilisant ces trois fonctions, avec les transformations (MENU TRANSFORMS) et les opérations (SET OPS) nous pouvons créer d'ores et déjà de nombreux profils.

The screenshot displays the BlocksCAD software interface. On the left, a sidebar lists various function categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 2D grid with a pink circular profile containing several radial lines. The word "Profil" is written in pink text on the grid. The interface includes a top menu bar with "Project", "Options", "Help", and "Examples" menus, a "Get Started!" button, and a "Save" button. The bottom of the workspace features a toolbar with icons for selection, zooming, and a "diagonal" grid option, along with a "Smooth" setting set to "Medium".

The code blocks visible in the workspace are as follows:

- difference** block containing:
 - scale** block (X: 1, Y: 1)
 - circle** block (radius: 10)
- minus** block containing:
 - count with** block (from: 1, to: 5, by: 1, hull:)
 - do** loop containing:
 - translate** block (X: 0, Y: 0)
 - rotate** block (X: 0°, Y: 0°, Z: 30)
 - square** block (X: 1, Y: 15, centered)
- translate** block (X: 20, Y: 0)
- 2D text** block (text: "Profil", size: 10, font: Roboto)

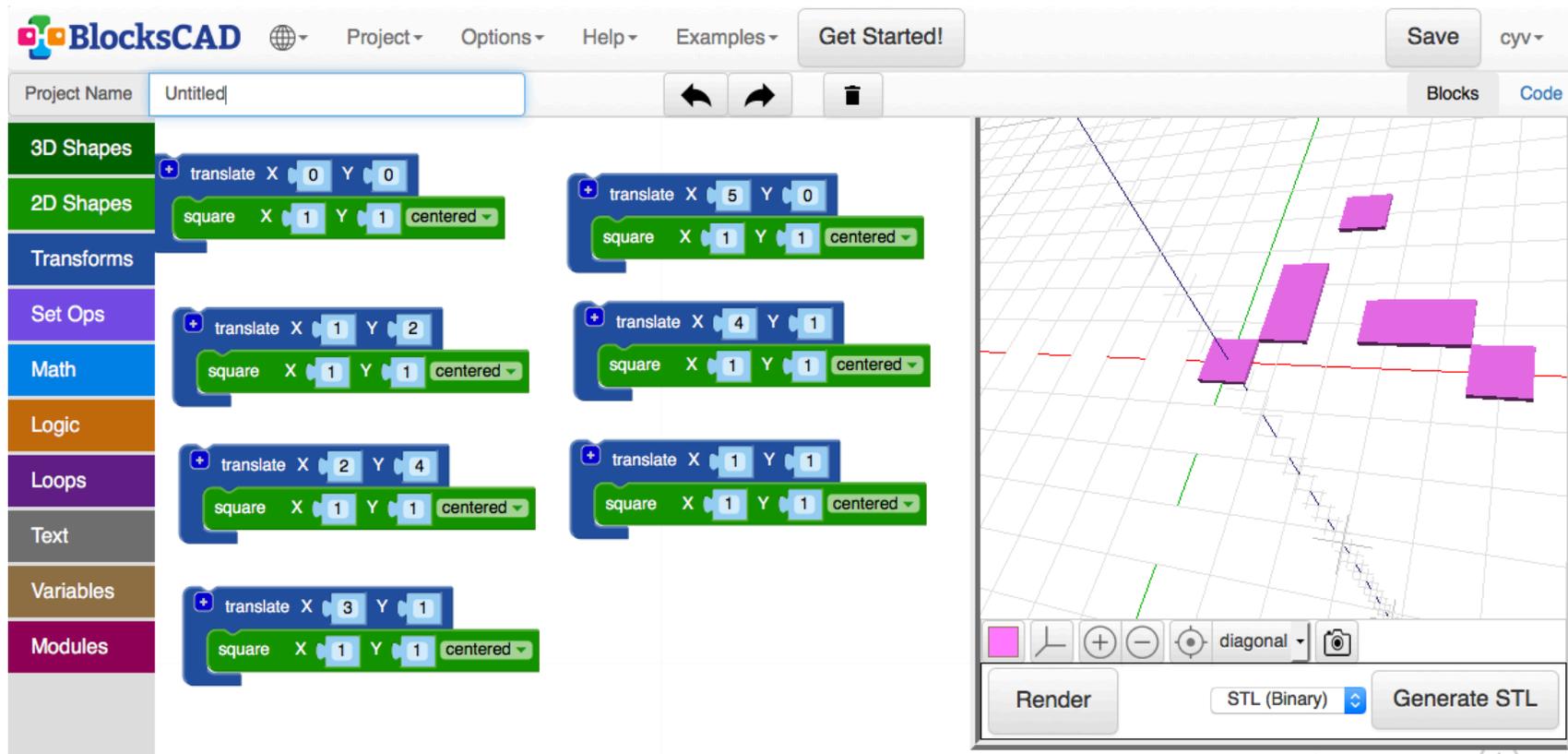
Courbes et comment les dessiner pour en faire des profils

Premiers pas

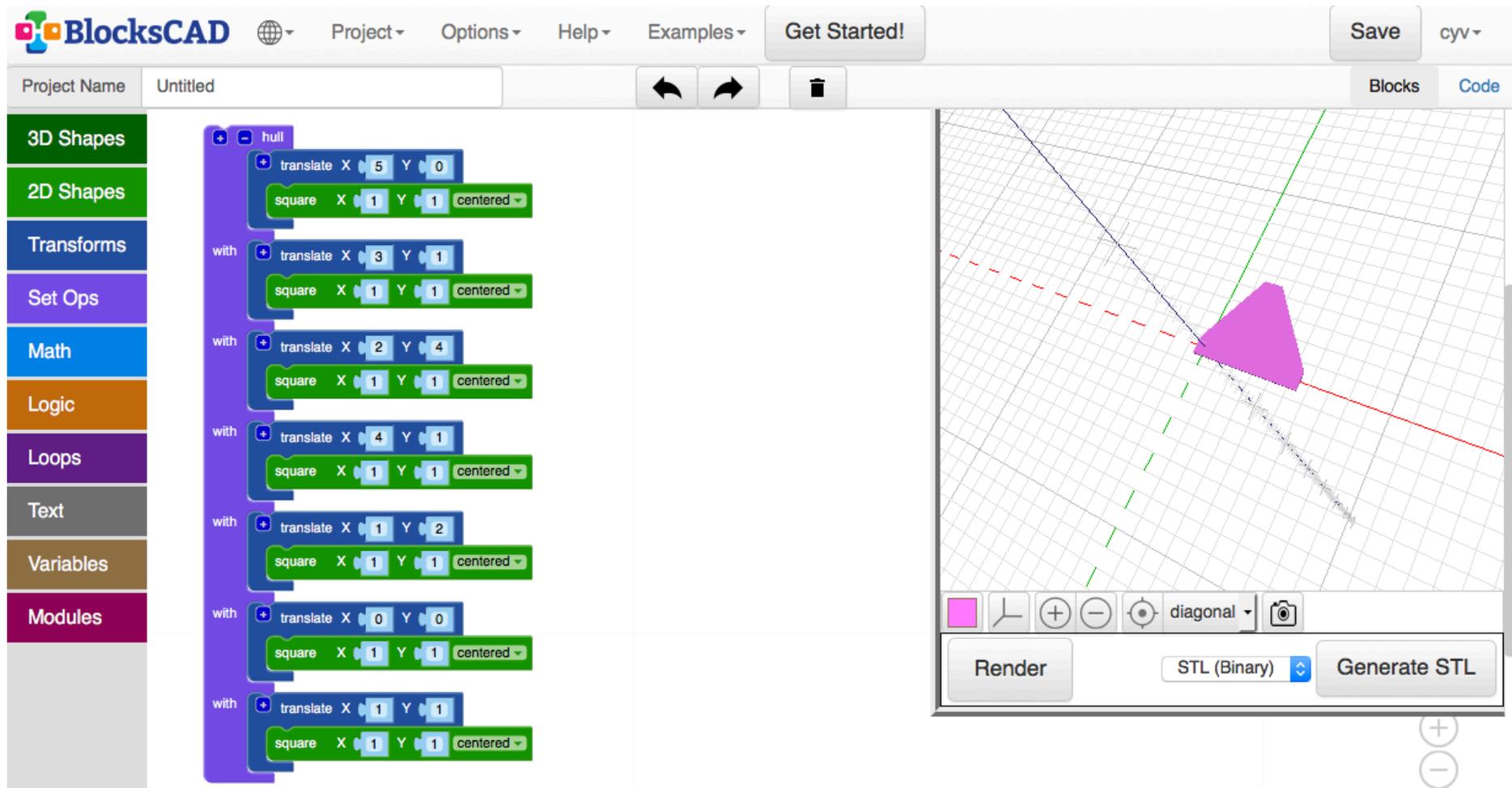
Cependant, beaucoup de profils ne sont pas compliqués à faire avec SQUARE et CIRCLE.

En fait, dessiner un profil revient à réaliser une courbe, mot souvent utilisé par les mathématiciens et les physiciens : une courbe c'est un ensemble de points contigus, c'est à dire très proches.

On peut toujours dessiner une courbe avec BlocksCAD en donnant tous les points à la main.



Non seulement, le résultat est nul, mais en plus cela prend du temps et puis si on veut une vraie courbe (venant d'un dessin) il va falloir perdre des heures à tout mesurer.



En regroupant tous ces points dans une coque (HULL), le résultat est ... moins pire car le profil se dessine un peu mais il n'est pas imaginable de dessiner de jolis profils rapidement avec cette méthode.

Les fonctions mathématiques

Nous ne sommes pas les premiers à chercher comment dessiner facilement une courbe. Les mathématiciens s'y emploient depuis des millénaires à commencer par les Grecs. Des outils ont été inventés pour cela : la règle pour une courbe en ligne droite et le compas pour une courbe en cercle. Ce qui correspond à nos deux fonctions SQUARE (une règle à deux dimensions) et CIRCLE.

Ensuite, les mathématiciens ont imaginé deux sortes d'outils :

- Les opérations, addition, soustraction, multiplication, division et puissance,
- Les fonctions.

Les fonctions sont utilisées pour exprimer qu'un nombre, appelons le Y, est le résultat d'un calcul utilisant un autre nombre, appelons le X. Ce calcul peut être très simple ou ... très compliqué. On note souvent le nombre $Y = F(X)$: F étant la fonction. On peut dnc représenter la fonction par un calcul, celui qui donne F(X).

On peut aussi représenter une fonction F par sa courbe : on prend un segment sur l'axe X, puis on place un point (X,Y) en prenant pour Y le résultat de la fonction F(X). Par exemple si X varie de 0 à 10, nous aurons les points [0, F(0)], [1, F(1)], [2, F(2)] et ainsi de suite. Si vous ne comprenez pas bien, allez vite voir un professeur ou regardez la suite, cela devrait s'éclairer un peu avec les blocks.

Autre notation que vous verrez sûrement en mathématique :

Fonction F : $X \rightarrow F(X)$

Et bien commençons avec quelques **fonctions simples** :

- $X \rightarrow 1$, fonction constante
- $X \rightarrow X$, fonction identité
- $X \rightarrow X+2$, fonction affine
- $X \rightarrow 4*X$, fonction linéaire (j'ai mis une étoile mais pour ne pas confondre avec le X mais cette étoile est le signe multiplié).

Lorsque l'on utilise les quatre opérations, on peut créer une infinité de fonctions qui sont appelées **fonctions polynomiales** de plus en plus complexes surtout si on multiplie X par lui-même :

- $X*X$, qui se dit « X au carré » et qui se note aussi en mathématique X^2 (qui se note $X \wedge 2$ sous BlocksCAD). Cela se dit aussi « X puissance 2 ».
- $X*X*X$ qui se dit « X au cube » et qui se note aussi X^3 (soit $X \wedge 3$ sous BlocksCAD)
- $X*X*X*X$ qui se dit « X puissance 4 » (car X est répété quatre fois) et qui se note aussi X^4 (soit $X \wedge 4$ sous BlocksCAD)
- Et ainsi de suite ...

L'équation paramétrique

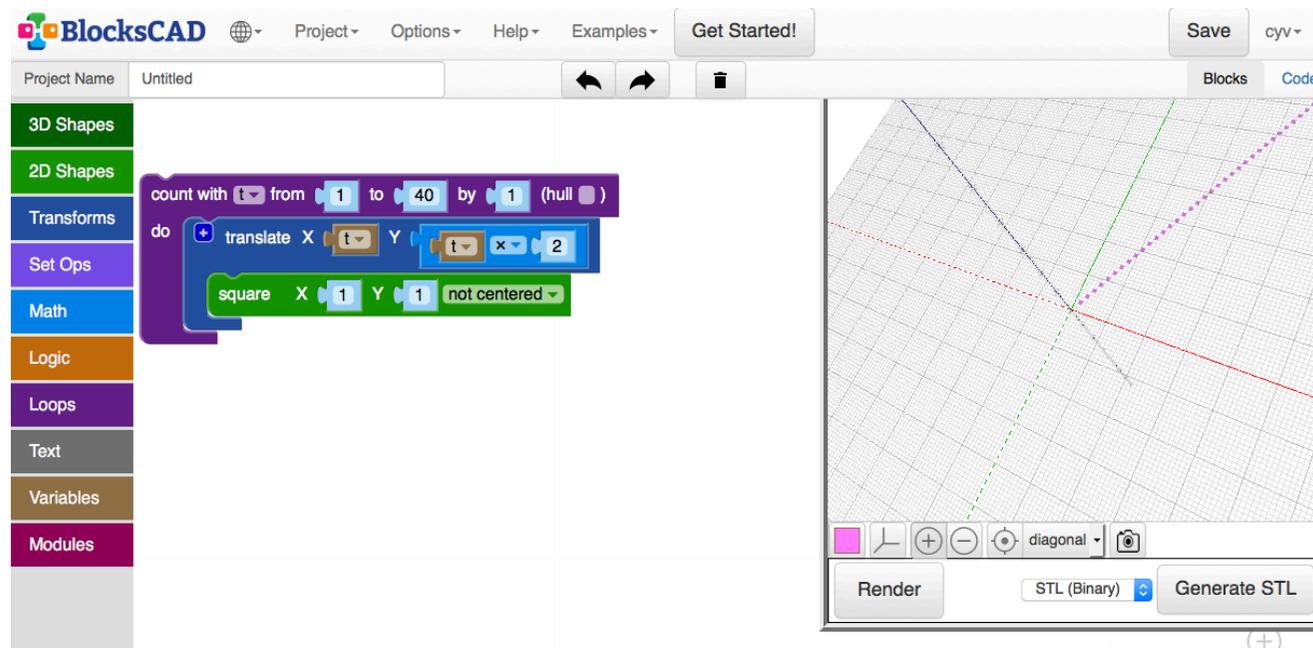
L'avantage des fonctions polynomiales est la facilité pour l'ordinateur de les calculer. Dans le menu gauche MATH, il y a tout ce que l'on veut pour écrire une formule. Prenons une fonction simple : $X \rightarrow 2*X$

Autant cette notation est pratique pour les mathématiciens autant elle n'est pas complète pour nous car il faut indiquer des points à l'ordinateur et aussi on ne peut pas dessiner la courbe pour tous les X : ils y en a une infinité. Nous allons donc utiliser une notation qui est très proche et qui se nomme l'équation paramétrique. Pour cela nous utilisons une autre variable souvent appelé (t) car elle représente le temps qui s'écoule à tracer la courbe.

Ainsi la courbe peut s'écrire :

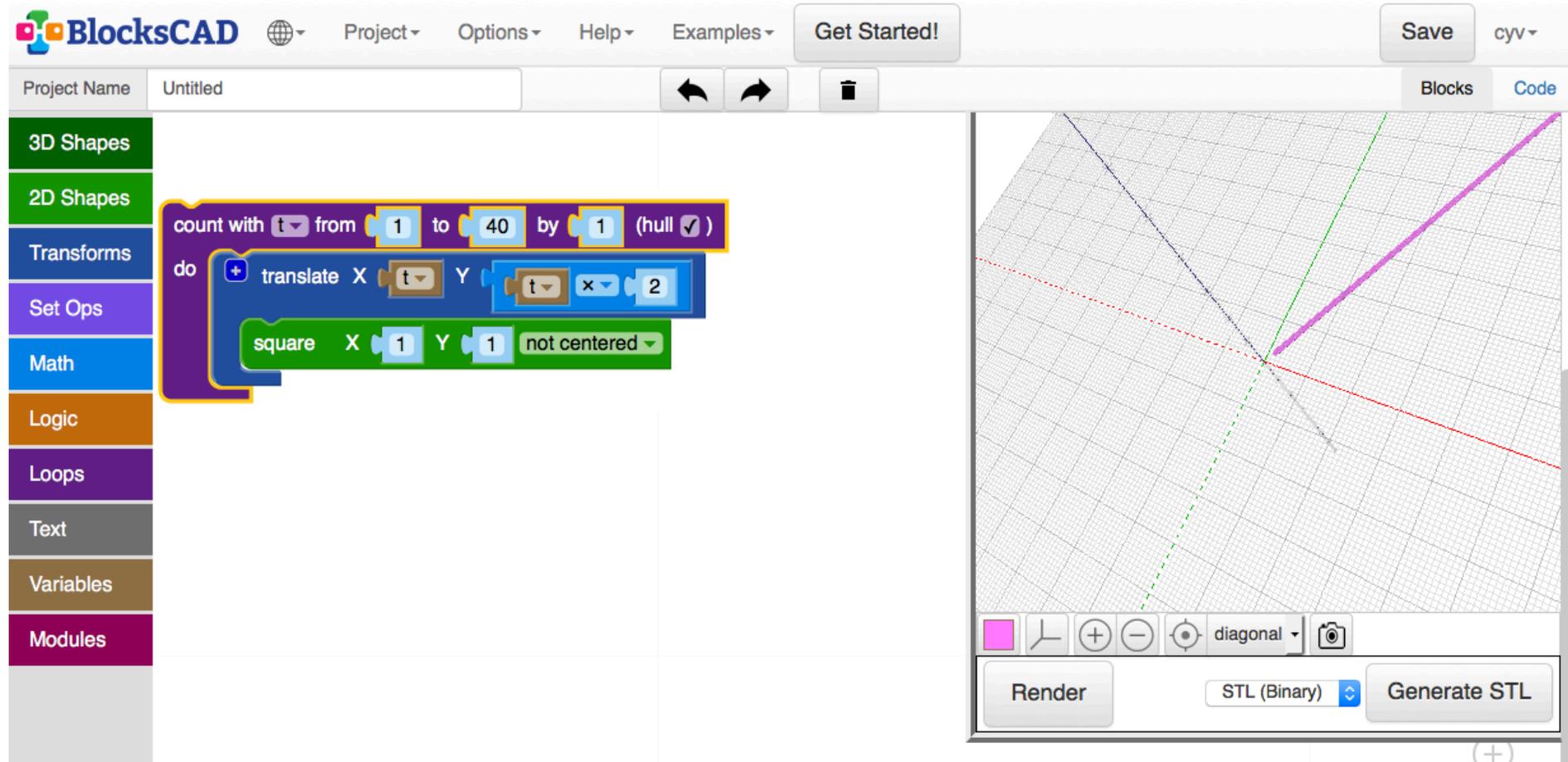
$$\begin{aligned} X &= t \\ Y &= 2*t \end{aligned}$$

Et puis on choisit comment (t) va varier, c'est à dire toutes les valeurs que (t) va prendre successivement. Et pour traduire tout cela de façon informatique, le seul langage que comprend notre ordinateur, nous utilisons la boucle LOOP qui va nous dessiner les points de la fonction $X \rightarrow 2*X$.



Une courbe lissée avec HULL

En rajoutant l'option HULL, nous obtenons une courbe lissée.



Au centre, nous avons mis SQUARE mais nous aurons pouvons utiliser aussi CIRCLE.

L'autre avantage d'utiliser la variable (t) réside dans la possibilité de mettre aussi une formule dans le X de TRANSLATE. Ce n'est pas utile pour les fonctions simples mais nous verrons que cela nous sera utile.

Ecrire des formules avec BlocksCAD

Expliquons maintenant comment écrire les formules mathématiques en code BlocksCAD. Le menu de gauche MATH offre tout ce qu'il faut pour écrire toutes sortes de formules mathématiques.

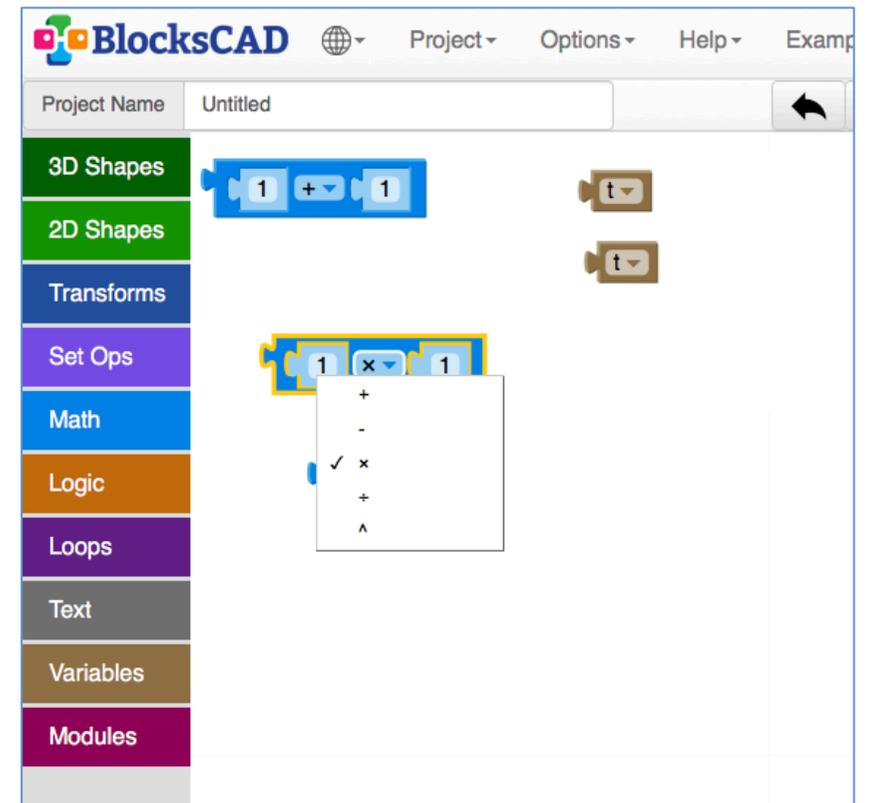
Le bloc de base pour une formule est en haut à gauche. On peut en imbriquer autant que l'on veut. Imaginons que nous voulions faire la formule suivante :

$$t*(t-1) + 4$$

Attention les parenthèses sont importantes : elles indiquent à l'ordinateur les calculs à faire en premier. Regardez la solution ci-dessous.



- En premier (t-1)
- En second $t*(t-1)$
- En dernier $t*(t-1)+4$



Formule de la parabole avec les « puissances »

Et pour terminer dans le bloc de formule, en plus des quatre opérations, le symbole \wedge indique la « puissance ».

Regardez la courbe suivante de la fonction $X \rightarrow X^2 \div 64$ (ou $X^2 \div 64$ ou $X \wedge 2 \div 64$). Cette courbe est appelée aussi parabole. C'est aussi une parabole qui est utilisée pour faire des antennes satellites ou des fours solaires.

The screenshot shows the BlocksCAD software interface. The project name is "Parabole et foyer". The script consists of the following blocks:

- set foyer to 64
- count with t from -40 to 40 by 1 (hull checked)
- do loop:
 - translate X t Y $t \wedge 2 \div \text{foyer}$
 - square X 1 Y 1 centered
- translate X 0 Y foyer $\div 4$
- circle radius 2

The 3D view on the right shows a purple parabolic curve on a grid. A green dashed line represents the axis of symmetry, and a red dashed line represents the focal axis. A pink dot marks the focus. The parabola is symmetric about the vertical axis. The interface includes a top menu bar with "Project", "Options", "Help", and "Examples", a "Get Started!" button, and a "Save" button. The bottom right of the 3D view has a "Render" button and a "Generate STL" button.

Le schéma présente aussi le foyer, donné par la formule $64 \div 4$. La parabole est une courbe étudiée depuis des siècles. Entre autres propriétés, lorsqu'elle reçoit de la lumière (ou des ondes satellites) toute cette lumière se concentre dans le point appelé le foyer.

Une antenne Satellite

Il ne nous reste plus qu'à faire un ROTATE EXTRUDE sur la formule précédente pour avoir une belle antenne satellite.

The screenshot shows the BlocksCAD interface with a script on the left and a 3D view on the right. The script defines a variable 'foyer' as 64, then uses a 'rotate extrude' block with 30 sides. A 'count with' loop from 0 to 40 by 1 (with 'hull' checked) contains a 'do' block. Inside the 'do' block, a 'translate' block moves a 'square' (1x1, centered) to X=t, Y=foyer/t^2. After the loop, another 'translate' block moves a 'circle' (radius 2) to X=0, Y=foyer/4. The 3D view shows a purple satellite dish on a grid, with a red dashed line indicating the focal point. The interface includes a menu bar (Project, Options, Help, Examples, Get Started!), a 'Save' button, and a 'Render' button.

Une fois peinte en argenté ou bien recouvert de papier d'aluminium, vous pourrez vous faire cuire un œuf !

Un four solaire à hot dog

Et pour ceux qui n'aiment pas les œufs, alors utilisons LINEAR EXTRUDE et nous obtenons un parfait four solaire en ligne avec une tige située sur le foyer et permettant d'embrocher la saucisse du Hot Dog.

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', and 'Examples', along with a 'Get Started!' button and a 'Save' button. The project name is 'Parabole logitudinale'. The left sidebar shows a list of categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D model of a solar oven, which is a purple parabolic dish with a black rod (the hot dog stick) passing through its focal point. The model is rendered on a grid. The code blocks are as follows:

- set foyer to 64
- linear extrude height 100 twist 0° scale: x 1 y 1 not centered
- count with t from -40 to 40 by 1 (hull ✓)
- do
 - translate X t Y $t^2 \div 2 + \text{foyer}$
 - square X 1 Y 1 centered
- translate X 0 Y $\text{foyer} \div 4$
- circle radius 2

The bottom right corner of the interface features a 'Render' button, a dropdown menu set to 'FL (Binary)', and a 'Generate STL' button. A camera icon and a 'top' view selector are also visible.

Une autre formule pour réaliser une hyperbole

La fonction $x \rightarrow 1 \div X$ donne une hyperbole. Cette courbe est utilisée dans de nombreuses surfaces et pourra servir plus tard.

The screenshot shows the BlocksCAD interface. The project name is "Hyperbole". The script on the left consists of a "count with" loop from 0.5 to 40 by 0.1, with a "do" block containing a "translate" block (X: t, Y: 25 + t) and a "square" block (X: 1, Y: 1, centered). The 3D view on the right shows a purple hyperbola curve on a grid, with a red dashed line representing the asymptote. The interface includes a menu bar (Project, Options, Help, Examples, Get Started!), a toolbar (Save, cyv), and a bottom panel with "Render" and "Generate STL" buttons.

Dans notre exemple, nous avons pris $X \rightarrow 25 \div X$.

Et puis en utilisant ROTATE EXTRUDE, on obtient au choix, une fleur à l'envers ou un pied de table ou une girole.

The screenshot displays the BlocksCAD interface. The top menu bar includes "Project", "Options", "Help", "Examples", and "Get Started!". The project name is "Hyperbole". The block palette on the left shows categories like "3D Shapes", "2D Shapes", "Transforms", "Set Ops", "Math", "Logic", "Loops", "Text", "Variables", and "Modules". The main workspace shows a 3D model of a pink octagonal base with a thin vertical stem. The block palette shows the following blocks used to create the model:

- rotate extrude sides: 10
- count with t from: 0.5 to: 40 by: 0.1 (hull checked)
- do loop containing:
 - translate X: t Y: 25 ÷ t
 - square X: 1 Y: 1 not centered

The 3D view area includes a grid, axes, and a "Render" button. The bottom right corner has a "Generate STL" button with a dropdown menu set to "STL (Binary)".

Mais à quoi servent les hyperboles

Rajouter un panache de fumée blanche, un panneau jaune avec trois triangles noirs et vous avez une jolie cheminée de centrale nucléaire.

The screenshot displays the BlocksCAD software interface. On the left, a sidebar lists categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D view of a purple chimney-like object on a grid. The code blocks are as follows:

- rotate extrude** (blue block): sides: 10
- rotate** (blue block): X: 0°, Y: 0°, Z: 345°
- translate** (blue block): X: 0, Y: 0
- count with** (purple block): from: 2, to: 10, by: 0.1, (hull: checked)
- do** (purple block):
 - translate** (blue block): X: t, Y: 25 ÷ t
 - square** (green block): X: 1, Y: 1, not centered

The 3D view shows a purple chimney with a flared top and a narrower base. The interface includes a 'Render' button and a 'Smooth' dropdown set to 'Medium'.

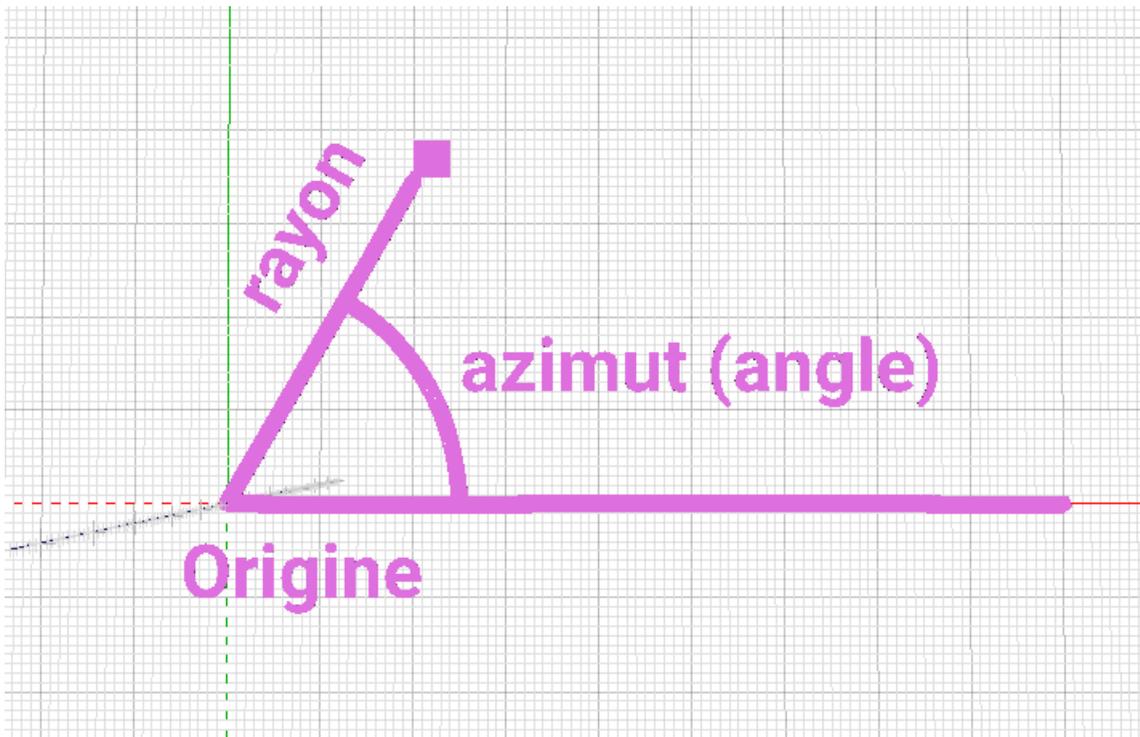
Jolie ?

Les coordonnées Polaires

Les mathématiciens ont inventé d'autres systèmes que la représentation paramétrique ou cartésienne. Sur le plan, il existe aussi la représentation polaire.

La représentation polaire se fonde sur un angle et une longueur.

Imaginez vous au centre du monde, à l'Origine, à l'intersection entre X et Y. Pour regarder les choses qui vous entourent, il vous suffit de vous tourner, donc de connaître un angle, par rapport à un axe (l'axe X dans notre dessin). Une fois dans la direction de l'objet (le petit carré), il vous suffit de connaître la distance entre vous et cet objet pour connaître sa position exacte.



L'angle s'appelle l'AZIMUT et la distance se nomme le RAYON. Cette représentation est très pratique lorsque l'on se place à l'origine, c'est à dire au centre. Les marins, les cartographes, les radars des aéroports l'utilisent car il est assez facile de mesurer un AZIMUT, c'est à dire de mesurer un angle avec un compas de relèvement, un sextant ou une antenne.

Bien sûr, vous aurez remarqué qu'il est nécessaire de définir une droite partant de l'origine car un angle nécessite 2 demi-droites.

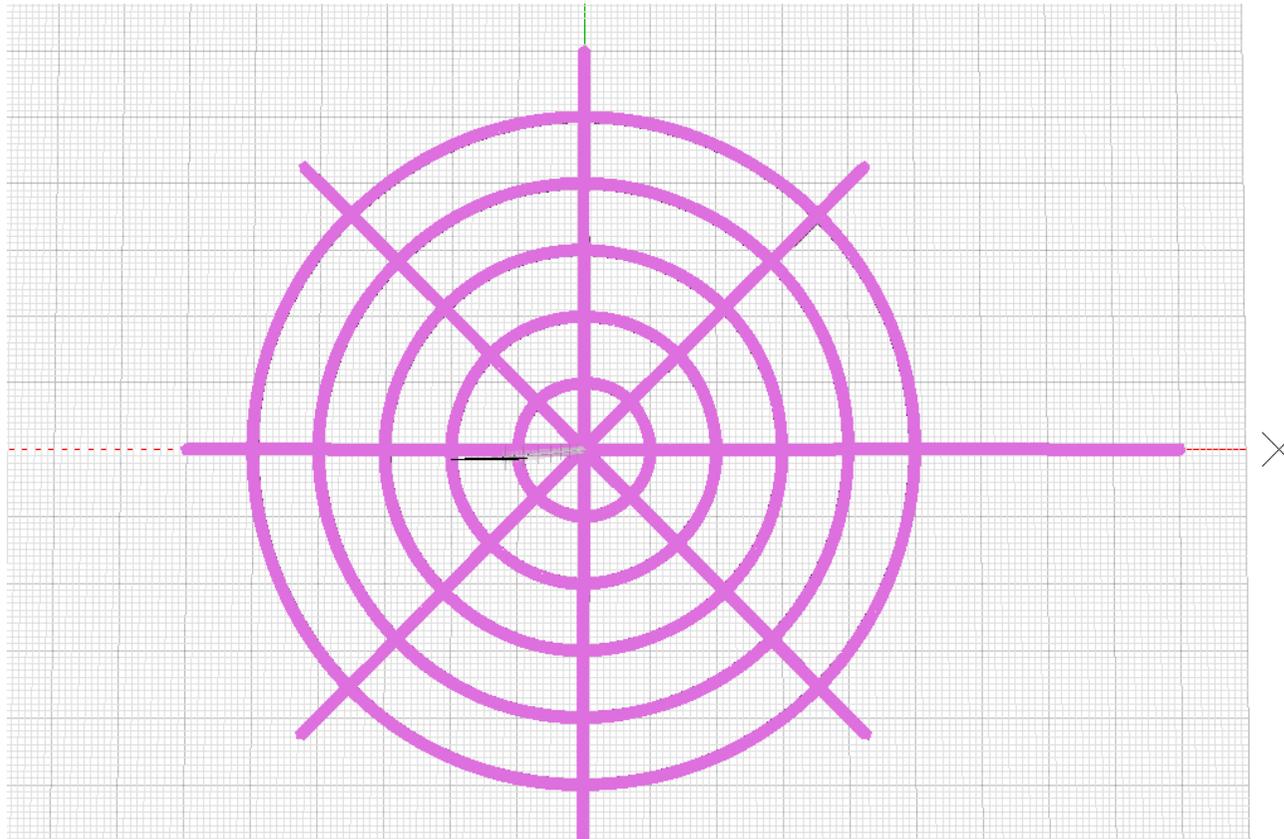
Les quadrillages

Un quadrillage représente les lignes quand les coordonnées sont constantes. Par exemple, pour les coordonnées cartésiennes :

- ❑ Ligne pour $X = 0, X=10, X=20$
- ❑ Ligne pour $Y = 0, Y=10, Y=20$

BlocksCAD donne un double quadrillage, en épais pour les dizaines et en fin pour les unités.

Et si l'on veut dessiner un quadrillage pour les coordonnées polaires, il va ressembler au dessin ci-dessous.



Ce qui est bien différent du quadrillage carré des coordonnées cartésiennes.

Créer un point en coordonnées polaires

Avant de créer des courbes en coordonnées polaire, il faut commencer par un point, dont les coordonnées seront A (pour AZIMUT) et R (pour RAYON).

Faire un point en coordonnées polaires avec BlocksCAD se fait assez facilement. Prenons l'exemple suivant pour lequel nous voulons faire un point polaire avec les coordonnées Rayon = 25 et Azimut = 59°.

- Positionnons un point sur l'origine numéro « 1 »
- Le deuxième point noté « 2 » réalise un TRANSLATE de la longueur du Rayon
- Le troisième point, noté « 3 », réalise un ROTATE du deuxième point d'un angle (suivant Z) de 59°.

On notera au passage que le chiffre 3 est incliné, ceci dû à sa rotation.

The screenshot displays the BlocksCAD software interface. The main workspace shows a 2D coordinate system with a grid. A red dashed line indicates a translation vector from the origin to the point (25, 0). A green dashed line indicates a rotation axis. A green number '3' is shown, rotated 59 degrees from the horizontal. The number '2' is also visible, representing the translated point. The interface includes a menu bar (Project, Options, Help, Examples, Get Started!), a project name field (2D-POLAIRE), and a left sidebar with categories like 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains a script with three blocks: 1. A 2D text block with text '1', size 5, font Roboto. 2. A translate block with X=25, Y=0. 3. A rotate block with X=0°, Y=0°, Z=59°, followed by a translate block with X=25, Y=0, and a 2D text block with text '3', size 5, font Roboto. The 3D view shows a grid with a red dashed line representing the translation vector, a green dashed line representing the rotation axis, and a green number '3' rotated 59 degrees from the horizontal. The number '2' is also visible, representing the translated point. The bottom of the interface has a toolbar with icons for selection, zoom, and rendering, and a 'Render' button.

Le module simple pour créer un point en coordonnées polaires

Dans ce module, nous avons mis en place les variables RAYON et AZIMUT, réalisé le TRANSLATE et le ROTATE.

The screenshot shows the BlocksCAD interface with a custom module named "2D-Point-Polaire" defined. The module's definition is as follows:

```
to 2D-Point-Polaire with: rayon, azimuth
  rotate X 0° Y 0° Z azimuth
  translate X rayon Y 0
  circle radius 1
end
```

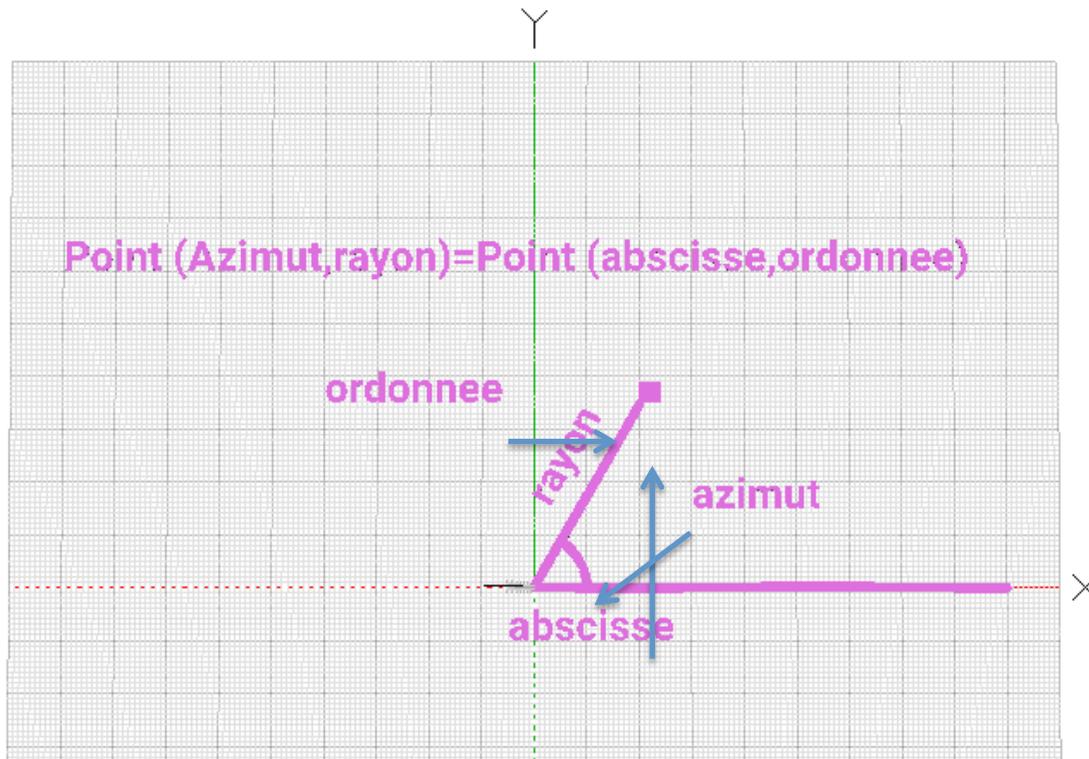
The module is being used in a project named "2D-POLAIRE". The main workspace shows a 2D grid with a red line representing the azimuth angle of 59 degrees and a green circle representing the point at a radius of 25. The interface includes a menu bar (Project, Options, Help, Examples, Get Started!), a toolbar, and a sidebar with categories like 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The bottom right corner has a "Render" button and a "STL (Binary)" dropdown menu.

Avec ce simple module, vous pouvez créer toutes les courbes polaires. Dans le module suivant, nous allons essayer de voir un peu les mathématiques qu'il y a derrière le ROTATE ce qui nous permettra de voir encore d'autres courbes.

Créer un point en coordonnées polaires uniquement avec TRANSLATE

Pour introduire de nouvelles fonctions mathématiques, nous allons faire ce même point polaire mais sans utiliser la fonction ROTATE, c'est à dire uniquement avec la fonction TRANSLATE. Il va falloir donc fabriquer les fonctions qui permettent de passer des coordonnées polaires vers les coordonnées cartésiennes. Ainsi nous allons découvrir ce que fait la fonction ROTATE en son for intérieur.

En fait, on ne va pas aller les chercher très loin ces fonctions pour passer des coordonnées polaires aux coordonnées cartésiennes. Cela fait tellement longtemps que les mathématiciens travaillent dessus que ces fonctions sont présentes dans presque toutes les calculatrices et tous les langages informatiques. On peut même considérer que c'est la suite des 5 opérations (+, -, x, ÷, \wedge) ...

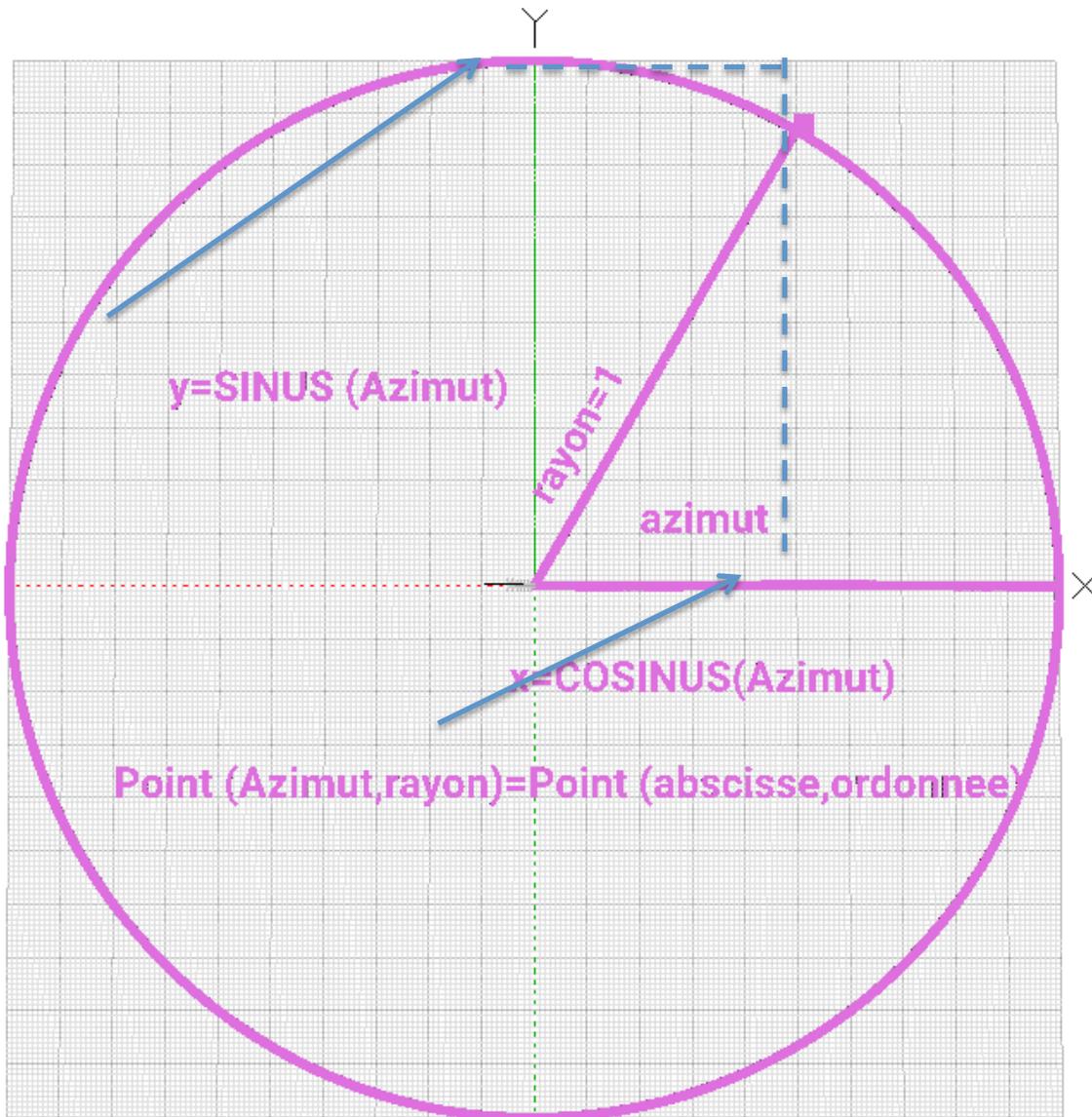


Lorsque le Point qui est représenté en polaire par l'angle A et le rayon 1, et en coordonnées cartésiennes par (x) et (y), alors il existe 2 fonctions appelées cosinus (COS) et sinus (SIN) :

- $x = \text{COS}(A)$
- $y = \text{SIN}(A)$

Et si le rayon n'est pas 1, il suffit de multiplier par le rayon.

- $x = \text{Rayon} * \text{COS}(A)$
- $y = \text{Rayon} * \text{SIN}(A)$

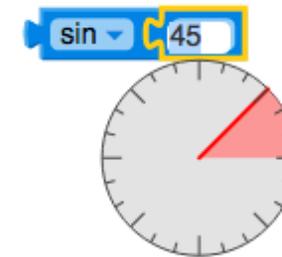


Et pas la peine de réfléchir comment calculer sinus et cosinus, ce sont des fonctions disponibles dans le menu gauche MATH.



La fonction sin, permet de choisir entre SIN (le sinus), COS (le cosinus) et puis d'autres que nous verrons plus tard.

Ensuite, il est assez facile de d'utiliser la fonction SIN en choisissant l'angle, ou aussi en insérant une autre formule.



Le block Point Polaire mathématique

Et pour terminer, le module qui réalise un point CIRCLE en donnant uniquement l'azimut et le rayon.

En haut, l'ordre qui appelle le module. Pour créer le point, il suffit de 2 lignes, un TRANSLATE pour positionner le CIRCLE qui vient ensuite.

The screenshot shows the BlocksCAD interface for a project named "Point Polaire". The left sidebar lists various block categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace displays a block "2D Point Polaire with: rayon 15, azimuth 45°, text 1" connected to a "translate X" block. The "translate X" block contains two math blocks: "rayon * cos(azimut)" for the X-axis and "rayon * sin(azimut)" for the Y-axis. Below these is a "circle radius 1" block. The right side of the interface shows a 3D grid with a pink point and several colored lines (red, green, blue, purple) representing the axes and the point's position. At the bottom right, there are buttons for "Render", "STL (Binary)", and "Generate STL".

Rappelez vous que le cosinus et le sinus ont été définis avec un rayon égal à 1. Lorsque le rayon est différent de 1, il suffit de multiplier par ce rayon. La formule est la suivante :

- $x : \text{rayon} * \cos(\text{azimut})$
- $y : \text{rayon} * \sin(\text{azimut})$

Et voilà nous avons notre autre module pour un point polaire, équivalent au premier et uniquement avec une formule mathématique et pas de ROTATE.

La fonction IF (« si » en anglais)

En fin, le block suivant est identique au précédent avec quelques fonctions pour écrire en texte dans BlocksCAD le rayon et l'azimut en utilisant une nouvelle fonction informatique : IF (qui veut dire « si » en anglais).

The screenshot displays the BlocksCAD interface. On the left, a script is built using the following blocks:

- 2D Point Polaire** with: rayon 15, azimut 45°, text 1.
- to [2D Point Polaire] with: rayon, azimut, text**
- union**
- translate X** (rayon * cos azimut) **Y** (rayon * sin azimut)
- circle** radius 1
- if** (text == 1)
- do**
 - union**
 - 2D text** (rayon) size 3 font Roboto
 - plus** **translate X** 5 **Y** 0
 - 2D text** (azimut) size 3 font Roboto
 - plus** **translate X** 8 **Y** 0
 - 2D text** ("° ") size 3 font Roboto

On the right, the 3D view shows a grid with a purple point, a green circle, and a red line. The text "15 45°" is rendered in purple, with "15" at the end of the red line and "45°" below it. The interface includes a top menu bar, a project name field ("Point Polaire"), and buttons for "Render" and "Generate STL".

La fonction IF permet de donner un ordre conditionnel à l'ordinateur : **si** la variable (text) égale 1 alors on réalise les fonctions suivantes, sinon on ne fait rien.

Une courbe en coordonnées polaires

Comme pour une courbe en coordonnées cartésiennes, la courbe en coordonnées polaires va utiliser la boucle informatique LOOP.

The screenshot shows the BlocksCAD interface for a project named "2D Courbe Polaire". The left sidebar lists various categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace displays a 2D grid with a pink curve and several dashed lines. The code blocks are as follows:

- 2D Courbe with: tour (0.7)
- to 2D Courbe with: tour
- count with i from 1 to $\text{tour} \times 360$ by 1 (hull checked)
- do loop:
 - 2D Point Polaire with:
 - rayon: $10 + 10 / i$
 - azimut: i
 - text: 0
- to 2D Point Polaire with: r...

The bottom right of the interface includes a "Render" button, a "STL (Binary)" dropdown, and a "Generate STL" button.

Le bloc comprend :

- Une boucle LOOP avec une variable (TOUR) qui va indiquer combien de fois il faut faire de tour d'angle.
- La formule du rayon en fonction de l'angle. La formule donnée dans le dessin a été faite un peu au hasard

Le cercle polaire

Un cercle, que l'on peut aussi appeler une courbe circulaire, est simple : le rayon ne bouge pas et l'angle varie de 0 à 360°.

The screenshot shows the BlocksCAD interface. The script on the left is as follows:

```
count with i from 0 to 360 by 1 (hull ✓)
do
  2D Point Polaire with:
    rayon 10
    azimuth i
  to 2D Point Polaire with: rayon, azimuth
    translate
      X rayon x cos azimuth
      Y rayon x sin azimuth
    circle radius 1
```

The rendered view on the right shows a 2D grid with a pink circle centered at the origin. The circle has a radius of 10 units. The axes are labeled X and Y. The script uses the '2D Point Polaire' block to generate the circle's points, and the 'circle' block to draw it.

Bien sur, cela donnerait exactement le même résultat en utilisant le point polaire avec ROTATE.

L'arc de cercle

Je vous entends déjà : « mais pourquoi faire un cercle en coordonnées polaires alors que nous avons la fonction CIRCLE ». Effectivement au premier abord, cela semble assez peu utile. Seulement, la fonction CIRCLE ne permet pas de tracer des arcs de cercle. Pour le faire, il faut passer par une fonction CIRCLE et une opération DIFFERENCE, ce qui n'est pas très pratique.

Avec notre cercle polaire, nous pouvons facilement faire l'arc que nous voulons en choisissant le début et la fin en changeant simplement la variable de LOOP puisqu'elle représente l'angle. Reprenons, nos blocs qui dessinaient un cercle et réalisons deux arcs.

The screenshot shows the BlocksCAD interface with a script for drawing two arcs. The script consists of three main blocks:

- A loop block: `count with i from 15 to 77 by 1 (hull ✓)`
- A `do` block containing a `2D Point Polaire` block with `rayon` set to 40 and `azimut` set to `i`.
- A second loop block: `count with i from 99 to 154 by 1 (hull ✓)`
- A `do` block containing a `2D Point Polaire` block with `rayon` set to 40 and `azimut` set to `i`.
- A `to 2D Point Polaire with: rayon, azimut` block containing:
 - `translate X rayon * cos azimut Y rayon * sin azimut`
 - `rotate X 0° Y 0° Z -1 * rayon`
 - `circle radius 1`

The right side of the interface shows a 2D grid with two arcs drawn in purple. The first arc is between 15° and 77°, and the second is between 99° and 154°. The interface also includes a `Render` button, a `STL (Binary)` dropdown, and a `Generate STL` button.

Le premier bloc fait un arc entre 15° et 77°, c'est à dire un arc de 62° et le second arc de 55° commençant à 99°. Moins facile à faire avec CIRCLE, DIFFERENCE et INTERSECTION !

La part de tarte

Bien sûr, dès que l'on voit le dessin précédent, on se demande pourquoi ne pas les transformer en deux parts de tartes. Cela irait si bien avec la glace à l'italienne réalisée en début de livre. Miam !

The screenshot shows the BlocksCAD interface. The project name is '2D droite polaire'. The script consists of several blocks: four '2D droite polaire' blocks with specific azimuth, start radius, and end radius values; two 'count with' blocks; and two 'do' blocks containing '2D Point Polaire' blocks. The 3D view shows a pie chart with three segments in pink, purple, and blue, on a grid. The interface includes a 'Render' button and a 'Generate STL' button.

Regardons un peu ces blocs. Un module « 2D droite polaire » permet de faire un segment de droite, c'est à dire un bout de droite avec trois paramètres :

- L'azimut, c'est à dire l'angle que va faire cette demi-droite par rapport à l'axe des X
- Le début du rayon, pour commencer ce segment
- La fin du rayon, pour finir le segment

Ensuite, il suffit d'appeler autant de fois le module que nous voulons dessiner de segments. Tiens quelqu'un a mordu dans la part de gauche !!

Une rosace (2D Rosace)

Avec les arcs et les segments, il y a un grand nombre de possibilités pour dessiner des profils. Maintenant passons à d'autres formes qui sont le résultat d'équations (c'est à dire de formules) déjà connues depuis longtemps par les mathématiciens.

La rosace est une courbe qui se décrit facilement par une formule mathématique en coordonnées polaires. La formule magique est :

$$\text{RAYON} = (\text{taille}) \times \text{SIN} (\text{pétale} \times \text{AZIMUT})$$

Nous avons déjà vu le RAYON et l'AZIMUT des coordonnées polaires ainsi que le sinus (SIN).

The screenshot shows the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', and 'Examples'. The main workspace displays a 2D grid with a pink rose-like shape (rosace) drawn. The left sidebar shows a '2D Shapes' category with a '2D Rosace' block. The central workspace shows a script with a 'to 2D Rosace with: taille, pétale, tour' block, a 'count with i from 0 to tour x 360 by 1 (hull checked)' block, and a 'do' loop containing a '2D Point Polaire with: rayon (taille x sin (pétale x i)), azimut (i), text (0)' block. The right sidebar shows 'Render' and 'Generate STL' buttons.

Les paramètres de la Rosace

Les 3 paramètres permettent :

- TAILLE détermine la longueur d'un pétale du centre à son extrémité
- PETALE
 - Si pair (c'est à dire divisible par 2), alors cela produit une fleur de (PETALE divisée par 2) pétales
 - Si impair, alors la fleur comprend autant de pétales que le nombre PETALE
 - Si PETALE est un nombre rationnel (la division de 2 nombres entiers) alors les pétales se chevauchent. Exemple ci-dessous (13/7)
- TOUR, permet de facilement augmenter le nombre de boucle que doit faire le programme lorsque PETALE est rationnel. Si vous le laissez à 1, la courbe ne sera pas finie.

La Spirale d'Archimède

La Spirale d'Archimède, découverte par Archimède paraît-il, a une formule simple du rayon en fonction de l'azimut.

$RAYON = (PAS) \times AZIMUT$. Le paramètre PAS est la distance entre les bras de la spirale.

The screenshot shows the BlocksCAD software interface. The project name is "2D Spirale Archimede". The code blocks are as follows:

- 2D Spirale Archimede** with:
 - pas: 1
 - tour: 4
- to 2D Spirale Archimede with: pas, tour**
 - count with i from 0 to $\text{tour} \times 360$ by 1 (hull checked)
 - do
 - 2D Point Polaire** with:
 - rayon: $\text{pas} + 20 \times i$
 - azimut: i
 - text: 0
- to 2D Point Polaire with: r...**

The right side of the interface shows a 2D grid with a purple Archimedean spiral. The bottom right of the interface features a "Render" button, a "STL (Binary)" dropdown, and a "Generate STL" button.

Le paramètre TOUR, permet de régler le nombre de tour de spirale désiré.

Les courbes périodiques

Une onde sinusoïdale

L'onde c'est cette déformation qui se propage sur un liquide ou un solide : un petit caillou dans une flaque, un drap secoué et une onde apparaît.

On parle souvent d'onde sinusoïdale car assez souvent ces ondes ont la forme d'une courbe donnée par la fonction sinus. Mais quelle est cette forme ? Nous avons utilisé les sinus et cosinus pour passer des coordonnées polaires aux coordonnées cartésiennes mais que se passe-t-il si on utilise la fonction SIN pour tracer une courbe en coordonnées cartésiennes.

The screenshot shows the BlocksCAD interface. On the left, a code block is visible with the following structure:

```
count with i from -60 to 60 by 0.3 (hull ✓)  
do  
  translate X i Y 30 x sin 20 x i  
  circle radius 1
```

On the right, a 2D plot shows a pink sine wave on a grid. The wave oscillates around a horizontal red dashed line. A vertical green dashed line is at the center. Blue double-headed arrows indicate the period and amplitude of the wave. Below the plot, there are controls for rendering and generating STL files.

Et voilà à quoi ressemble la fonction $X \rightarrow \sin(X)$, des petites vagues. Dans le programme, nous avons dû mettre des coefficients pour que cela soit bien visible sur le dessin, mais c'est bien la fonction sinus.

Amplitude et période

Maintenant, faisons le raisonnement à l'envers. Nous voulons une période de 10 (la période d'une fonction sinus est la distance entre 2 crêtes) et une amplitude de 2 (l'amplitude est la moitié de la hauteur de la crête). La formule est :

$$\text{Amplitude} * \sin (360 \div \text{période}).$$

Et voilà un petit module pour facilement réaliser cela.

The screenshot shows the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', and 'Examples'. The project name is 'Sinusoidale Plus'. The sidebar on the left lists various categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace displays a 3D grid with a purple sine wave. The code blocks are: 'sinus with: période 10, amplitude 1', 'to sinus with: période, amplitude', 'count with from -60 to 60 by 0.3 (hull checked)', 'do' loop containing 'translate X 1 Y amplitude * sin (360 / période) * 1 Z 0' and 'sphere radius 0.5'. The bottom right has 'Render' and 'Generate STL' buttons.

Un peu d'osier

Pour aller plus loin sur la fonction sinus, maintenant que nous connaissons la période et l'amplitude, introduisons la phase. La phase est le décalage par rapport à l'origine. Nous pourrions utiliser la fonction TRANSLATE, mais il est souvent plus facile de décaler en utilisant la formule du sinus la phase comme variable.

Regardez ce que cela donne pour fabriquer une plaque en osier. Il manque quelques tiges mais nous vous laissons le soin de les rajouter.

Cet exemple va aussi nous permettre d'introduire la fonction REMAINDER OF. Dans la boucle, nous voulons avoir une phase à 0° puis une phase à 180° puis une phase à 0° et ainsi de suite.

REMAINDER OF est le reste d'une division. Si la division est par 2 ($\div 2$), le reste est 0 pour les nombres pairs et 1 pour les nombres impairs. Il suffit ensuite de multiplier par 180 pour obtenir : 0, 180, 0, 180, ...

Restons en phase

Cette expression du langage courant vient de la phase des sinusoides. Voici 4 courbes :

- en rouge, la courbe de base,
- en jaune une phase de 90° , c'est à dire une courbe décalée d'un quart de la période. Les physiciens disent que les courbes sont en quadratures.
- en vert, une courbe décalée d'une demi-période, totalement symétrique par rapport à la courbe rouge. On dit que les courbes sont en opposition de phase. Elles ne s'aiment vraiment pas !!
- en bleu, une phase de 270° ce qui revient aussi à -90° .

The screenshot shows the BlocksCAD interface with a project named "Sinusoidale Plus". The main window displays a 3D plot of four sine waves on a grid. The waves are colored red, yellow, green, and blue, corresponding to the descriptions in the text. The red wave is the base sine wave. The yellow wave is shifted by 90° . The green wave is shifted by 180° . The blue wave is shifted by 270° . The plot includes a coordinate system with X, Y, and Z axes. Below the plot are controls for "Render" and "Generate STL" (with a dropdown for "STL (Binary)").

On the left, there are four "sinus with:" blocks, each with a color selector and parameters for "periode" (20), "amplitude" (5), and "phase" (0° , 180° , 90° , and 270° respectively).

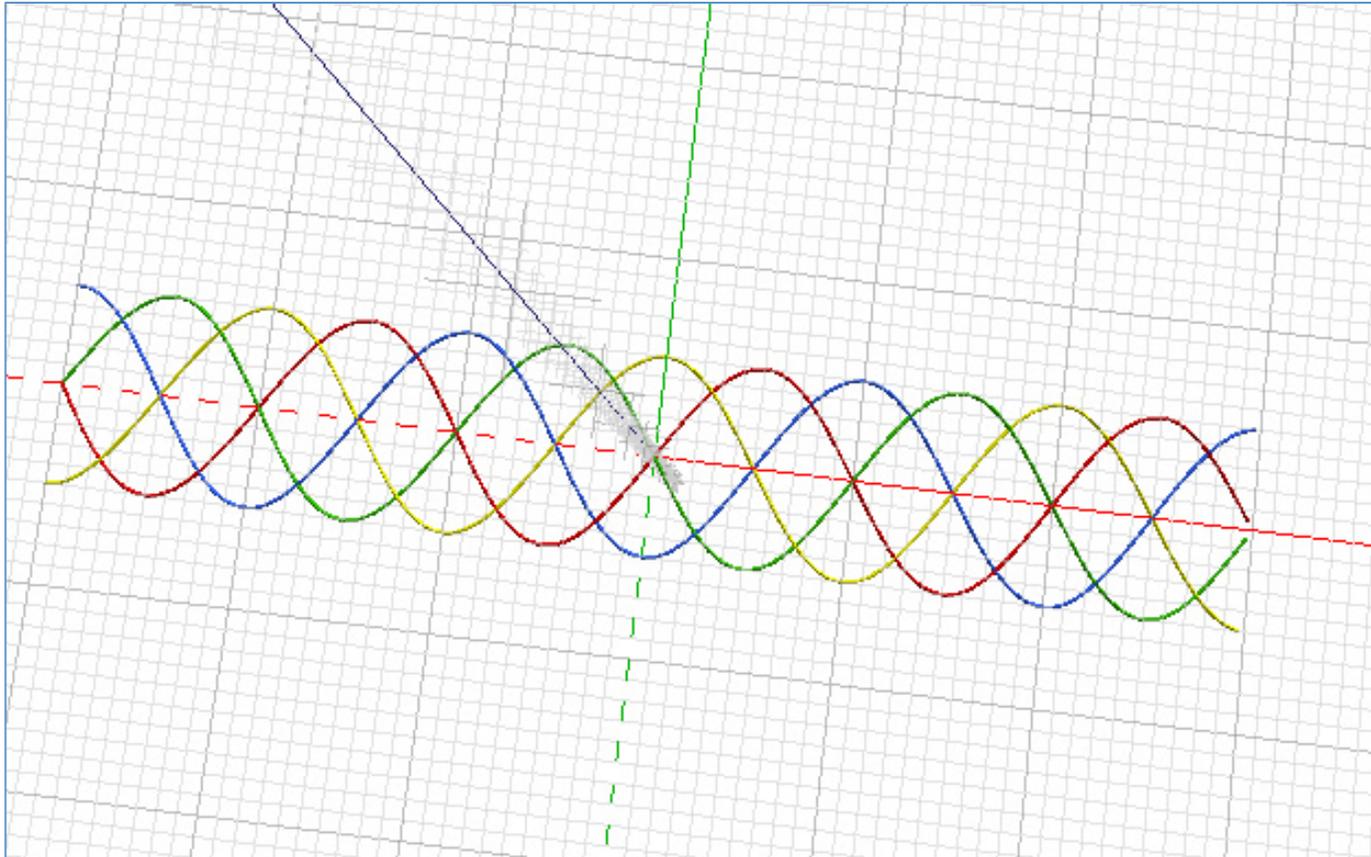
At the bottom, a code block is shown, which is a loop that generates a series of small spheres. The code is as follows:

```

to sinus with: periode, amplitude, phase
  count with i from -30 to 30 by 0.3 (hull ✓)
  do
    translate X i Y [amplitude] x sin [360 + periode] x i + phase Z 0
    sphere radius 0.1
  
```

Mieux gérer les extrémités avec la phase

Et si on regarde le résultat de plus loin, on comprend l'intérêt d'utiliser la phase et non pas une transformation TRANSLATE. Les courbes démarrent et finissent au même endroit.



REMAINDER

Utilisons encore cette fonction REMAINDER OF. Maintenant, nous voulons 4 courbes, c'est à dire 4 déphasages. Au lieu de diviser par 2, nous divisons par 4. Une petite astuce. IL faut commencer de compter à partir de 4 sinon le reste de la division pour (1, 2, 3 et 4) sera toujours égal à zéro. Pour cela il suffit d'additionner 4 à la variable j.

The screenshot shows the BlocksCAD software interface. The project name is "REMAINDER OF 4". The script in the workspace consists of the following blocks:

- count with () from 1 to 10 by 1 (hull)
- do loop containing:
 - sinus with:
 - periode: 20
 - amplitude: 5
 - phase: 90°
 - remainder of (j + 4 / 4)
- to sinus with: periode, amp...

The 3D view on the right shows a grid with a rendered purple sine wave and a red dashed line. The bottom right of the interface has a "Render" button and a "Generate STL" button.

HSV ou TSV

Les courbes précédentes sont toutes en violet. Il serait sympa de les différencier avec des couleurs. Pour cela nous allons jouer avec COLOR, une transformation disponible dans le menu de gauche TRANSFORMS. Cette transformation permet de changer la couleur avec le paramétrage HSV qui se dit TSV en français. https://fr.wikipedia.org/wiki/Teinte_Saturation_Valeur

Le T c'est pour « Teinte » c'est à dire la couleur.

The screenshot shows the BlocksCAD interface. On the left is a sidebar with categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains a script with the following blocks:

- A 'count with' block: 'count with j from 1 to 10 by 1 (hull)'
- A 'do' loop containing:
 - A 'color HSV' block with 'hue' set to 'j x 36', 'saturation' to 100, and 'value' to 100.
 - A 'sinus with' block with 'periode' 20, 'amplitude' 5, and 'phase' '90° x remainder of j + 4 / 4'.
- A 'to sinus with: periode, amp...' block.

On the right, a 3D plot shows a grid with several colored sine waves (magenta, cyan, blue, green, yellow) plotted. The plot has a 'Render' button and a 'Generate STL' button.

Pour simplifier, « HUE » varie entre 0° rouge (et aussi 360°) en passant par 60° : jaune ; 120° : vert ; 180° : cyan ; 240° : bleu ; 300° : magenta.

Concrètement comment faire pour bien différencier les couleurs des sinusoides. Comme « i » varie de 1 à 10, et que le maximum est de 360, alors il faut que HUE égale 360 lorsque $J=10$. La formule est donc $HUE = J \times 36$.

Un cercle ondulé

Comme nous l'avons déjà dit, il n'est pas utile de savoir comment l'ordinateur calcule la fonction sinus, il suffit de l'utiliser comme si c'était une opération de plus.

Le bloc suivant permet de faire un cercle ondulé qui pourra ensuite facilement se transformer en tube ondulé. Cela se fait en additionnant la fonction sinus (les ondulations) à un cercle classique. Ensuite, il faut régler les différents paramètres pour obtenir ce que l'on souhaite.

The screenshot displays the BlocksCAD interface for a project named "2D Rosace". The left sidebar shows various tool categories, with "2D Shapes" selected. The main workspace features a 2D grid with a pink wavy circle. The top bar includes navigation and utility buttons, and the bottom bar contains rendering and export options.

Cela ressemble un peu à un engrenage mais seulement un peu. Il devient plus facile de comprendre l'utilité des formules : nous pouvons les additionner et les multiplier et les combiner pour faire des courbes très complexes.

Une fonction EXTRUDE sur le profil précédent

The screenshot displays the BlocksCAD software interface. The top menu bar includes "Project", "Options", "Help", "Examples", and "Get Started!". The project name is "2D cercle ondulé". The left sidebar shows various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules.

The main workspace shows a 3D model of a purple wavy ring. The ring is composed of multiple overlapping, wavy segments. The 3D view includes a coordinate system with X, Y, and Z axes. Below the 3D view is a toolbar with icons for color selection, zooming, and camera control, along with a "diagonal" dropdown menu and a camera icon.

The code editor on the left shows the following blocks:

- linear extrude** block with parameters: height: 40, twist: 45°, scale: x: 1, y: 1, not centered.
- 2D cercle ondulé** block with parameters: taille: 30, ondulation: 5, frequence: 20.
- to 2D cercle ondulé** block with parameters: taille, ondulation, frequence.
- count with** block with parameters: from: 0, to: 360, by: 1, (hull checked).
- do** loop containing:
 - 2D Point Polaire** block with parameters: rayon: taille + ondulation * sin, azimut: i * frequence, text: 0.
- to 2D Point Polaire** block with parameter: r...

At the bottom of the interface, there are buttons for "Render" and "Generate", and a dropdown menu set to "STL (Binary)".

Une formule d'ellipse

Une bonne formule pour dessiner une ellipse en utilisant l'équation paramétrique suivante :

$$X \rightarrow \text{largeur} * \text{COS} (i)$$

$$Y \rightarrow \text{hauteur} * \text{SIN} (i)$$

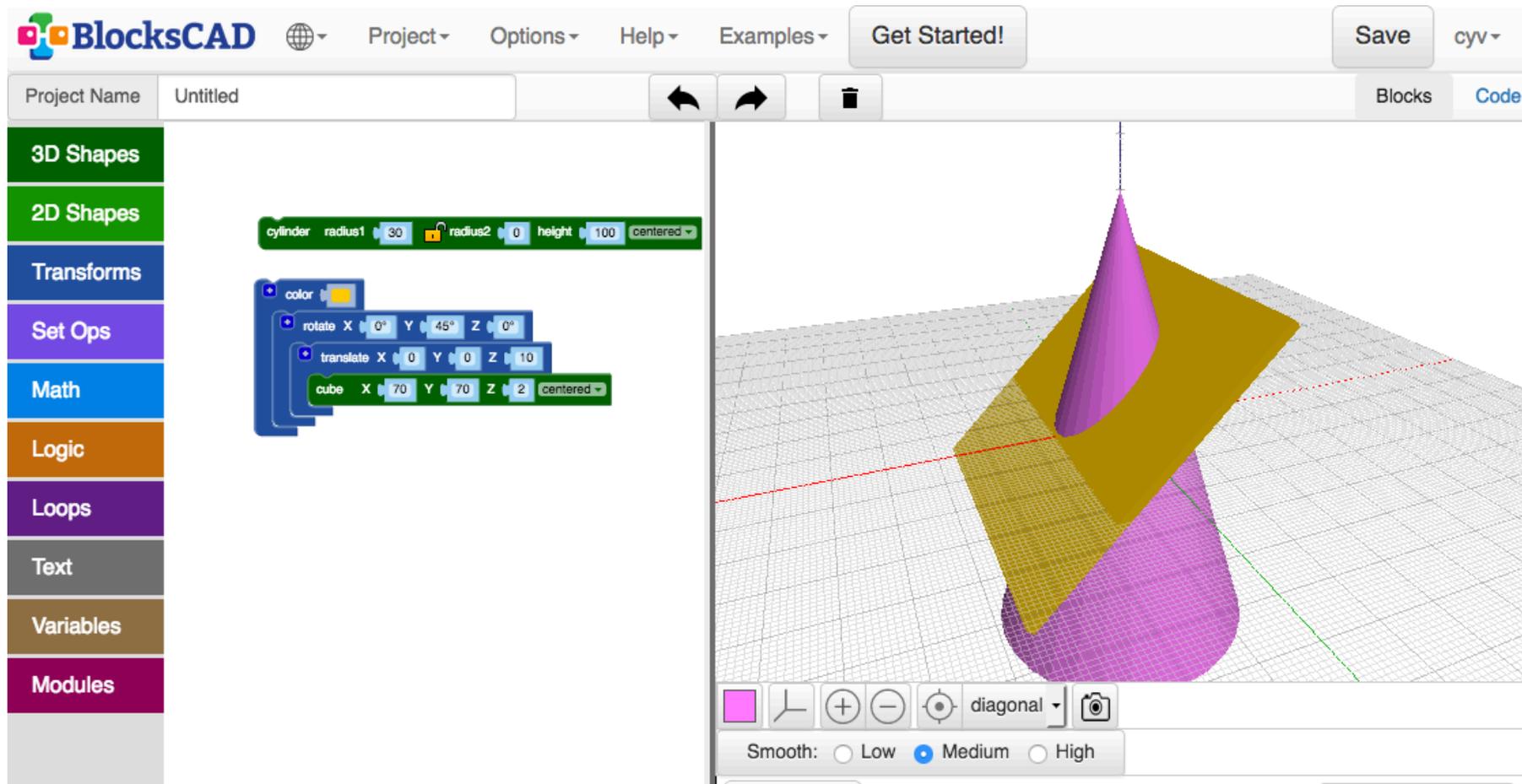
On obtient une belle ellipse, vite calculée par BlocksCAD.

The screenshot shows the BlocksCAD software interface. The project name is "2D arc d'ellipse". The sidebar on the left lists various categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains two "Arc ellipse with:" blocks and a "do" loop block. The "do" loop block is configured with a "count with i" from "angle début" to "angle fin" by "1" (hull checked). Inside the loop, there is a "translate X" block with "X" set to "largeur * cos(i)" and "Y" set to "hauteur * sin(i)", followed by a "circle radius 1" block. The right panel shows a 3D grid with a pink arc of an ellipse drawn on it. The bottom right of the interface has a "Render" button, a "STL (Binary)" dropdown, and a "Generate STL" button.

Voici donc, un module tout simple que l'on peut facilement paramétrer pour avoir soit une ellipse complète (début à zéro et fin à 360) soit un arc d'ellipse. L'ellipse a l'avantage de dessiner des courbes plus douces que le cercle.

Les coniques : Ellipse, hyperbole et parabole

Puisque nous avons abordé l'ellipse, généralisons en rajoutant l'hyperbole et la parabole. Ces trois courbes sont nommées des coniques car elles peuvent être définies par l'intersection d'un plan et d'un cône. Suivant l'inclinaison du plan par rapport à ce cône, on obtient les différents types de courbes. Donc traçons rapidement un cône et un plan pour voir ce que cela donne.



Pour les schémas suivants nous appliquerons l'opération INTERSECTION pour bien montrer la courbe.

Le cercle est une conique

Si le plan est horizontal (ROTATE 0°), cela donne un cercle.

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', and 'Examples' menus, along with a 'Get Started!' button. The 'Project Name' field is set to 'conique cercle'. The left sidebar shows a list of categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D grid with a yellow circle. The 'Intersection' block is expanded, showing a 'cylinder' block with radius1: 30, radius2: 0, height: 100, and centered: true. It is followed by a 'with' block containing a 'color' block (yellow), a 'rotate' block (X: 0°, Y: 0°, Z: 0°), a 'translate' block (X: 0, Y: 0, Z: 0), and a 'cube' block (X: 70, Y: 70, Z: 2, centered: true). The bottom toolbar includes a 'Render' button, a 'Smooth' dropdown set to 'Medium', and a 'Generate STL' button with a format dropdown set to 'STL (Binary)'.

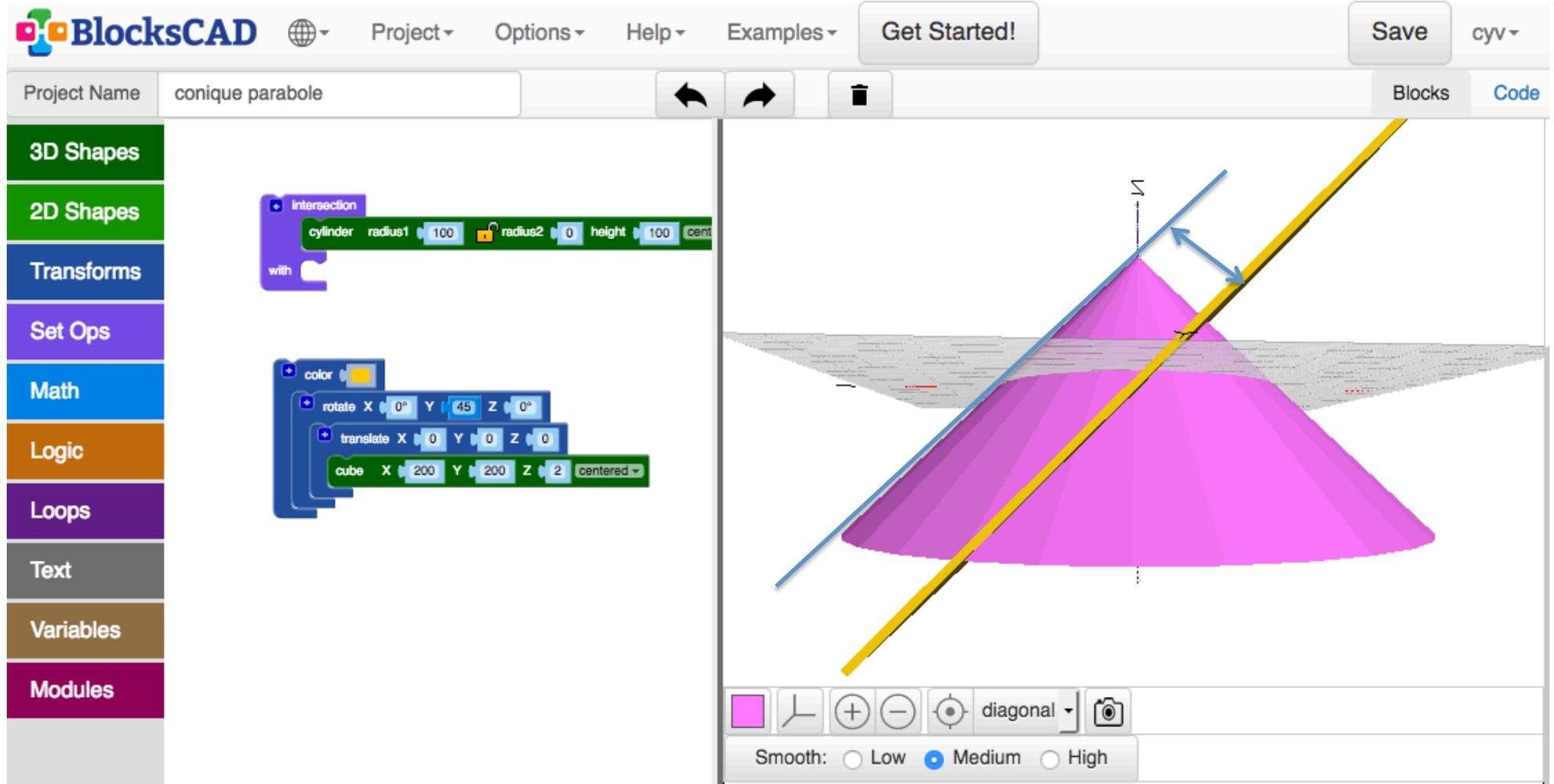
L'ellipse est une conique

SI le plan est incliné un peu (c'est à dire entre l'horizontal et la pente du cône) cela devient une ellipse.

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', 'Examples', and 'Get Started!'. The 'Project Name' field is set to 'conique ellipse'. The left sidebar shows a list of categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The central workspace shows a 3D model of a yellow elliptical cone on a grid. The model is created using a block-based script: 'Intersection' containing 'cylinder' (radius1: 30, radius2: 0, height: 100, centered), 'with' containing 'color' (yellow), 'rotate' (X: 0°, Y: 45°, Z: 0°), 'translate' (X: 0, Y: 0, Z: 0), and 'cube' (X: 70, Y: 70, Z: 2, centered). Below the 3D view are controls for color, zoom, rotation, and smoothness (Low, Medium, High). At the bottom are 'Render' and 'Generate STL' buttons.

La parabole est une conique

Si l'inclinaison est exactement la pente du cône, cela devient une parabole.



On voit bien sur le dessin précédent que le plan est dans la même inclinaison que le côté du cône.

Voici donc ci-dessous la parabole réalisé par BlocksCAD.

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', and 'Examples', along with a 'Get Started!' button and a 'Save' button. The project name is 'conique parabole'. The left sidebar contains a vertical menu with categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D model of a yellow paraboloid. The block-based code on the left defines the object as follows:

- Intersection** block containing:
 - cylinder** block with parameters: radius1: 100, radius2: 0, height: 100, centered.
 - with** block containing:
 - color** block with a yellow color swatch.
 - rotate** block with parameters: X: 0°, Y: 45, Z: 0°.
 - translate** block with parameters: X: 0, Y: 0, Z: 0.
 - cube** block with parameters: X: 200, Y: 200, Z: 2, centered.

The bottom toolbar includes a color selection tool (pink square), navigation tools (pan, zoom in, zoom out, rotate), a 'diagonal' dropdown menu, a camera icon, and a 'Smooth' control with radio buttons for Low, Medium (selected), and High.

l'hyperbole est une conique

Et pour terminer, lorsque le plan est plus inclinée que la pente du cône.

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', and 'Examples', along with a 'Get Started!' button and a 'Save' button. The project name is 'conique hyperbole'. The left sidebar contains categories: '3D Shapes', '2D Shapes', 'Transforms', 'Set Ops', 'Math', 'Logic', 'Loops', 'Text', 'Variables', and 'Modules'. The central workspace shows a 3D model of a pink cone intersected by a grey horizontal plane. A yellow line represents the cone's axis, and a purple hyperbola is visible on the cone's surface. The block editor on the left shows the following blocks: 'Intersection' (cylinder radius1: 100, radius2: 0, height: 100), 'with', 'color' (yellow), 'rotate X: 0°, Y: 75, Z: 0°', 'translate X: 0, Y: 0, Z: 0', and 'cube X: 200, Y: 200, Z: 2, centered'. The bottom toolbar includes a color picker, a 'diagonal' view selector, and a 'Smooth' control set to 'Medium'.

Nous obtenons notre fameuse hyperbole.

BlocksCAD

Project Name: conique hyperbole

3D Shapes

2D Shapes

Transforms

Set Ops

Math

Logic

Loops

Text

Variables

Modules

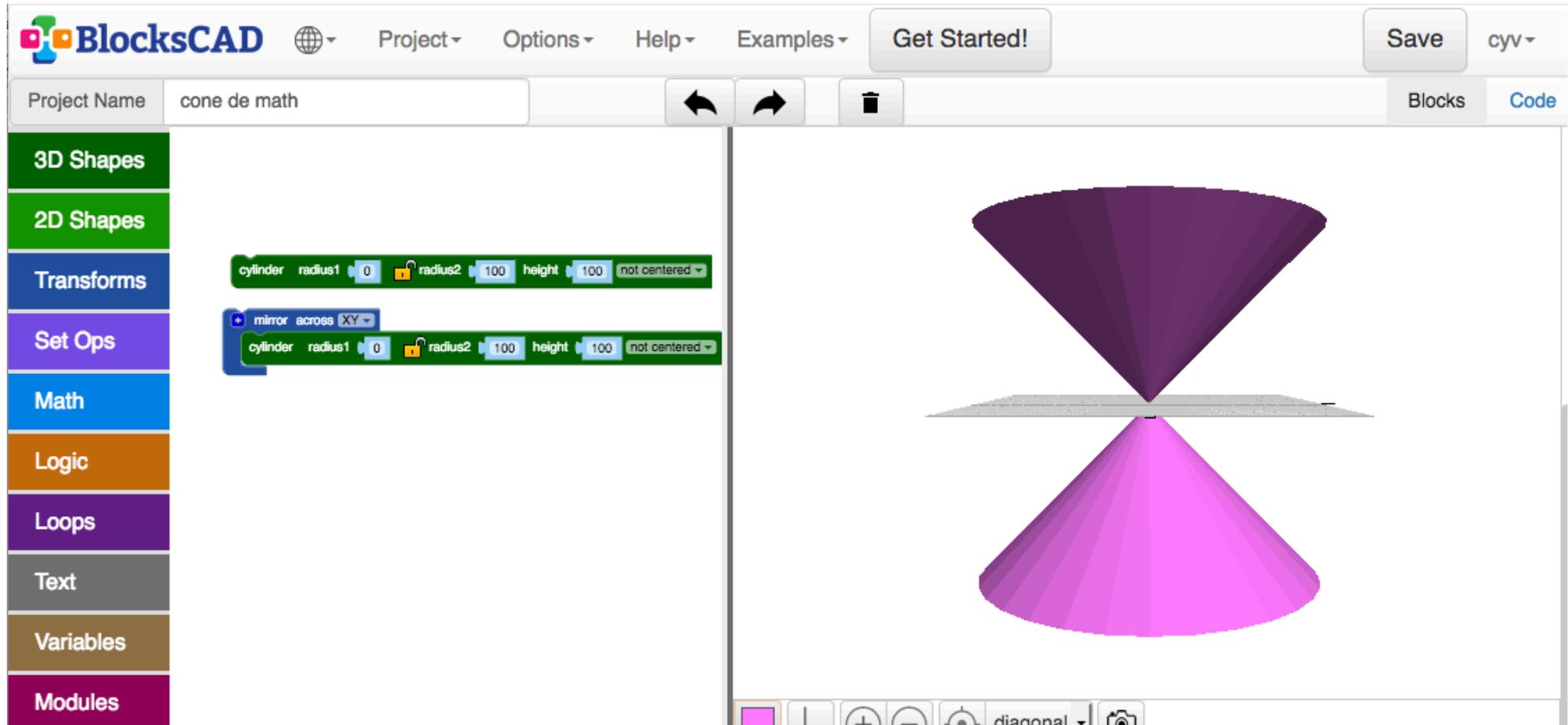
Intersection

- cylinder radius1: 100 radius2: 0 height: 100 centered
- with
 - color: yellow
 - rotate X: 0° Y: 75 Z: 0°
 - translate X: 0 Y: 0 Z: 0
 - cube X: 200 Y: 200 Z: 2 centered

Smooth: Low Medium High

L'hyperbole des mathématiciens

Souvent, pour les hyperboles, les dessins présentés montrent aussi la même courbe en symétrie. Cela est dû à la définition d'un cône pour un mathématicien qui est composé de 2 cônes physiques (le monde dans lequel nous sommes) symétriques et pointe à pointe.



Passons au stade suivant pour voir comment faire les mêmes courbes en équation.

Les équations des différentes coniques

Sans jouer avec les intersections, nous allons prendre la formule polaire, oublié dans un cours de mathématique et retrouvé sur Internet.

$$\text{RAYON} = (\text{paramètre}) / (1 - (\text{excentricité}) \times \text{COS} (\text{AZIMUT}))$$

Avec cette formule, l'origine est appelée foyer de la conique.

Si l'excentricité est :

- Supérieure à 1, c'est une hyperbole
- Egale à 1, c'est une parabole
- Inférieure à 1, c'est une ellipse. Avec un cas particulier si égale à 0 cela devient un cercle.

Remarque : ne pas commencer la boucle à 1 car pour la parabole, cela commence par une division par zéro ce qui n'est pas idéal !

Encore plus de courbes : Pierre Bézier et Cyril Grandpierre

Les courbes de Béziérs

Les courbes de Béziérs ont été décrites par Pierre Bézier dans les années 1950 pour dessiner des pièces d'automobiles. Elles sont aussi utilisées pour lisser les lettres des polices de caractères et sont présentes dans la plupart des logiciels de dessin.

Les courbes, car c'est une famille de courbes, se calculent suivant une même formule à partir de points. La courbe de Bézier d'ordre 3 nécessite de définir 4 points (et ainsi de suite). La courbe commence avec le premier point (on va l'appeler P_0) et se termine avec le dernier point (P_3).

Une fois P_0 et P_3 définis, on peut changer la forme de la courbe en simplement déplaçant les points P_1 et P_2 .

La courbe ne passe pas par les points intermédiaires : ceux-ci sont plus des aimants qui attirent la courbe (façon de parler).

Le block 2D Beziers3

La courbe de Bézier d'ordre 3 est la formule suivante :

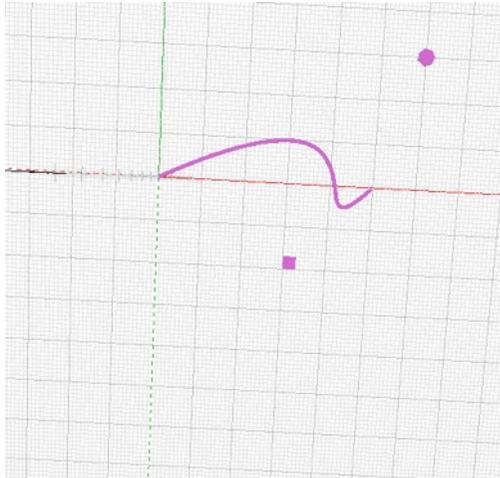
$$P0*(1-t)^3 + 3*P1*t*(1-t)^2 + 3*P2*t^2*(1-t) + P3*t^3$$
 et cela pour la variable (t) qui varie entre 0 et 1.

Cette formule doit se calculer pour toutes les abscisses des Points, c'est à dire P0x, P1x, P2x, P3x et aussi pour les ordonnées de ces points, c'est à dire P0y, P1y, P2y, P3y.

The screenshot displays the BlocksCAD interface. On the left, a sidebar lists categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a block titled '2D Beziers3 with: P0x, P0y, P1x, P1y, P2x, P2y, P3x, P3y, Di...'. The block's configuration is visible, showing a 'count with' loop from 0 to 1 by 0.01, and a 'do' loop containing 'translate X' and 'Y' blocks with numerical inputs for P0x, P1x, P2x, P3x and P0y, P1y, P2y, P3y. A 'circle radius 0.5' block is also present. A 'plus' block contains a 'Display' block set to 1, and another 'do' loop with 'translate X' and 'Y' blocks for P1x, P1y and P2x, P2y, followed by a 'square X 3 Y 3' block. A separate block titled '2D Beziers3 with:' lists the parameters: P0x: 0, P0y: 0, P1x: 60, P1y: 30, P2x: 30, P2y: -20, P3x: 50, P3y: 0, and Display: 1. On the right, a 3D view shows a grid with a purple curve and several points (P0, P1, P2, P3) marked with a small circle and a square. A 'Render' button and 'STL' export option are visible at the bottom right.

C'est un bloc qui dessine la courbe une fois après avoir choisi les points. Un paramètre « DISPLAY » permet de dessiner le point P1 (un petit cercle) et le point P2 (un carré) pour faciliter le déplacement des points pour avoir la courbe définitive.

Une fois la courbe désirée, il suffit de mettre zéro dans le paramètre display et la courbe est seule.



avec les points



sans les points

Le Bloc 2D Beziers4

Nous allons aussi définir la courbe de Bézier d'ordre 4, c'est à dire avec 5 points car elle permet des courbes avec des formes un peu plus élaborées.

La courbe de Bézier d'ordre 4 est la formule suivante :

$P0*(1-t)^4 + 4*P1*t*(1-t)^3 + 6*P2*t^2*(1-t)^2 + 4*P3*t^3*(1-t) + P4*t^4$ et cela pour la variable (t) qui varie entre 0 et 1.

The screenshot shows the BlocksCAD interface with the '2D Beziers4' block selected. The block configuration is as follows:

- to 2D Beziers4 with: P0x, P0y, P1x, P1y, P2x, P2y, P3x, P3y, P4x, P4y...
- union
- count with t from 0 to 1 by 0.01 (hull ✓)
- do
- translate X: $P0x - x + 4x - P1x$
- translate Y: $P0y - x + 4x - P1y$
- circle radius 0.5
- plus
- if Display == 1
- do
- union
- translate X: P0x, Y: P0y
- 2D text "0" size 5 font Roboto
- plus
- translate X: P1x, Y: P1y
- 2D text "1" size 5 font Roboto
- plus
- translate X: P2x, Y: P2y
- 2D text "2" size 5 font Roboto
- plus
- translate X: P3x, Y: P3y
- 2D text "3" size 5 font Roboto
- plus
- translate X: P4x, Y: P4y
- 2D text "4" size 5 font Roboto

The 2D grid view shows a purple curve defined by five control points (P0 to P4) on a coordinate system. The control points are located at (0,0), (75,20), (30,-20), (50,-5), and (80,0). The curve starts at P0 and ends at P4, passing through the points defined by the Bezier formula. The grid also shows a coordinate system with X and Y axes, and a 'diagonal' view option.

2D Beziers4 with:

P0x	0
P0y	0
P1x	75
P1y	20
P2x	30
P2y	-20
P3x	50
P3y	-5
P4x	80
P4y	0
Display	1

Render STL (Binary) Generate STL

Le Bloc 2D Beziers5

Nous allons aussi définir la courbe de Bézier d'ordre 5, c'est à dire avec 6 points.
La courbe de Bézier d'ordre 5 est la formule suivante :

$P0*(1-t)^5 + 5*P1*t*(1-t)^4 + 10*P2*t^2*(1-t)^3 + 10*P3*t^3*(1-t)^2 + 5*P4*t^4*(1-t) + P5*t^5$ et cela pour la variable (t) qui varie entre 0 et 1.

The screenshot displays the BlocksCAD software interface. The main workspace shows a 2D grid with a pink curve of order 5. The curve starts at point P0 (0,0) and ends at point P5 (56,19). The curve is defined by the following points: P0(0,0), P1(75,20), P2(30,30), P3(50,-20), P4(80,0), P5(56,19). The software interface includes a menu bar with options like Project, Options, Help, and Examples. A toolbar at the top right has buttons for Save, cyv, Blocks, and Code. The left sidebar contains buttons for 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The central workspace shows a block-based script for the 2D Bezier curve. The script includes a 'to 2D Beziers5' block, a 'union' block, a 'count with' loop, and a 'plus' block with a 'display' block. The 2D grid view shows the rendered curve with a coordinate system and axes. The bottom right of the grid view has a 'Render' button and a 'Generate STL' button.

Un profil d'aile

Assez simplement, en utilisant 2D Bezier5, on peut réaliser un profil d'aile et en faire une aile avec un LINEAR EXTRUDE.

Project Name: Aile

linear extrude height: 100 twist: 0° scale: x: 0.2 y: 0.2 not centered

2D Bezier5 with:

- P0x: 0
- P0y: 0
- P1x: 75
- P1y: 20
- P2x: 80
- P2y: -10
- P3x: 50
- P3y: -5
- P4x: 80
- P4y: 0
- P5x: 0
- P5y: 0
- Display: 0

Render

STL (Binary) Generate STL

Le triangle de Pascal

Comme nous sommes à peu près sûr qu'Edwin va nous poser la question, autant l'aborder maintenant. Mais pourquoi s'arrêter au block Bézier5 ? et si on veut un block Bézier 6, 7, 8,

Le principe est toujours le même : il faut réaliser la formule d'un polynôme de la forme version 8) :

$(1-t)^8 + t(1-t)^7 + t^2(1-t)^6 + t^3(1-t)^5 + t^4(1-t)^4 + t^5(1-t)^3 + t^6(1-t)^2 + t^5(1-t) + t^8$: quand la puissance de $(1-t)$ diminue celle de (t) augmente.

Mais il faut rajouter des coefficients (des nombres constants) devant chaque des multiplications. Les coefficients sont donnés par le triangle de Pascal.

Commençons par le niveau 1

Le niveau 2

Le niveau 3

Le niveau 4

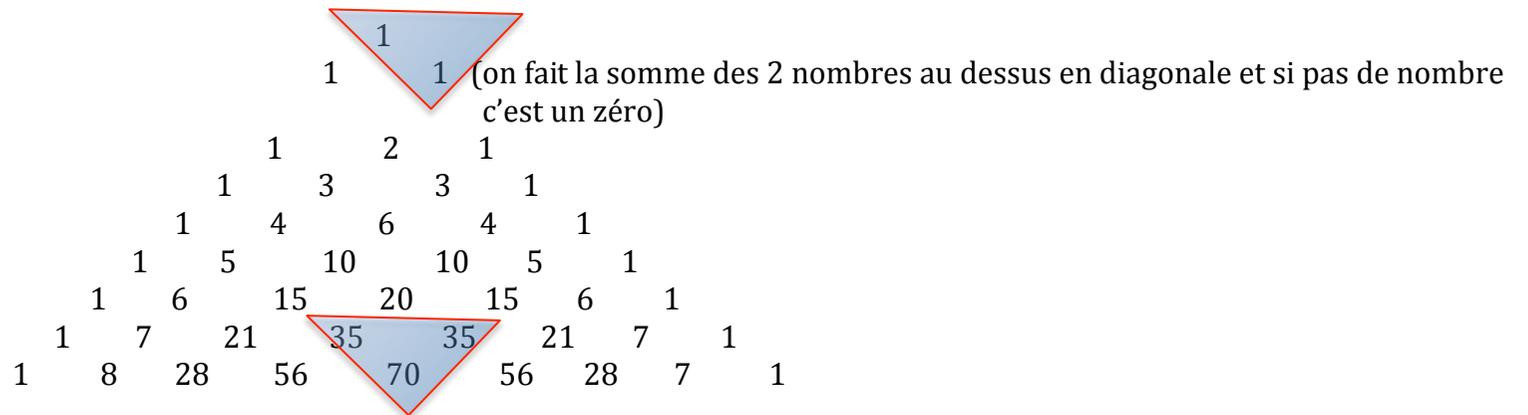
Le niveau 5

Le niveau 6

Le niveau 7

Le niveau 8

Le niveau 9



Vous avez compris le principe ? Tant mieux !

Voici donc la formule en y ajoutant les coefficients :

$$1*(1-t)^8 + 8*t*(1-t)^7 + 28*t^2*(1-t)^6 + 56*t^3*(1-t)^5 + 70*t^4*(1-t)^4 + 56*t^5*(1-t)^3 + 28*t^6*(1-t)^2 + 7*t^7*(1-t) + 1*t^8$$

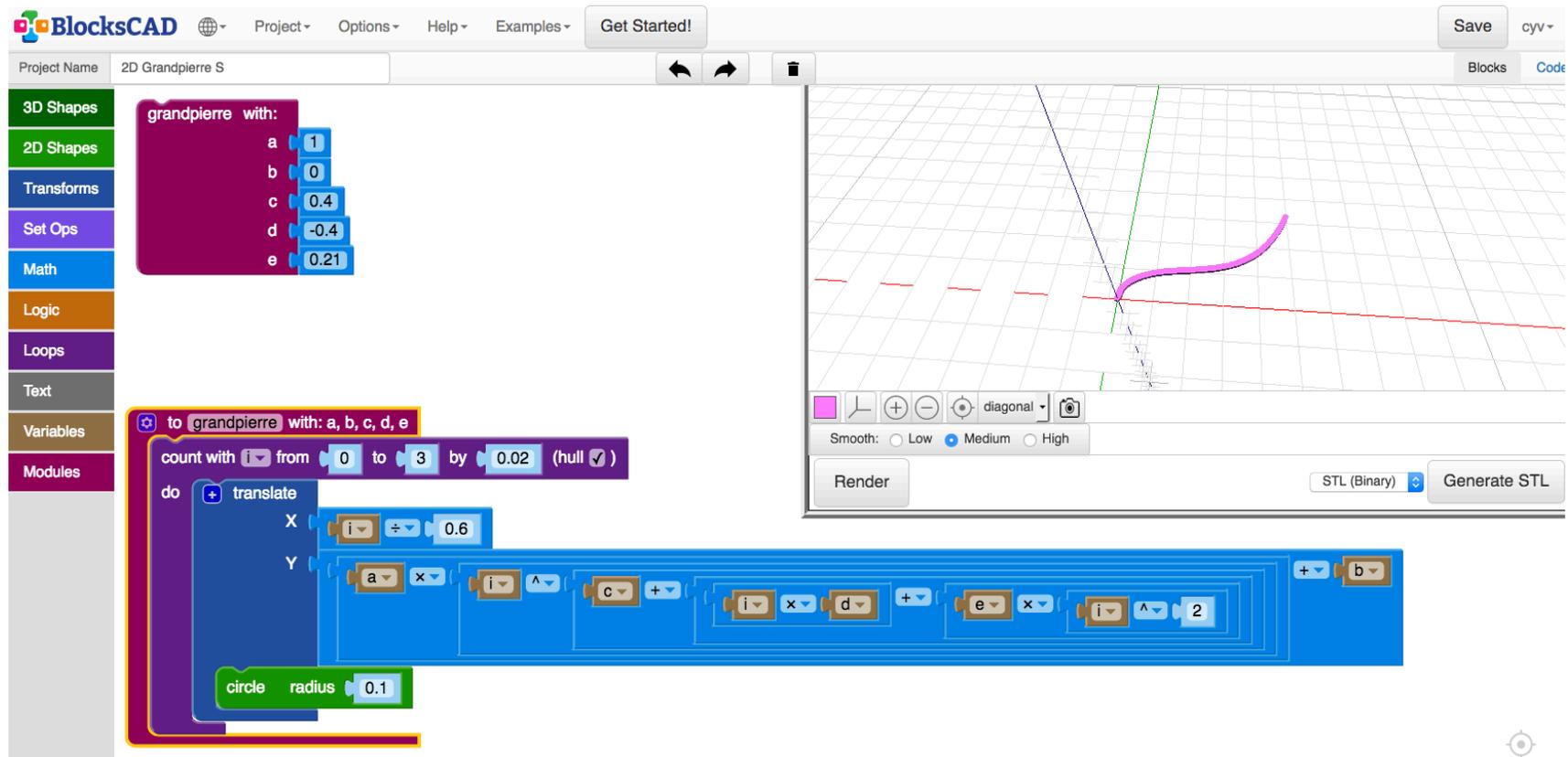
La méthode C. Grandpierre

Cyril Grandpierre est un architecte naval qui, entre autres, a publié dans *Loisirs Nautiques* en 1980 une méthode de tracé des carènes assez originale pour l'époque en utilisant une simple calculatrice.

Il existe maintenant des logiciels complets de tracé de carène, mais l'innovation de C. Grandpierre a été de proposer une famille de courbes sous la forme suivante :

$$Y = a * X \wedge (c + d * X + e * X \wedge 2) + b$$

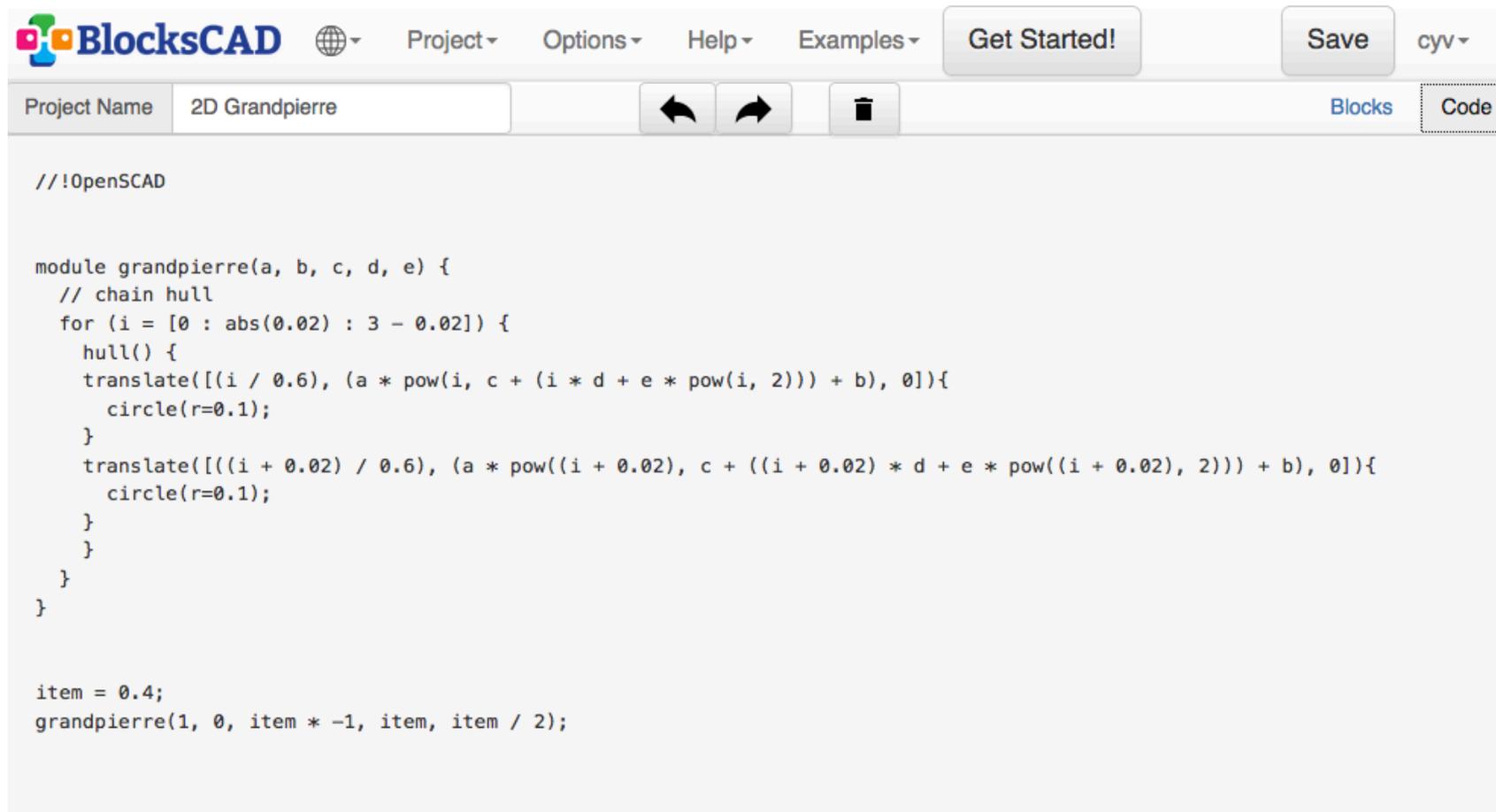
Et voilà un premier résultat. Pour vous, je ne sais pas, mais moi je trouve que cela ressemble bien à un demi maître-couple de bateau.



Il restera à étudier différentes valeurs des paramètres c, d, e.

BlocksCAD versus Openscad

Si vous regardez en haut à droite de l'écran, juste à coté du bouton « blocks », il y a un bouton « code ». Aventurons nous !



```
#!/OpenSCAD

module grandpierre(a, b, c, d, e) {
  // chain hull
  for (i = [0 : abs(0.02) : 3 - 0.02]) {
    hull() {
      translate([(i / 0.6), (a * pow(i, c + (i * d + e * pow(i, 2))) + b), 0]){
        circle(r=0.1);
      }
      translate([(i + 0.02) / 0.6), (a * pow((i + 0.02), c + ((i + 0.02) * d + e * pow((i + 0.02), 2))) + b), 0){
        circle(r=0.1);
      }
    }
  }
}

item = 0.4;
grandpierre(1, 0, item * -1, item, item / 2);
```

Ce sont les lignes de code informatique « OPENCAD », qui sont le reflet des blocks de BlocksCAD à la page suivante.

The screenshot shows the BlocksCAD interface. At the top, there is a navigation bar with the logo, a globe icon, and menu items: Project, Options, Help, Examples, and a 'Get Started!' button. Below this is a 'Project Name' field containing '2D Grandpierre' and navigation icons for back, forward, and delete. On the left, a vertical sidebar contains category buttons: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace displays two code blocks. The top block is a 'to grandpierre with: a, b, c, d, e' block containing a 'count with' block (from 0 to 3 by 0.02, hull checked), a 'do' block with a 'translate' block (X: 0.6, Y: a * i, c +, i * x, d +, e * x, i ^ 2) and a 'circle' block (radius 0.1). The bottom block is a 'set item to 0.4' block followed by a 'grandpierre with:' block containing five assignments: a: 1, b: 0, c: item * -1, d: item, and e: item + 2.

Vous pouvez les comparer et l'on retrouve le module « grandpierre », HULL, CIRCLE, TRANSLATE, les noms de variables. Seule la boucle LOOP change de nom et devient FOR.

Il n'y en a pas un qui est meilleur que l'autre, ils ont chacun leurs avantages. Nous trouvons que BlocksCAD reprend une interface plus facile à lire pour les enfants et bricoleurs du dimanche comme nous.

BlocksCAD permet d'avoir le code Openscad ... mais Openscad ne le propose pas. L'intérêt d'Openscad réside pour certains calculs un peu longs. Par exemple, nous n'avons pas réussi certains essais en faisant varier les paramètres. En Openscad, cela prend quelques secondes.

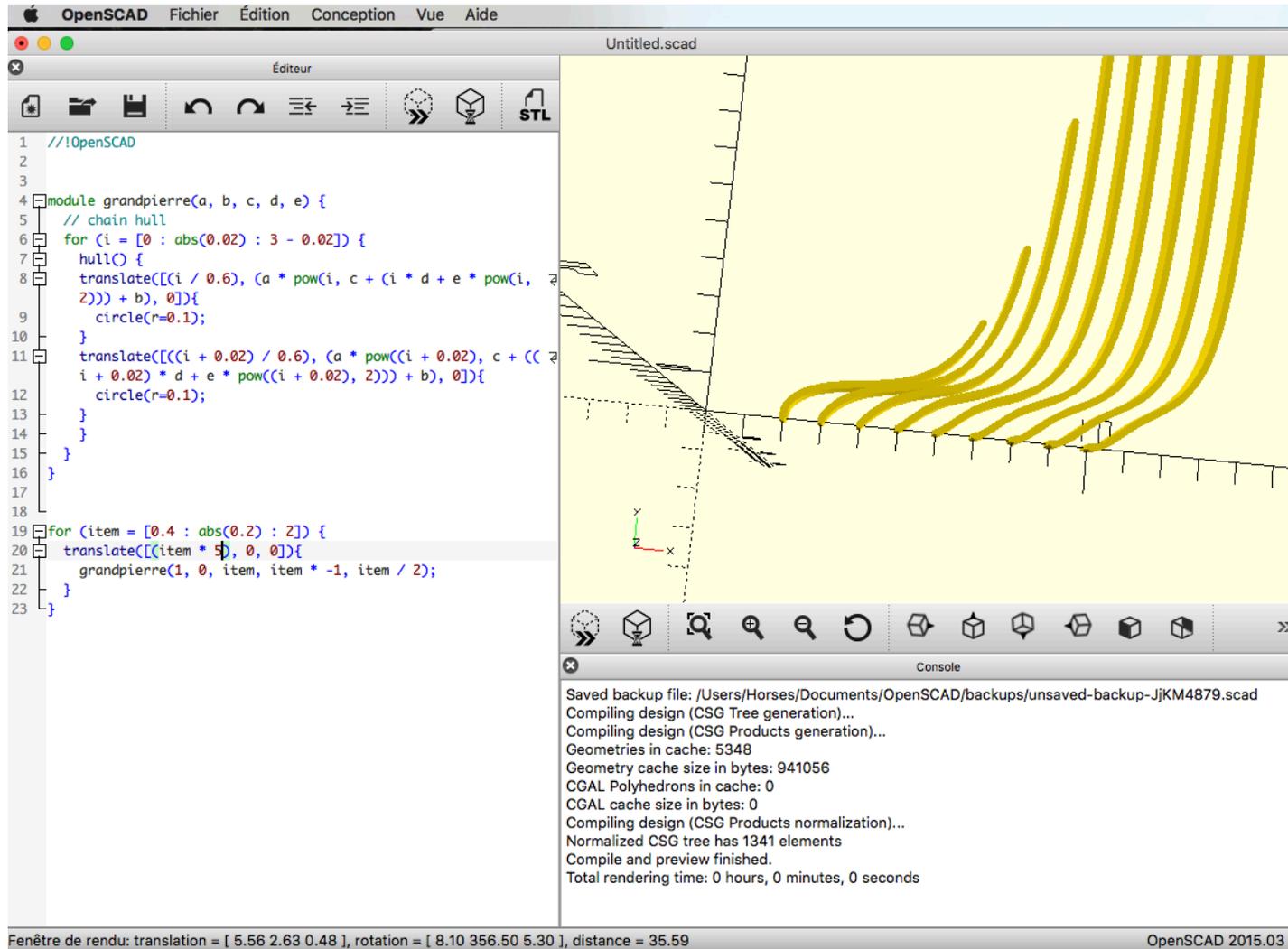
Cherchons donc à voir l'influence des paramètres sur la formule de Grandpierre. Voici notre code BlocksCAD.

The screenshot shows the BlocksCAD software interface. At the top, there is a navigation bar with the BlocksCAD logo, a globe icon, and menu items: Project, Options, Help, Examples, and Get Sta. Below this is a 'Project Name' field containing '2D Grandpierre' and navigation buttons (back, forward, delete). On the left side, there is a vertical sidebar with colored buttons for different categories: 3D Shapes (green), 2D Shapes (green), Transforms (blue), Set Ops (purple), Math (blue), Logic (orange), Loops (purple), Text (grey), Variables (brown), and Modules (purple). The main workspace contains a script starting with a 'to grandpierre with: a, b, ...' block. This is followed by a 'do' loop block containing a 'count with item' block (from 0.4 to 2 by 0.2) and a 'translate X' block (Y = 0). Inside the 'do' loop, there is a 'grandpierre with:' block with five parameters: a = 1, b = 0, c = item, d = item * -1, and e = item + 2.



Cliquons, sur code, sélectionnons toutes les lignes et « copier ». Sélectionnons Openscad (je vous laisse le soin de trouver comment l'installer sur votre ordinateur, c'est très bien expliqué sur le site Openscad <http://www.openscad.org/>)

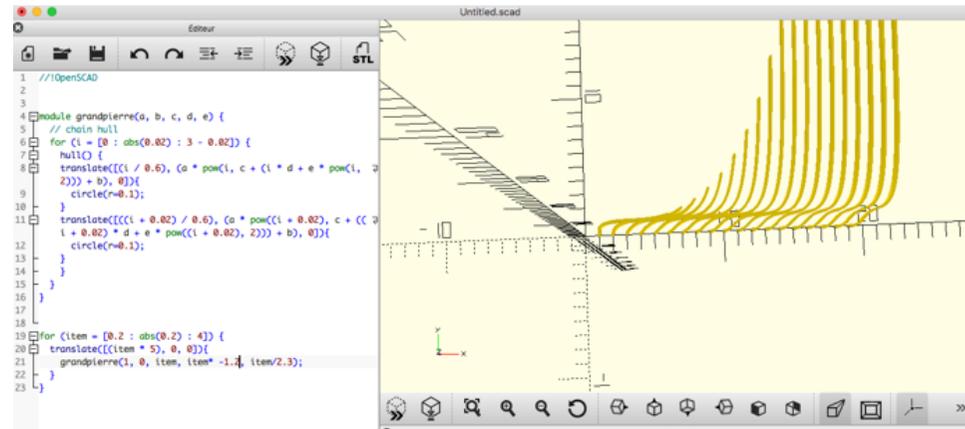
Une fois le code « collé » dans Openscad, il suffit de faire F5.



Ce qui peut prendre quelques dizaines de secondes avec BlocksCAD (prend moins d'une seconde sous Openscad, en tout cas avec mon ordinateur).

Pour moi, le principal inconvénient d'Openscad, ce sont ces formules qu'il faut se rappeler avec des accolades, crochets et parenthèses et dans le bon ordre s'il vous plaît !

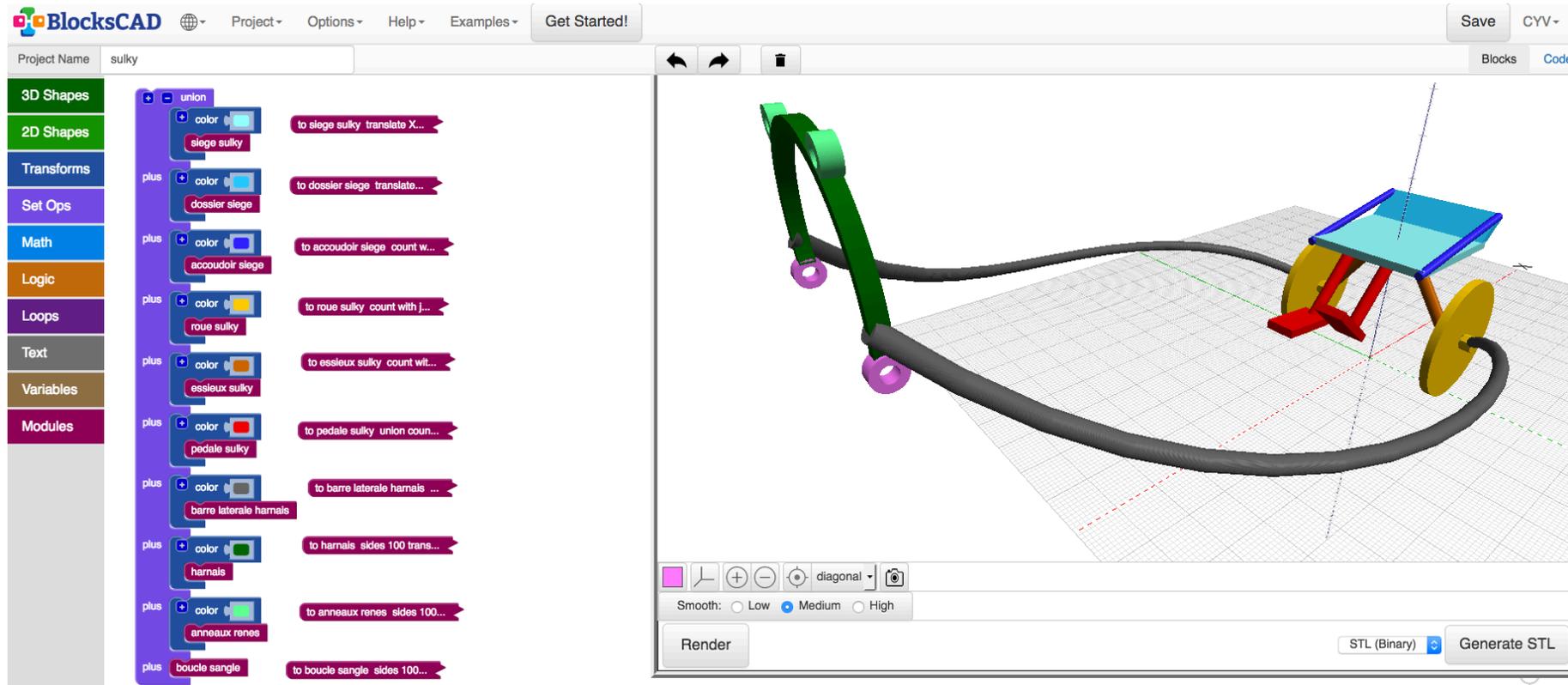
Et pour terminer, les résultats identiques d'un même programme.



Cela peut être utile de passer sous Openscad pour des programmes assez longs. Nous en avons fait un qui demandait plus de 2 heures sous Openscad !

Un sulky pour Duchesse des Myrtilles

Avant d'aller plus loin dans la géométrie et les mathématiques, faisons une pose avec un magnifique sulky proposé par Quitrie.



Voici le sulky complet. Il est fait à l'échelle pour une jument (à vous d'en deviner le nom) et aussi une figurine de cavalier un peu spéciale qui a obligé Quitrie à faire des pédales adaptées.

Chaque block a été fait sous la forme d'un module ce qui permet de s'y retrouver facilement.

Le siège du sulky

Le siège du Sulky est assez simple, une plaque horizontale, une plaque inclinée et deux cylindres comme accoudoirs.

The screenshot shows the BlocksCAD software interface. The project name is "sulky". The main workspace displays a 3D model of the sulky seat, which consists of a horizontal plate, an inclined plate, and two cylindrical armrests. The script in the workspace defines the construction of the seat:

- to siege sulky**: translate X 0 Y 0 Z 34, cube X 31 Y 29 Z 2 centered.
- to dossier siege**: translate X 17 Y 0 Z 37, rotate X 0° Y 50° Z 0°, cube X 2 Y 29 Z 10 centered.
- to accoudoir siege**: count with from -14 to 14 by 28 (hull), do: translate X 2 Y 37, rotate X 0° Y 80° Z 0°, sides 30, cylinder radius1 1 radius2 1 height 37 centered.

Les accoudoirs se font dans une boucle qui passe de -1 à 1 : sans souci.

Les roues

Les deux sont faites ensemble avec la fameuse boucle (-1,1). Chaque roue est composée :

- ❑ D'un disque avec un trou en son centre réalisé avec la fonction DIFFERENCE
- ❑ D'un axe, bien entendu plus petit que le trou du disque
- ❑ De deux cylindres, un bleu et un rouge, pour bloquer l'axe.

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', 'Examples', and 'Get Started!'. The project name is 'sulky'. The left sidebar shows a list of modules: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace is filled with a complex block-based script for creating a wheel. The script starts with a 'to siege sulky translate X...' block, followed by a 'plus' block with 'color' and 'siege sulky'. This is followed by a 'plus' block with 'color' and 'dossier siege translate...'. Then another 'plus' block with 'color' and 'accoudoir siege count w...'. The main part of the script is a 'to roue sulky' block containing a 'count with' loop from -1 to 1 by 2 (hull). Inside the loop, there is a 'do' block with several steps: 'translate X 0 Y 20 X Z 15', 'rotate X 90° Y 0° Z 0°', a 'difference' block (a large purple cylinder with radius 15 and height 2, minus a smaller purple cylinder with radius 1.5 and height 3), a 'minus' block (a large purple cylinder with radius 200), a 'sides' block (200), a 'color' block (green), a 'cylinder' block (radius 1, radius 2, height 4, centered), a 'translate X 0 Y 0 Z -2.5' block, a 'color' block (blue), a 'cylinder' block (radius 1, radius 2, height 2.5, centered), a 'translate X 0 Y 0 Z 2.5' block, a 'color' block (red), and a final 'cylinder' block (radius 1, radius 2, height 2.5, centered). On the right side of the interface, a 3D rendering window shows two purple wheels on a grid. The left wheel has a red axle and a blue axle block, while the right wheel has a blue axle and a red axle block. The rendering window includes a toolbar with 'diagonal' view, 'Smooth' settings (Low, Medium, High), and a 'Render' button. The output format is set to 'STL (Binary)'.

Essieux

C'est le nom donné au block mais ce ne sont pas vraiment des essieux mais plutôt des barres de soutènement des axes de roues.
Sans difficulté, deux cylindres bien orientés avec des TRANSLATE et des ROTATE.

The screenshot displays the BlocksCAD software interface. On the left, a sidebar lists various block categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a script for a block named 'essieux sulky'. The script is as follows:

```
to essieux sulky
  count with k from -1 to 1 by 2 (hull )
  do
    translate X 0 Y 10.5 x k Z 24
    rotate X 35 x k Y 0° Z 0°
    sides 200
    cylinder radius1 1.5 radius2 1.5 height 24 centered
```

On the right, a 3D rendering shows two yellow wheels with brown axles (essieux) passing through their centers. The axles are oriented at a 35-degree angle to the vertical. The scene is set on a grid with a coordinate system. Below the rendering, there are controls for camera view (diagonal), smoothness (Low, Medium, High), and a 'Render' button. The output format is set to 'STL (Binary)'.

Les pédales

The screenshot shows the BlocksCAD interface. The central workspace contains a block-based programming tree for a model named 'pedale sulky'. The tree starts with a 'union' block containing several sub-blocks: a 'color' block (blue), a 'count with' block (from -1 to 1 by 2), a 'do' loop containing a 'translate' block (X: -18, Y: 5, Z: 26), a 'rotate' block (X: 0°, Y: 40°, Z: 0°), a 'sides' block (200), and a 'cylinder' block (radius1: 1.5, radius2: 1.5, height: 20, centered). This is followed by a 'plus' block containing a 'color' block (green), a 'translate' block (X: -29, Y: -5, Z: 23), a 'rotate' block (X: 0°, Y: 40°, Z: 0°), a 'sides' block (200), and a 'cube' block (X: 15, Y: 7, Z: 2, centered). This sequence is repeated for a second 'plus' block with a 'color' block (red), a 'translate' block (X: -30, Y: 5, Z: 19), a 'rotate' block (X: 0°, Y: 5°, Z: 0°), a 'sides' block (200), and a 'cube' block (X: 15, Y: 7, Z: 2, centered). The 3D view window on the right shows the rendered model with a red base, two blue cylinders, and two green rectangular feet. The 'Render' button and 'Smooth' dropdown are visible at the bottom of the 3D view window.

On commence par deux cylindres en bleu, et deux repose-pieds. Ils ne sont pas inclinés du même angle car la figurine a les pieds décalés. Il faut savoir s'adapter à l'environnement.

Les barres de harnais

Elles sont réalisées à partir d'un simple cercle et d'un LINEAR EXTRUDE avec un TWIST. Il a fallu bien sur quelques réglages pour arriver aux bonnes dimensions.

The screenshot displays the BlocksCAD software interface. On the left, a sidebar lists various categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D model of two curved harness bars on a grid. The code editor on the right contains the following blocks:

- union** (purple block) containing:
 - color** (light blue) block with a color picker.
 - slège sulky** (grey) block.
 - to siege sulky translate X...** (pink) block.
 - plus** (purple) block containing:
 - color** (light blue) block.
 - dossier siege** (grey) block.
 - to dossier siege translate...** (pink) block.
 - plus** (purple) block containing:
 - color** (blue) block.
 - accoudoir siege** (grey) block.
 - to accoudoir siege count w...** (pink) block.
 - plus** (purple) block containing:
 - color** (yellow) block.
 - roue sulky** (grey) block.
 - to roue sulky count with j...** (pink) block.
 - plus** (purple) block containing:
 - color** (orange) block.
 - essieux sulky** (grey) block.
 - to essieux sulky count wit...** (pink) block.
 - plus** (purple) block containing:
 - color** (yellow) block.
 - pedale sulky** (grey) block.
 - to pedale sulky union colo...** (pink) block.
 - plus** (purple) block containing:
 - color** (grey) block.
 - barre laterale harnais** (red) block.
 - plus** (purple) block containing:
 - color** (green) block.
 - harnais** (grey) block.
 - to harnais sides 100 trans...** (pink) block.
 - plus** (purple) block containing:
 - color** (light green) block.
 - anneaux renes** (grey) block.
 - to anneaux renes sides 100...** (pink) block.
 - plus** (purple) block containing:
 - boucle sangle** (grey) block.
 - to boucle sangle sides 100...** (pink) block.
- to barre laterale harnais** (red block) containing:
 - count with** (m) from -1 to 1 by 2 (hull) (purple) block.
 - do** (purple) block containing:
 - translate** X -5 Y 0 Z 2 (blue) block.
 - rotate** X 0° Y 305° Z 0° (blue) block.
 - translate** X -6 Y (m) x 23 Z 70 (blue) block.
 - linear extrude** height 133 twist (m) x 203 scale: x 1 y 1 centered (blue) block.
 - translate** X 20 Y 0 (blue) block.
 - sides** 100 (blue) block.
 - circle** radius 1.3 (green) block.

Harnais et boucles

Et pour terminer, le harnais qui sera posé sur le dos du cheval avec 4 boucles, deux pour passer les rênes et deux pour attacher le cheval.

The screenshot shows the BlocksCAD software interface. The main workspace contains a complex block-based script for creating a harness (harnais) and rings (anneaux renes). The script is organized into several sections:

- to siege sulky translate X...**: A block for translating the harness to the saddle.
- to dossier siege translate...**: A block for translating the harness to the saddle.
- to accoudoir siege count w...**: A block for counting the saddle.
- to roue sulky count with j...**: A block for counting the wheels.
- to essieux sulky count wit...**: A block for counting the axles.
- to pedale sulky union colo...**: A block for unioning the pedals.
- to barre laterale harnais ...**: A block for the side bar.
- harnais**: A block for the harness.
- anneaux renes**: A block for the rings.
- boucle sangle sides 100...**: A block for the saddle girth.

The 3D preview window on the right shows the rendered model of the harness and rings. The harness is a green arch-shaped structure with two purple rings. The rings are positioned at the ends of the arch. The model is shown on a grid.

Nous ne montrons pas les 2 boucles en violet, mais à ce stade de la lecture, aucun souci pour les réaliser soi-même.

Enfin voici le résultat complet après impression 3D. Vous reconnaissez la jument ?



Un peu de géométrie

Dans ce chapitre, nous allons réaliser les dessins de géométrie de CM1 et CM2 qu'il faut faire avec une règle et un compas. Pour cela, il nous faut réaliser un module COMPAS et un module REGLE.

Le module REGLE

Pour le module REGLE, il nous faut le point de départ P1 et le point d'arrivée P2. Bien sûr P0 et P1 ont chacun la coordonnée X et la coordonnée Y. Maintenant, il nous faut avoir la formule pour un segment de droite.

Coordonnée suivant X. $x = P0x + (P1x - P0x)*t$ (t) est la variable pour tracer la courbe et qui bouge entre 0 et 1.

Nous voyons que $x = P0x$ lorsque la variable (t) vaut 0, et x vaut P1x lorsque la variable (t) égal 1.

Coordonnée suivant Y, c'est la même chose que pour X mais en prenant les valeurs suivants Y. $y = P0y + (P1y - P0y)*t$

The screenshot shows the BlocksCAD interface with the 'REGLE' module code and a 3D rendering of a cylinder. The code is as follows:

```
to REGLE with: P0x, P0y, P1x, P1y, epaisseur
  count with t from 0 to 1 by 0.01 (hull ✓)
  do
    translate
      X P0x + t * (P1x - P0x)
      Y P0y + t * (P1y - P0y)
      Z 0
    sphere radius epaisseur
  color yellow
end
```

Two instances of the 'REGLE' module are shown with the following parameters:

Instance	P0x	P0y	P1x	P1y	epaisseur
1	10	30	30	50	0.5
2	0	10	30	5	0.5

The 3D rendering shows a cylinder on a grid. A blue dashed line represents the path of the cylinder's center, and a red dashed line represents the path of the cylinder's radius. A yellow cylinder is shown at the end of the path, and a green cylinder is shown at the beginning of the path.

Notre module REGLE est une boucle de la variable (t) de 0 à 1 en reprenant nos deux formules

Le module REGLE GRADUEE : le théorème de Pythagore

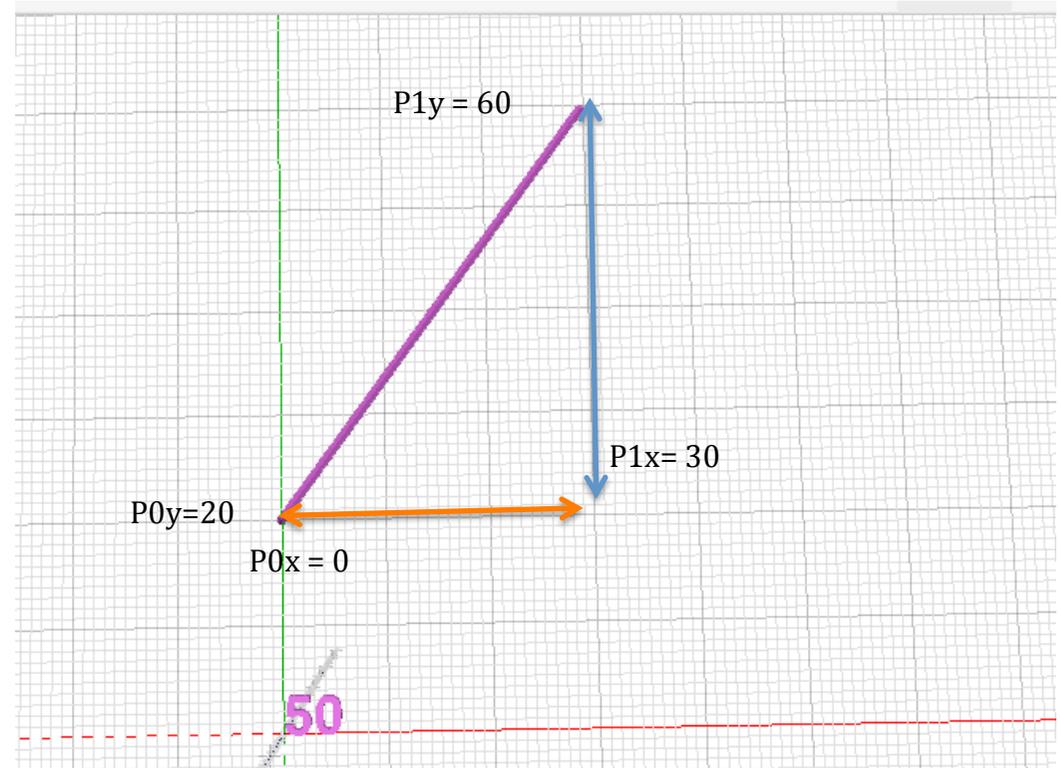
Maintenant, il serait bien que notre règle puisse mesurer la taille du segment. Pour cela nous allons utiliser le théorème de Pythagore. Reprenons un trait en violet fait avec notre REGLE.

Nous avons indiqué dessus les points :

- ❑ P0, avec P0x et P0y
- ❑ P1, avec P1x et P1y

En plus nous avons dessiné :

- ❑ Une flèche orange dont la longueur est $(P1x-P0x)$ c'est à dire 30 dans notre exemple
- ❑ Une flèche bleue, dont la longueur est $(P1y-P0y)$, c'est à dire 40 dans notre exemple



Et votre œil de lynx, désormais attentif, vient de reconnaître un triangle rectangle avec ces deux flèches et notre trait violet ? Parfait, vous êtes trop fort !

Que nous dit ce fameux théorème de Pythagore (allez voir sur Wikipédia pour l'histoire de ce théorème). Il nous dit que :

Le carré de la distance bleu plus le carré de la distance orange égale le carré de la distance violette

C'est à dire :

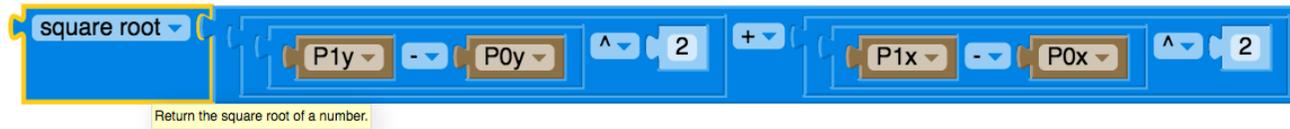
$$\text{(Distance violette)}^2 = (P1x-P0x)^2 + (P1y-P0y)^2$$

Reprenons notre exemple : $30*30 + 40*40 = 900+1600 = 2500 = 50*50$, donc la distance du trait violet est de 50. 50 est aussi appelé racine carré de 2500.

Racine carré ou Square Root

Autant il est assez facile de trouver la racine carré de 64 (c'est 8 car $8*8 = 64$), de 49 (c'est 7 car $7*7=49$), de 16, 9, 81 et quelques autres car nous connaissons parfaitement nos tables de multiplications, autant pour 27, 32, 48, ... et les plupart des autres nombres c'est plus compliqué.

BlocksCAD est gentil avec nous, il donne une fonction mathématique Square Root (pour Racine Carré en français) qui trouve le nombre quand on connaît son carré. Voyez ce que cela donne en formule BlocksCAD.



et voici le block avec la formule.

Project Name: REGLE GRADUEE

Script:

```
to REGLE GRADUEE with: P0x, P0y, P1x, P1y, epaisseur, longueur
  union
    count with t from 0 to 1 by 0.01 (hull ✓)
    do
      translate X (P0x + t * (P1x - P0x)) Y (P0y + t * (P1y - P0y)) Z 0
      sphere radius epaisseur
  plus
    if longueur == 1
      do
        3D text square root (P1y - P0y) ^ 2 + (P1x - P0x) ^ 2 size 5 font Roboto thickness 1
end
```

Variables:

REGLE GRADUEE with:	
P0x	0
P0y	26
P1x	32
P1y	57

3D View: 44.553338819890925

Il y a encore un petit point à améliorer, car il y a beaucoup de chiffres après la virgule.

2 chiffres après la virgule

Pour finaliser notre block, nous voulons écrire uniquement 2 chiffres après la virgule.

1. Prenons un nombre : 44, 55784325
2. Multiplions le par 100 : 4455, 784325
3. Supprimons tous les chiffres après la virgule : 4455 ; dans BlocksCAD c'est la fonction ROUND
4. Divisons le par 100 : 44, 55

Si nous voulons 3 chiffres après la virgule, on multiplie par 1000 ... puis on divisera par 1000 et ainsi de suite.

The screenshot shows the BlocksCAD software interface. The main workspace displays a 3D grid with a purple line drawn at an angle. The text '44.55' is visible on the grid, indicating the angle of the line. The interface includes a menu bar with options like 'Project', 'Options', 'Help', and 'Examples'. A toolbar at the top right contains 'Save' and 'cyv' buttons. The left sidebar shows various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a custom block named 'REGLE GRADUEE' with parameters: P0x, P0y, P1x, P1y, epaisseur, and longueur. The block is currently set to draw a line with a length of 1 and an angle of 57 degrees. The 3D view shows a purple line drawn on a grid, with the text '44.55' indicating the angle. The interface also includes a 'Render' button and a 'Generate STL' button.

Et voilà la fin de notre block pour tracer des traits simplement et connaître leur longueur ... ou pas.

Le module COMPAS

Le module COMPAS doit nous permettre : a) de donner le centre, là où il faut mettre la pointe du compas ; b) de donner le rayon ; c) de donner le début du cercle et la fin du cercle à tracer

The screenshot shows the BlocksCAD interface with the COMPAS module code and a 3D rendering of a circle arc. The code is as follows:

```
to COMPAS with: Ox, Oy, Rayon, Debut, Fin, Centre, epaisseur
+ translate X Ox Y Oy Z 0
+ union
+ count with azimuth from Debut to Fin by 1 (hull ✓)
do
+ rotate X 0° Y 0° Z azimuth
+ translate X rayon Y 0 Z 0
sphere radius epaisseur
plus
+ if Centre = 1
do
cylinder radius1 1 radius2 1 height 1 not centered
+ color
```

Two variable lists are shown below the code:

COMPAS with:	Value
Ox	20
Oy	10
Rayon	20
Debut	45°
Fin	135°
Centre	1
epaisseur	0.5

COMPAS with:	Value
Ox	0
Oy	30
Rayon	30
Debut	-20
Fin	30°
Centre	1
epaisseur	0.5

The 3D rendering shows a grid with a circle arc drawn in purple and yellow. A pink cube is positioned at the center of the arc, and a yellow cube is at the end of the arc. The interface includes a 'Render' button and a 'Smooth' dropdown menu set to 'Medium'.

Le module comprend donc comme variables : Ox, la coordonnée X du centre, Oy, la coordonnée Y du centre, le rayon, l'angle de début et l'angle de fin, un code pour visualiser le centre (s'il est à 1, le centre est imprimé) et l'épaisseur du trait.

Pour le module, nous nous sommes inspirés :

- du module réalisant un arc de cercle (le haut du block)
- un petit IF (test) pour voir si CENTRE était égal à 1 et, si oui, un petit cylindre pour dessiner le centre
- le tout TRANSLATE de Ox et de Oy

Une première rosace

Maintenant que nous avons un compas, nous allons essayer de dessiner une rosace.

The screenshot shows the BlocksCAD interface with a project named "ROSACE". The left sidebar contains a menu with categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace displays several "COMPAS with:" blocks with the following parameters:

- Block 1: Ox: 0, Oy: 0, Rayon: 20, Debut: 0, Fin: 360, Centre: 0, epaisseur: 0.5
- Block 2: Ox: 20, Oy: 0, Rayon: 20, Debut: 120, Fin: 240, Centre: 0, epaisseur: 0.5
- Block 3: Ox: -20, Oy: 0, Rayon: 20, Debut: -60, Fin: 60, Centre: 0, epaisseur: 0.5
- Block 4: Ox: 0, Oy: -20, Rayon: 20, Debut: 30, Fin: 150, Centre: 0, epaisseur: 0.5
- Block 5: Ox: 0, Oy: 20, Rayon: 20, Debut: 210, Fin: 330, Centre: 0, epaisseur: 0.5

Additional blocks include "to REGLE GRADUEE with: P0x,..." with a value of 15°, and "to COMPAS with: Ox, Oy, Ray..." with a note "Creates a function with no output." The right panel shows a 3D view of the resulting rose pattern on a grid, with a "Render" button and "Smooth" options (Low, Medium, High) at the bottom.

Quelques traits de compas et nous obtenons facilement une première rosace.

Une deuxième, un peu plus achevée en utilisant une LOOP.

The screenshot displays the BlocksCAD interface for a project named "ROSACE 2". The left sidebar shows a menu with categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains a script with the following blocks:

- A purple block: "to COMPAS with: Ox, Oy, Ray..."
- A purple block: "COMPAS with:" containing:
 - Ox: 0
 - Oy: 0
 - Rayon: 20
 - Debut: 0
 - Fin: 360
 - Centre: 0
 - epaisseur: 0.5
- A purple block: "count with i from 0 to 360 by 60 (hull)"
- A purple block: "do" containing:
 - A purple block: "COMPAS with:" containing:
 - Ox: $20 \times \cos i$
 - Oy: $20 \times \sin i$
 - Rayon: 20
 - Debut: $i + 120$
 - Fin: $i + 240$
 - Centre: 0
 - epaisseur: 0.5

The right side of the interface shows a 3D grid with a rendered purple rose curve. Below the grid is a toolbar with icons for zooming, rotating, and rendering. The "Smooth" options are set to "Medium". At the bottom right, there are buttons for "Render" and "Generate" (with a dropdown menu set to "STL (Binary)").

Le module ROSACE

Nous allons généraliser le précédent programme pour facilement dessiner une Rosace en indiquant seulement son rayon et le nombre de pétales. Dans le dessin précédent, il y avait 6 pétales.

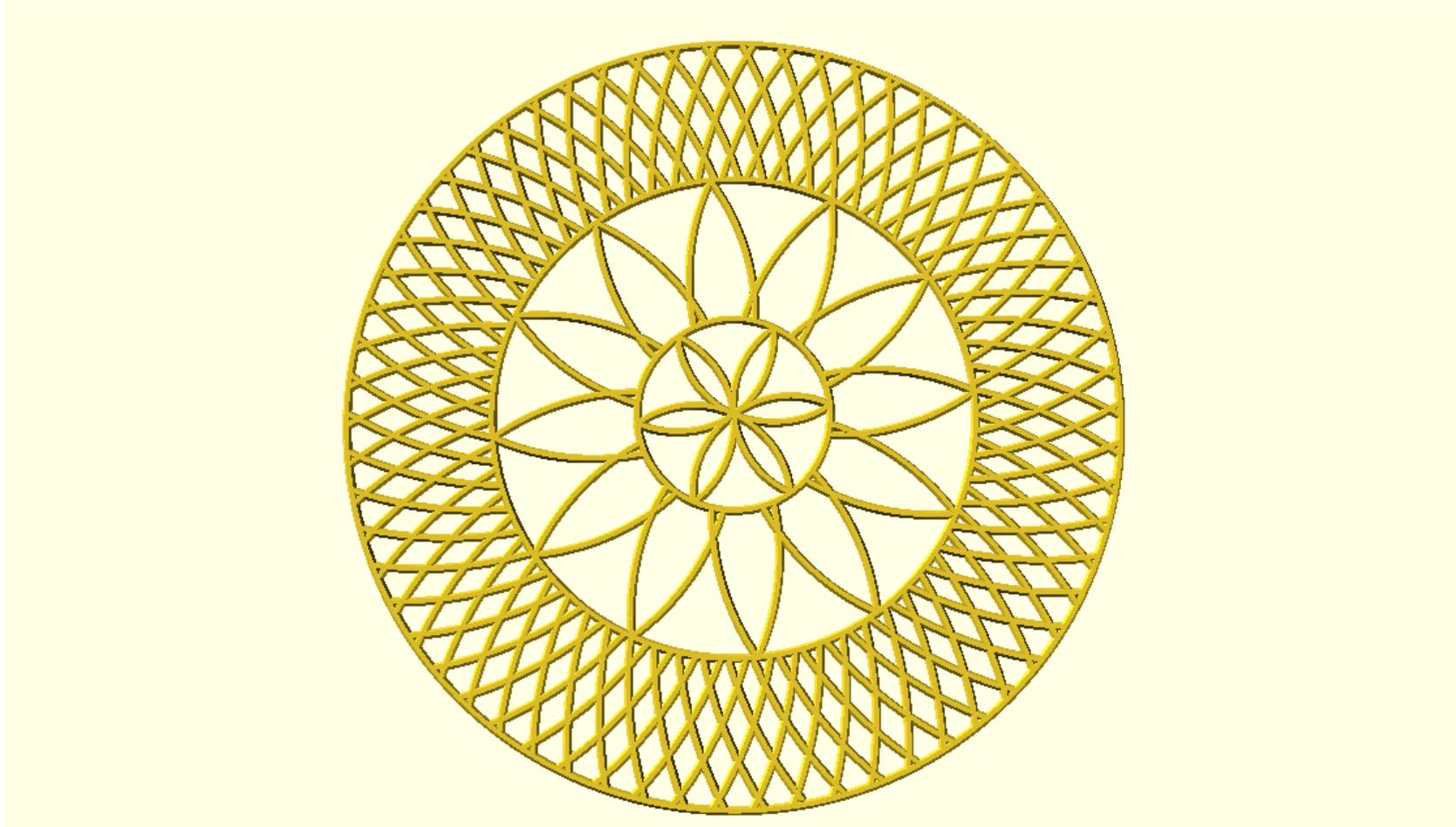
The screenshot shows the BlocksCAD interface with a script for drawing a rose. The script is as follows:

```
to COMPAS with: Ox, Oy, Rayon...  
  ROSACE with:  
    Rayon 40  
    nombre 24  
  
to ROSACE with: Rayon, nombre  
  count with i from 0 to 360 by 360 ÷ nombre (hull )  
  do  
    COMPAS with:  
      Ox rayon x cos i  
      Oy rayon x sin i  
      Rayon rayon  
      Debut i + 120  
      Fin i + 240  
      Centre 0  
      epaisseur 0.5
```

The right side of the interface shows a 3D rendering of a purple rose with 24 petals. The rendering is smooth and has a 'Render' button and 'Generate STL' button at the bottom.

Le programme est une boucle composée d'arc de cercle. Le centre de chaque arc de cercle (O_x, O_y) est disposé sur un cercle externe de la rosace. On retrouve la formule avec le cosinus et le sinus que nous avons déjà vue. NOMBRE est utilisé pour les pétales : Comme un cercle fait 360° , il suffit de régler le PAS (by de la formule LOOP) sur $360/\text{nombre}$.

Pour finir, une rosace multiple constituée de plusieurs rosaces, réalisée en OPENSCAD car le temps de calcul est vraiment plus court.



Quasimodo n'a qu'à bien se tenir !

Les fonctions mathématiques de BlocksCAD

Ce chapitre va passer en revue les différentes fonctions et possibilités offertes par BlocksCAD.

BlocksCAD nous propose une série de fonctions mathématiques :

- SQUARE ROOT
- ABSOLUTE
- (le signe moins) -
- LN
- Log10
- e^{\wedge}
- 10^{\wedge}

Racines carrées et valeur absolue

Square Root, nous l'avons déjà vu est la racine carrée d'un nombre. Appelons le X.

$$\text{Square Root } (X) \wedge 2 = X$$

Cette fonction est souvent dessinée sous la forme suivante \sqrt{x} . Elle est utilisée dans de nombreuses formules.

The screenshot shows the BlocksCAD software interface. On the left, a script is visible with the following blocks: a 'color' block (yellow), a 'count with' block (from 0.01 to 10 by 0.1), a 'do' loop containing a 'translate' block (X: i, Y: 10, Z: 0) and a 'sphere' block (radius 0.5). A dropdown menu for the 'square root' block is open, showing options: 'square root', 'absolute', '-', 'ln', 'log10', 'e^', and '10^'. On the right, a 3D plot shows a curve of yellow spheres representing the square root function \sqrt{x} for x from 0.01 to 10. A red horizontal line is drawn at Z=0, and a green vertical dashed line is at X=0. A red 'X' mark is visible on the right side of the plot area.

La valeur absolu d'un nombre est : ce nombre s'il est positif, l'opposé s'il est négatif, ou dit autrement s'il y a un plus devant le nombre on le laisse, s'il y a un moins on le remplace par un plus.

C'est très utile en informatique car on ne sait pas toujours si le résultat d'une formule va être positif et donc dans le doute on prend la valeur absolue, surtout si on utilise SQUARE ROOT. En effet, SQUARE ROOT donne une erreur sur un nombre négatif.

Exponentiel

La fonction exponentielle qui se note « e^x » ou en BlocksCAD « e^\wedge » est une fonction mathématique très utilisée et donc présente dans tous les programmes d'ordinateurs.

A quoi elle sert en maths ? Vous verrez cela avec votre professeur de maths. Ce qui nous intéresse, c'est quelle existe et que de nombreuses courbes l'utilisent.

Sur le dessin suivant nous avons indiqué la fonction exponentielle seule et une autre assez classique.

The screenshot shows the BlocksCAD software interface. The project name is "Exponentiel". The left sidebar contains categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. Two code blocks are visible:

- The top block is a loop starting with "count with" from -10 to 10 by 0.1 (hull checked). It contains a "do" loop with a "translate" block (X: i, Y: 5, Z: 0) and a "sphere" block (radius: 0.5). The Y coordinate is calculated as $50 \div (1 + e^\wedge i)$.
- The bottom block is a loop starting with "count with" from -10 to 10 by 0.1 (hull unchecked). It contains a "do" loop with a "translate" block (X: i, Y: 5, Z: 0) and a "sphere" block (radius: 0.5). The Y coordinate is calculated as $e^\wedge i \div 2$.

The right side of the interface shows a 3D plot on a grid. Two curves are plotted:

- A red curve representing the function $Y = 1 / (1 + e^x)$.
- A yellow dotted curve representing the function $Y = e^x$.

The plot includes a "Render" button and a "Smooth" dropdown menu (Low, Medium, High) with "Medium" selected. A "Generate S" button is also visible at the bottom right.

Occupons nous de la courbe jaune : c'est celle de l'exponentielle. Nous n'avons pas mis l'option HULL pour que l'on voit bien les différents points. Ceux de gauche sont très serrés et plus nous avançons vers la droite plus ils sont écartés. Cette fonction est une fonction de croissance et dont la croissance accélère de plus en plus.

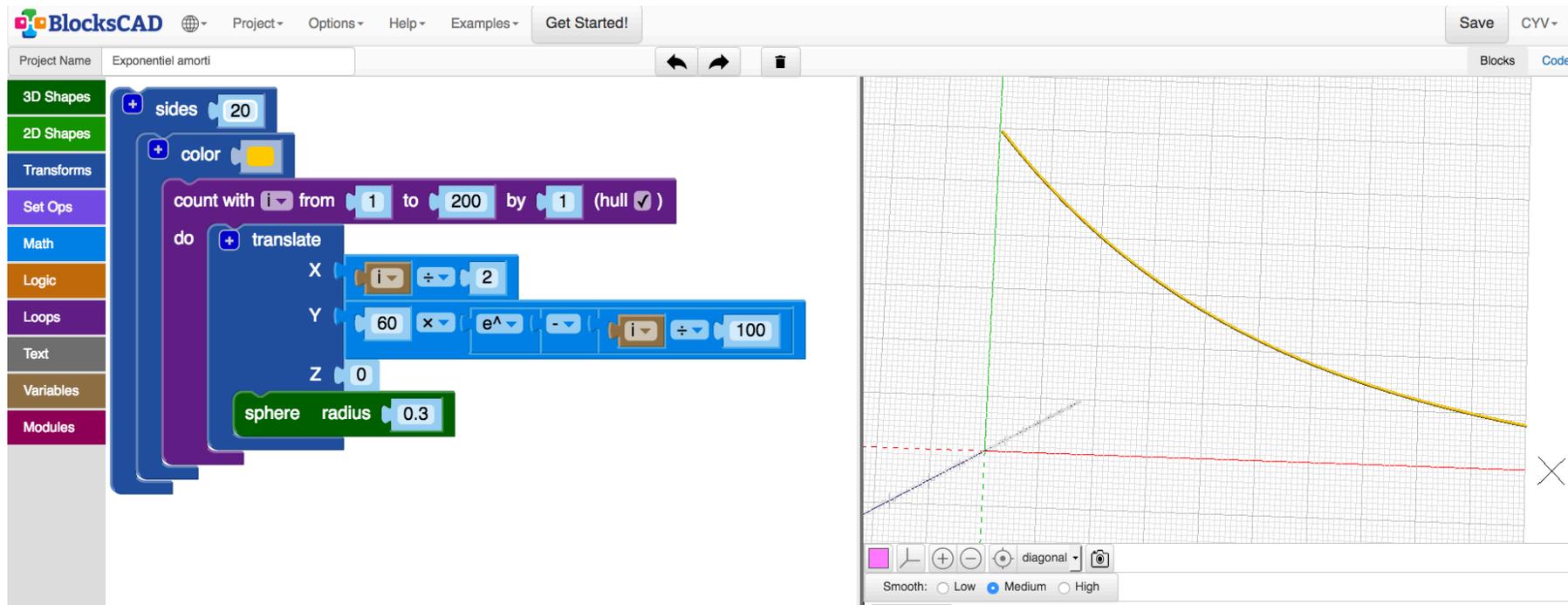
C'est pour cette raison que dans la vie courante, on dit d'une chose qui grandit de plus en plus vite qu'elle grandit de façon exponentielle.

La courbe rouge est une des nombreuses courbes déduites de la courbe de base mais elle est intéressante pour faire voyons ... voyons ? un toboggan ? ou un moule à Kouglouf ? à vous de voir.

Mais surtout n'ayez pas peur de l'exponentiel, elle est très pratique et en plus c'est BlocksCAD qui fait tous les calculs.

Exponentiel décroissante

L'exponentiel peut aussi être utilisé en tant que courbe décroissante en utilisant le signe - dans l'exposant. Voici un petit exemple ci-dessous.



Une courbe d'amortissement

Voici, la précédente courbe multipliée par une courbe sinusoïdale.

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', and 'Examples', along with a 'Get Started!' button. The project name is 'Exponentiel amorti'. The left sidebar shows various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules.

The script on the left consists of the following blocks:

- sides**: 20
- color**: Yellow
- count with**: from 1 to 200 by 0.5 (hull checked)
- do** loop containing:
 - translate** block with X: $i \div 2$, Y: $40 \times e^{-i \div 100} \times \sin(20 \times i)$, and Z: 0
 - sphere** block with radius 0.3

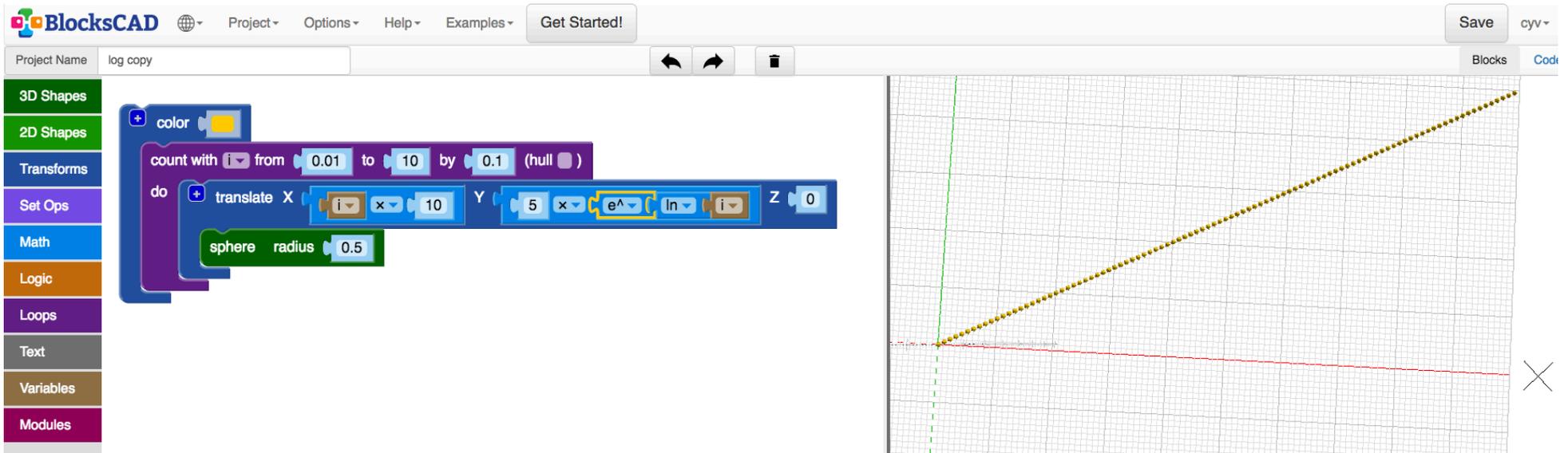
The right side of the interface shows a 3D plot of a yellow damped sine wave on a grid. The wave starts with a large amplitude and gradually decreases in amplitude as it oscillates. Below the plot, there are controls for 'Smooth' (Low, Medium, High) and a 'Render' button. At the bottom right, there is a 'Generate STL' button with a dropdown menu set to 'STL (Binary)'.

Logarithme et exponentiel

La fonction Logarithme (LN) est la fonction inverse de Exponentielle, c'est à dire que :

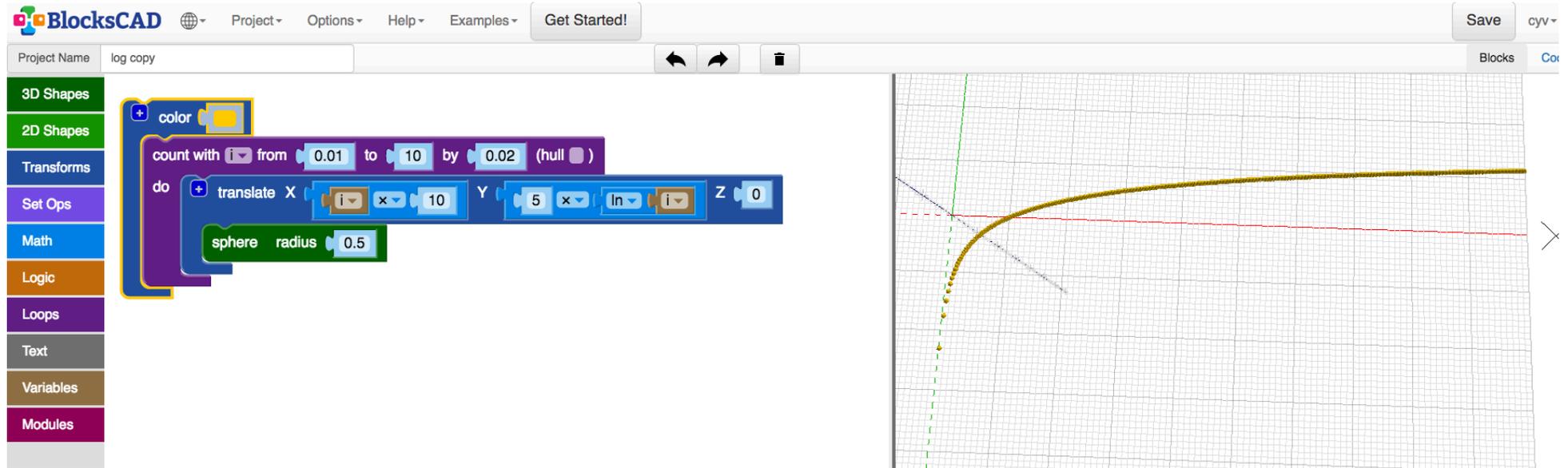
$$\text{LN} (e^x) = e^{\text{LN}(x)}=X$$

Regardez le dessin ci-dessous. Il montre bien que LN « annule » Exponentiel. La formule de $Y = 5 * e^{\text{LN} (i)}$ est la même chose que $Y=5*i$



Logarithme

Passons maintenant à la courbe de la fonction Logarithme. Elle est indiquée ci-dessous. C'est aussi une fonction croissante mais qui croît de plus en plus lentement au fur et à mesure que l'on va vers la droite : elle se fatigue vite 😊



C'est un peu comme la fonction exponentielle, Logarithme est utilisé dans de nombreuses autres fonctions donc autant savoir qu'il existe.

Pour l'anecdote, la fonction logarithme possède une propriété très surprenante dont voici un exemple : $\text{LN}(3) + \text{LN}(5) = \text{LN}(15)$, c'est à dire que la somme de 2 logarithme est égale au logarithme du produit de ces eux nombres.

Cette propriété a été utilisée aux temps préhistoriques pour fabriquer des règles (en bois !) à calcul. C'était avant le temps des calculatrices, donc voyez avec votre grand-père ou votre arrière grand-père pour une éventuelle explication.

Une dernière chose : on voit aussi écrit LOG à la place de LN et dans BlocksCAD il y a aussi LOG10. LN est pour Logarithme népérien qui est le plus utilisé. Mais la famille des logarithmes ne s'arrête pas là et LOG10 en est une autre variante qui correspond aux puissances de 10 (10^{\wedge}).

Pour la suite voyez aussi vos professeurs de Mathématiques.

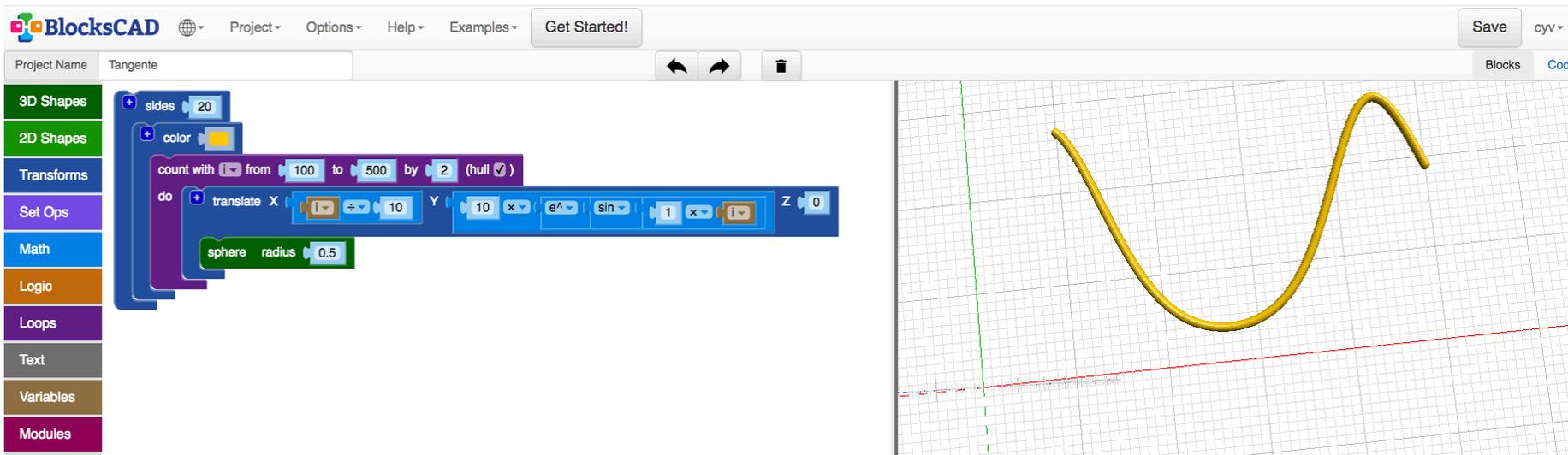
Prenons la tangente

« La [tangente](#) est une [droite](#) ayant un point de contact avec une [courbe](#) et qui fait un [angle](#) nul avec elle en ce point. » Nous dit Wikipédia.

Maintenant, essayons de voir ce que cela signifie :

- d'abord, il faut une courbe
- ensuite, il faut un point sur la courbe
- pour terminer, il faut une droite qui passe par le point de la courbe. Mais par n'importe quelle droite.

Commençons par la courbe. Une série de points dans une boucle (LOOP) avec une fonction un peu n'importe quoi : une exponentielle de sinus. Pour l'exemple, cela n'a pas grande importance.



Maintenant, nous allons mettre un point sur cette courbe. Prenons la valeur de $(i) = 230$. Positionnons notre point en réalisant une copie de la formule de la courbe.

The image shows a Scratch code editor with a project named "Tangente". The code is as follows:

```
3D Shapes
2D Shapes
Transforms
Set Ops
Math
Logic
Loops
Text
Variables
Modules

+ sides 20
+ color yellow
count with i from 100 to 500 by 2 (hull checked)
do
+ translate X (i / 10) Y (10 * e^sin(1 * i)) Z 0
  sph Translates (moves) one or more objects in specified dimensions x, y, and z.
+ translate X (230 / 10) Y (10 * e^sin(1 * 230)) Z 0
+ color red
  sphere radius 0.8
```

The right side of the editor shows a 3D grid with a yellow curve. A red dot is placed on the curve at the position corresponding to the second 'translate' block in the code. A red line is drawn tangent to the curve at this point.

Nous avons donc une courbe et point, faisons passer une droite par ce point, avec la troisième partie du block.

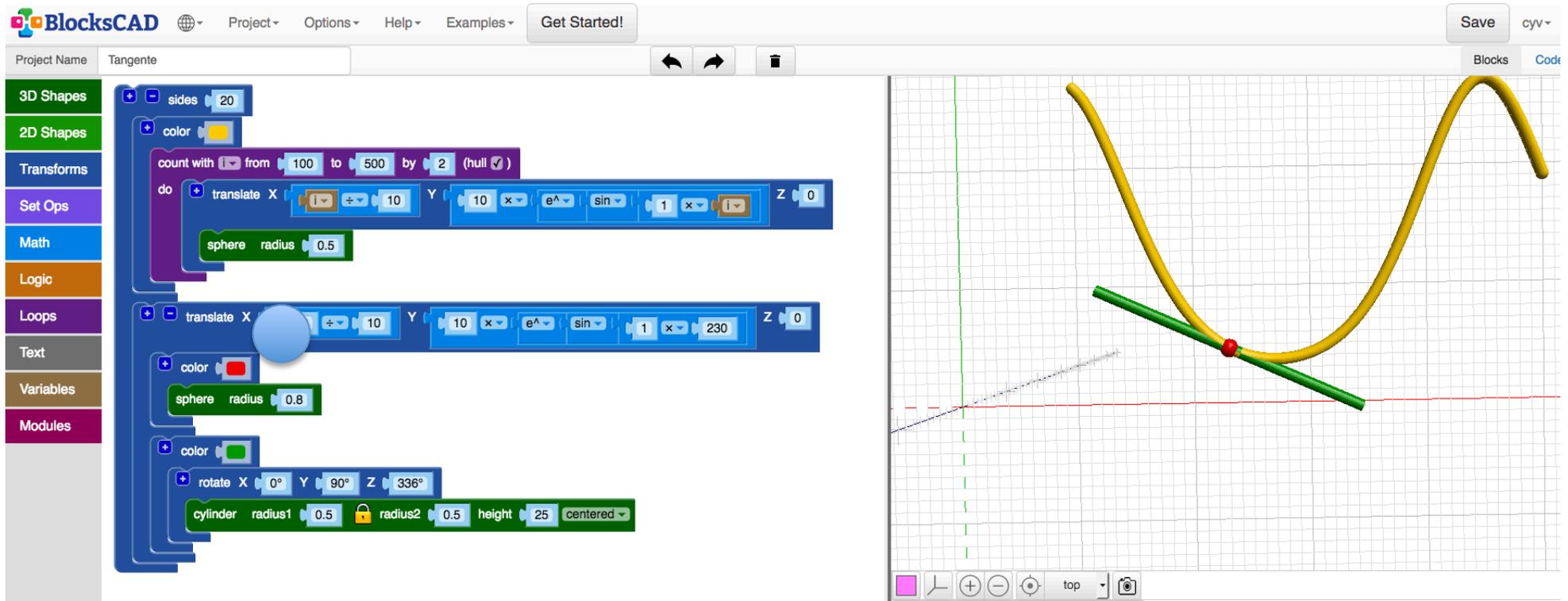
The image shows a Scratch project titled "Tangente". The code editor on the left contains the following blocks:

- 3D Shapes: sides 20
- 2D Shapes: color (yellow)
- Transforms: count with from 100 to 500 by 2 (hull checked)
- do loop:
 - translate X: $i - 10$, Y: $10 \times e^{\sin(1 \times i)}$, Z: 0
 - sphere radius 0.5
- translate X: 230 ÷ 10, Y: $10 \times e^{\sin(1 \times 230)}$, Z: 0
- color (red)
- sphere radius 0.8
- color (green)
- rotate X: 0°, Y: 90°, Z: 0°
- cylinder radius1 0.5, radius2 0.5, height 25, centered

The 3D canvas on the right shows a yellow curve on a grid. A red dot marks a point on the curve. A green line is drawn through this point, and a blue arrow points to the third part of the 'translate X' block in the code, indicating the line's position. The line is not tangent to the curve at that point.

On voit bien que l'angle entre la courbe et la droite verte (flèche bleue) n'est pas nul. Donc ce n'est pas la tangente.

Maintenant, vous pouvez faire vous même quelques essais et modifier l'angle de Z pour trouver celui qui correspond au dessin ci-dessous.



Je vous l'ai dit : c'est à vous à trouver tout seul (ou toute seule) la bonne valeur de Z. Maintenant nous avons la tangente de la courbe au point 230.

Les mathématiciens démontrent facilement qu'il n'y qu'une seule droite tangente pour ces courbes non brisées.

Allons plus loin. Faisons tourner notre programme en décalant la tangente. Simple en utilisant l'option NON CENTERED.

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', and 'Examples', along with a 'Get Started!' button. The project name is 'Tangente'. The left sidebar shows a category list: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules.

The script on the left consists of the following blocks:

- A 'sides' block set to 20.
- A 'color' block set to yellow.
- A 'count with' block from 100 to 500 by 2, with the 'hull' checkbox checked.
- A 'do' loop containing:
 - A 'translate' block with X: $i \div 10$, Y: $10 \times e^{\sin i}$, and Z: 0.
 - A 'sphere' block with radius 0.5.
- A 'translate' block with X: 230, Y: $10 \times e^{\sin 230}$, and Z: 0.
- A 'color' block set to red.
- A 'sphere' block with radius 0.8.
- A 'color' block set to green.
- A 'rotate' block with X: 0°, Y: 90°, and Z: 336°.
- A 'cylinder' block with radius1: 0.5, radius2: 0.5, height: 25, and the 'not centered' option selected.

The 3D view on the right shows a yellow sine wave on a grid. A red dot marks a point on the wave, with a green tangent line passing through it. A red horizontal line and a green vertical line are also visible on the grid. The bottom right of the 3D view includes a 'Smooth' control with options for Low, Medium (selected), and High.

Et pour finir, nous allons :

- Arrêter la courbe au point 230, en modifiant LOOP
- Supprimer le point rouge devenu inutile
- Changer la couleur de notre tangente.

Et voici à quoi sert la tangente en un point : à aller tout droit à partir d'une courbe sans que l'on voit une angle. Et voilà d'où vient l'expression « prendre la tangente ».

On peut aussi calculer directement la droite tangente en connaissant la formule de la courbe. Mais cela sera à voir avec votre professeur de Mathématiques.

Encore une tangente !

Pas tout à fait « une », mais LA tangente. La tangente est une fonction mathématique qui se calcule à partir du SINUS et du COSINUS. Noté « tg » ou « TAN », sa formule est :

$$\text{TAN (angle)} = \text{SIN (angle)} / \text{COS (angle)}$$

Une simple division pour une fonction que l'on retrouve souvent dans des formules de courbes.

https://fr.wikipedia.org/wiki/Fonction_trigonom%C3%A9trique pour plus d'explication.

Combinons les fonctions

Nous avons vu de nombreuses fonctions : polynomiales (c'est à dire avec des combinaison de X et de Y et de puissances), SINUS, COSINUS, EXPONENTIEL et plein d'autres.

Résumons les différentes façons de les associer entre elles :

- Nous pouvons faire les 4 opérations, les additionner, les diviser, les soustraire et les multiplier. $Y = \text{COS}(t) / 2^t$ par exemple
- Nous pouvons les mettre à la puissance, c'est un peu la cinquième des opérations : \wedge .
 $Y = (2^t)^{\text{COS}(t)}$ qui s'écrit aussi $Y = 2^{t^{\text{COS}(t)}}$
- Enfin, nous pouvons les combiner c'est à dire utiliser le résultat d'une fonction comme variable (on dit comme argument en mathématique) de la suivante.
Exemple $Y = \text{COS}(e^t)$.

Pour ce dernier exemple on note en mathématique $Y = \cos e(t)$ et cela se lit « rond ». Ne soyez pas étonné si vous voyez ce sigle dans différents sites ou forums sur les fonctions.

The screenshot shows the BlocksCAD software interface. The project name is "Exponentiel rond sinus". The left sidebar lists various categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace displays two loops of blocks. The top loop, "count with i from 1 to 20 by 0.5", contains a "translate" block with X=4, Y=40 * cos(e^i), and Z=0, followed by a "sphere radius 0.3" block. The bottom loop, "count with i from 1 to 30 by 0.5", contains a "translate" block with X=4, Y=20 * e^cos(i), and Z=0, followed by a "sphere radius 0.3" block. The right pane shows a 3D grid with yellow and red scribbles representing the objects being created. The bottom right of the interface has a "Render" button and a "Smooth" dropdown set to "Medium".

Comme vous pouvez le voir, « $\cos e$ » ne donne pas la même chose que « $e \cos$ », loin s'en faut. Donc, ne vous trompez pas en recopiant les formules !

Un coque de bateau avec SCALE

Nous allons réaliser une coque de bateau en utilisant un peu tout ce que nous avons déjà vu et surtout en allant plus loin avec la fonction SCALE. Reprenons notre forme de courbe de Grandpierre en remplaçant le CIRCLE par une SPHERE.

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', 'Examples', and 'Get Started!'. The project name is '3D Grandpierre Scale'. The left sidebar shows a list of categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D grid with a purple curve representing the hull's profile. The code blocks are as follows:

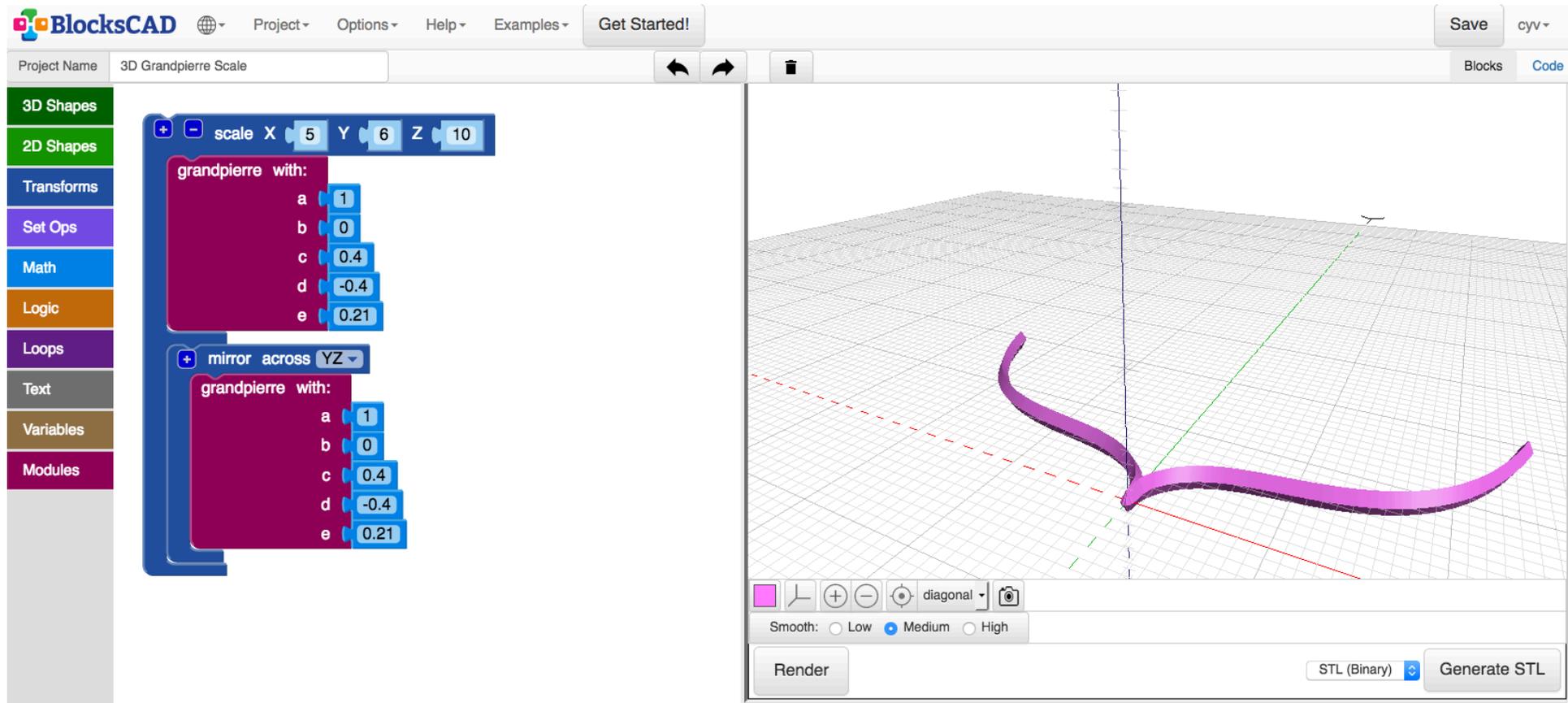
- grandpierre with:** a block with five parameters: a=1, b=0, c=0.4, d=-0.4, e=0.21.
- to grandpierre with: a, b, c, d, e** loop block.
- count with i from 0 to 3 by 0.02 (hull ✓)** loop block.
- do** loop block containing:
 - translate** block with X=0.6, Y=0, Z=0.
 - math** block: $(a \times i) + (c \times i) + (d \times i) + (e \times i^2) + b$.
 - sphere radius 0.1** block.

La formule mathématique de l'accolade

Donnons une forme symétrique avec la transformation MIRROR YZ : YZ car on doit s'imaginer que le miroir contient l'axe Y et l'axe Z.

Ensuite, utilisons SCALE pour grandir un peu notre forme qui est toute petite. SCALE permet d'étirer suivant un, deux ou les trois axes. Dans notre cas :

- 5 fois suivant X
- 6 fois suivant Y
- 10 fois suivant Z pour épaissir le bord



Au passage, remarquons que nous avons trouvé la formule mathématique de l'accolade 😊

Le Maître-Couple

Puis nous allons redresser notre couple (c'est le nom que l'on donne à cette forme pour un bateau) comme sur un chantier naval.

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', 'Examples', and 'Get Started!'. The project name is '3D Grandpierre Scale'. The left sidebar shows a category list: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules.

The main workspace shows a 3D grid with a purple boat hull model. The code blocks on the left are as follows:

- to couple** (purple block)
- translate** X: 0, Y: 0, Z: 20 (blue block)
- rotate** X: 270°, Y: 0°, Z: 0° (blue block)
- scale** X: 5, Y: 6, Z: 10 (blue block)
- grandpierre with:** (purple block) containing:
 - a: 1 (blue block)
 - b: 0 (blue block)
 - c: 0.4 (blue block)
 - d: -0.4 (blue block)
 - e: 0.21 (blue block)
- mirror across YZ** (blue block)
- grandpierre with:** (purple block) containing:
 - a: 1 (blue block)
 - b: 0 (blue block)
 - c: 0.4 (blue block)
 - d: -0.4 (blue block)
 - e: 0.21 (blue block)

A yellow callout box labeled 'couple' points to the 'to couple' block. A purple callout box labeled 'to grandpierre with: a, b, ...' points to the 'grandpierre with:' blocks. The 3D view includes a 'Render' button, a 'Smooth' dropdown (set to Medium), and a 'Generate' button.

Le liston et la quille

Rappelons que nous voulons faire une coque de bateau. Dans un premier temps, nous allons dessiner les contours, c'est à dire le liston (sur le plan XY) et la quille (dans YZ). Pour cela nous allons reprendre la formule de l'ellipse utilisée plus haut.

The screenshot displays the BlocksCAD interface for a project named "3D Grandpierre coque". The left sidebar shows a category menu with "3D Shapes" selected. The main workspace is divided into two panels: a block-based code editor on the left and a 3D view on the right.

Code Editor:

- to bordé avant:** A loop "count with i from 180 to 270 by 1" (hull) containing:
 - A "do" block with a "union" of:
 - A "color" block (orange).
 - A "translate" block with X: 26, Y: $100 \times \sin(i)$, Z: 0.
 - A "sphere" block with radius 0.5.
 - A "plus" block with:
 - A "color" block (yellow).
 - A "translate" block with X: 0, Y: $100 \times \sin(i)$, Z: $20 \times \cos(i) + 180$.
 - A "sphere" block with radius 0.5.
- to bordé arrière:** A loop "count with i from 90 to 180 by 1" (hull) containing:
 - A "do" block with a "union" of:
 - A "color" block (blue).
 - A "translate" block with X: 26, Y: $70 \times \sin(i)$, Z: 0.
 - A "sphere" block with radius 0.5.
 - A "plus" block with:
 - A "color" block (cyan).
 - A "translate" block with X: 0, Y: $70 \times \sin(i)$, Z: $20 \times \cos(i) + 180$.
 - A "sphere" block with radius 0.5.

3D View: Shows a 3D coordinate system with a grid. The hull is rendered with colored spheres. The front edge (yellow/orange) and back edge (blue) are visible. The keel (purple) is also shown. The view includes a "Render" button, "Smooth" options (Low, Medium, High), and a "Generate STL" button.

Block Legend:

- couple - count with i from 180...
- couple - count with i from 91...

En jaune et orange, l'avant du bateau et en bleu l'arrière. A noter, la formule pour Y est la même dans chacun des blocks.

Les couples

The screenshot displays the BlocksCAD interface for a project named "3D Grandpierre coque". The left sidebar lists various block categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains a 3D grid with a rendered model of a curved structure, likely a Grandpierre coque, with several colored lines (yellow, purple, pink) representing different parts or edges. The code blocks are organized as follows:

- to bordée avant** (Loop block):
 - count with** from 180 to 270 by 1 (hull)
 - do** block:
 - union** block:
 - color** (orange)
 - translate** X: 26 * cos(i), Y: 100 * sin(i), Z: 0
 - sphere** radius 0.5
 - plus** block:
 - color** (yellow)
 - translate** X: 0, Y: 100 * sin(i), Z: 20 * cos(i) + 180
 - sphere** radius 0.5

- to grandpierre with: a, b, ...** (Loop block):
- couple** block
- to couple translate X 0 Y ...** (Loop block):
 - translate** X: 0, Y: 100 * sin(i), Z: 0
 - scale** X: 1 * cos(i), Y: 1, Z: 1 * cos(i) + 180
 - couple** block
- to bordée arrière count wi...** (Loop block):
- count with** from 200 to 270 by 20 (hull)
- do** block:
 - translate** X: 0, Y: 100 * sin(i), Z: 0
 - scale** X: 1 * cos(i), Y: 1, Z: 1 * cos(i) + 180
 - couple** block

At the bottom right, there is a rendering control panel with a "Render" button, a "Smooth" dropdown set to "Medium", and a "Gener" button.

Regardez bien le nouveau block de la page précédente, en comparaison de celui de la bordé. Nous réalisons :

- un couple (module COUPLE)
- puis nous le mettons à l'échelle suivant X et suivant Z avec les mêmes formules de X et Z des bordées, mais sans les paramètres « 26 » ni « 20 » devant
- puis nous translatons le tout sur l'axe Y qui est l'axe de la coque, en utilisant la formule donnée pour Y dans le module « bordée avant ». C'est pour cela qu'il était important que le module « bordée avant » ait la même formule pour Y.

La boucle permet de faire plusieurs couples. Nous commençons à 200 pour ce dessin car nous avons déjà le maître-couple en 180.

Le dessin suivant reprend le même concept mais pour les couples de l'arrière du bateau.

BlocksCAD Project: 3D Grandpierre coque

3D Shapes
2D Shapes
Transforms
Set Ops
Math
Logic
Loops
Text
Variables
Modules

to bordée arrière

count with i from 90 to 180 by 1 (hull)

color

couple

to grandpierre with: a, b, ...

to couple translate X 0 Y ...

to bordée avant count with...

union

color

translate X 26 x cos i Y 70 x sin i Z 0

sphere radius 0.5

bordée avant

bordée arrière

plus

color

translate X 0 Y 70 x sin i Z 20 x cos i + 180

sphere radius 0.5

couple - count with i from 200...

count with i from 110 to 180 by 20 (hull)

do

translate X 0 Y 70 x sin i Z 0

scale X 1 x cos i Y 1 Z 1 x cos i + 180

couple

Render

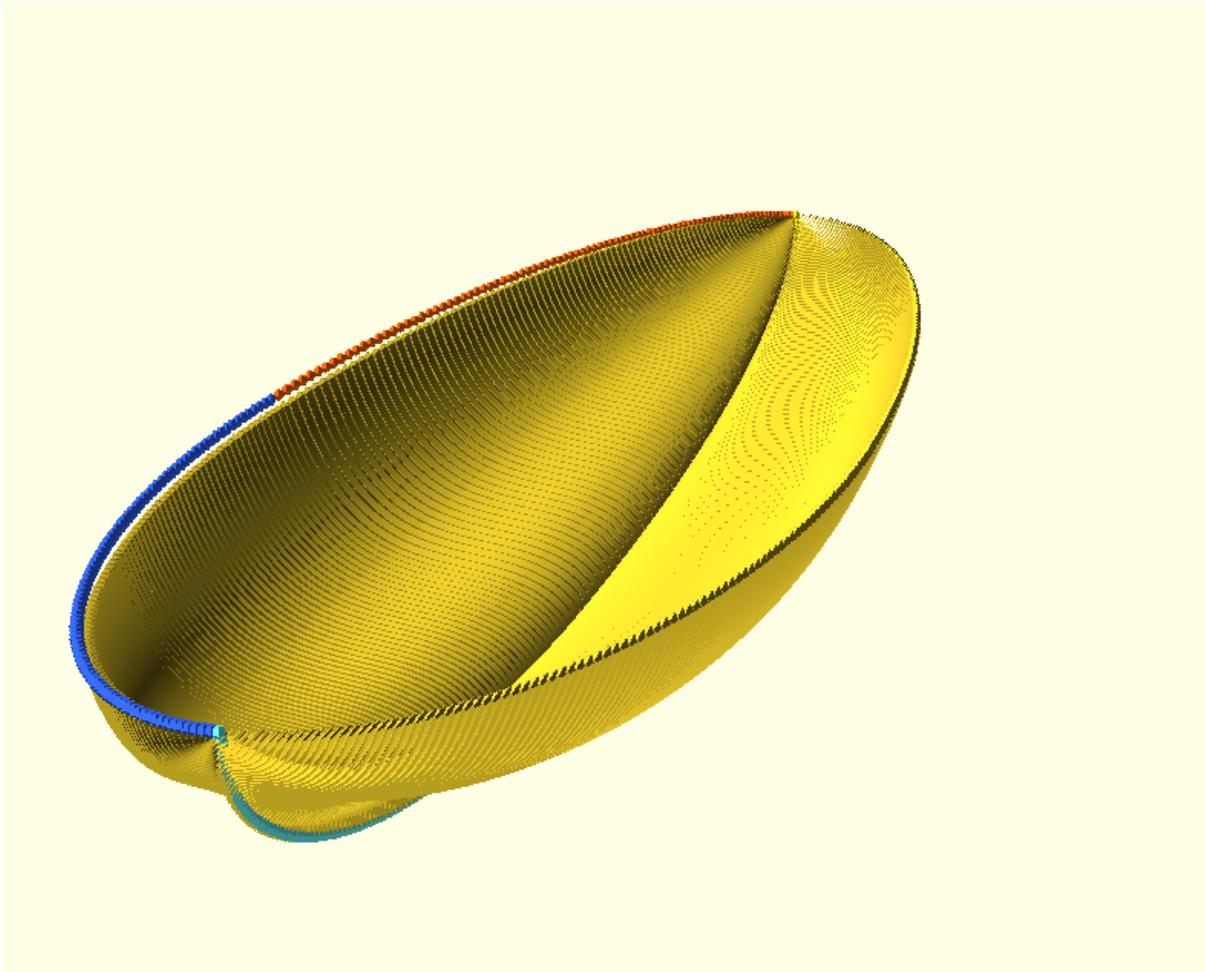
Smooth: Low Medium High

STL (Binary) Gene

Nous avons indiqué par des flèches :

- flèches bleues, la formule en Y qui est reprise pour le TRANSLATE
- flèche verte, qui reprend la formule Z
- flèche rouge, qui reprend la formule pour X

La coque complète



Il suffit maintenant de mettre les couples les uns à côté des autres, en mettant « 1 » dans le BY de la boucle (LOOP) et nous aurons une coque, peut-être un jour.

Dis Madame BlocksCAD, quand est-ce que tu nous fais un programme qui va aussi vite qu'OpenSCAD ?

L'image ci-contre est la version OpenSCAD (15 secondes), car le RENDER sous BlocksCAD tourne encore.

Et voici comment nos amis SINUS et COSINUS ainsi que SCALE nous ont permis de faire une coque de bateau, au moins acceptable pour une maquette.

Le sapin de Noël

Un joli sapin de Noël proposé par Xavière pour revoir quelques fonctions avant d'aborder la troisième dimension.

The screenshot displays the BlocksCAD software interface. On the left, a sidebar lists various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace is filled with a complex assembly of blocks. A central vertical 'union' block contains several 'translate' blocks, each followed by a 'torus' block with varying radii and heights. To the right, there are 'do' loops for '3D Spirale Archimede' and '3D Spirale boules', which use '3D Point cylindrique' and '3D Point boule' blocks respectively, with parameters for radius, azimuth, and height. The right side of the interface shows a 3D rendering of a green Christmas tree with yellow garlands and white ornaments. Below the 3D view, there are controls for 'Smooth' (Low, Medium, High) and a 'Render' button. At the bottom right, there is a 'Generate STL' button with a dropdown menu set to 'STL (Binary)'. The top of the window shows the 'Project Name' as 'sapin noel' and a 'Save' button.

Voici le sapin avec sa verdure, une guirlande et des boules de Noël.

Un feuillage de tores

Le feuillage se compose de tores empilés et d'un tronc cylindrique.

The screenshot displays the BlocksCAD software interface. On the left, a sidebar lists various categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D model of a Christmas tree, which is a stack of green torus shapes (forming the foliage) and a brown cylinder (forming the trunk). The tree is positioned on a grid. Below the 3D view, there are controls for rendering, including a 'Render' button, a 'Smooth' dropdown menu (set to Medium), and a 'Generate' button. The top of the interface shows the 'Project Name' as 'sapin Noel' and a 'Save' button. The code on the left is a sequence of blocks: a 'translate' block (X: 0, Y: 0, Z: 20), a 'union' block containing several 'torus' and 'cylinder' blocks with various parameters (radius, height, not centered), and a 'plus' block containing a 'color' block (brown) and a 'translate' block (X: 0, Y: 0, Z: -15). The 'torus' blocks have parameters for radius1, radius2, sides, and faces. The 'cylinder' blocks have parameters for radius1, radius2, height, and a 'not centered' dropdown menu.

La spirale d'Archimède pour guirlande

La guirlande est constituée d'une spirale d'Archimède que nous avons déjà vue avec quelques modifications. En effet notre précédente guirlande était plate. Il faut donc que nous fassions varier la hauteur en même temps que la spirale tourne.

The screenshot displays the BlocksCAD environment. On the left, a sidebar lists categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains a 3D view of a pink Archimedean spiral on a grid. The code blocks are as follows:

- translate** X: 0, Y: 0, Z: 20
- union** block containing:
 - scale** X: 1.3, Y: 1.3, Z: 1, color: #630
 - plus** block containing:
 - color** #630, translate X: 0, Y: 0, Z: 1
 - plus** block containing:
 - scale** X: 0.5, Y: 0.5, Z: 1
 - color** #630
 - 3D Spirale Archimede** with: pas: 1, tour: 3, hauteur: 30
- to 3D Spirale Archimede** with: pas, tour, hauteur
 - count with i from 0 to tour x 360 by 1 (hull checked)
 - do
 - 3D Point cylindrique** with: rayon: pas x i, azimuth: i, z: hauteur x i
- to 3D Point cylindrique** with: rayon, azimuth, z
 - translate** X: rayon x cos azimuth, Y: rayon x sin azimuth, Z: z
 - rotate** X: 0°, Y: 0°, Z: -1 x rayon
 - sphere** radius: 1

Nous avons repris les deux modules de la spirale, changé le nom en rajoutant 3D, et rajouté la variable HAUTEUR.

Regardez la formule de Z dans le block 3D SPIRALE... la variable (i) varie de 0 à « TOUR * 360 ». Quand (i) vaut 0 (c'est à dire au départ), Z doit être au plus haut, c'est dire valoir HAUTEUR.

Quand (i) vaut « TOUR * 360 » alors HAUTEUR doit valoir 0.



La formule est $Z = \text{HAUTEUR} * (1 - (i / (\text{TOUR} * 360)))$.

On vérifie bien que :

- si (i) = 0, $Z = \text{HAUTEUR} * 1$
- si (i) = TOUR*360, alors $[i / (\text{TOUR} * 360)]$ vaut 1 et donc $Z = \text{HAUTEUR} * (1 - 1) = 0$

Les Boules de Noël réparties sur la guirlande

Maintenant, passons aux boules placées sur la guirlande.

Nous avons dupliqué les deux blocs qui permettent de dessiner la guirlande :

- ❑ le block 3D POINT BOULE est modifier pour dessiner une boule avec une sphère un peu plus grande que dans le block initial. Vous pouvez remarquer une fonction SCALE à l'intérieur de ce block pour supprimer l'effet du SCALE dans le block principal
- ❑ le block 3D SPIRALE BOULE est quant à lui à peine modifié. On supprime l'option HULL pour ne pas faire une courbe mais uniquement des points, on commence vers 280 (et non pas à zéro) et surtout on fait des sauts de 57 (ou de n'importe quel nombre assez grand) pour disséminer les boules de Noël.

En reprenant une copie des blocks de la guirlande, nous sommes sûrs que les boules seront bien positionnées sur la guirlande. Dans le block principal, nous avons utilisé la transformation SCALE pour faire coller la guirlande avec la sapin.

Il ne vous reste plus qu'à fabriquer les cadeaux.

Les courbes en trois dimensions

Passons dans la troisième dimension : les coordonnées cartésiennes

Pour ceux qui veulent voir la quatrième dimension sans passer par la troisième, je leur conseille le lien Hipparque en fin de document et si vous ne comprenez pas tout, alors bienvenue au club, nous nous sentirons moins seul.

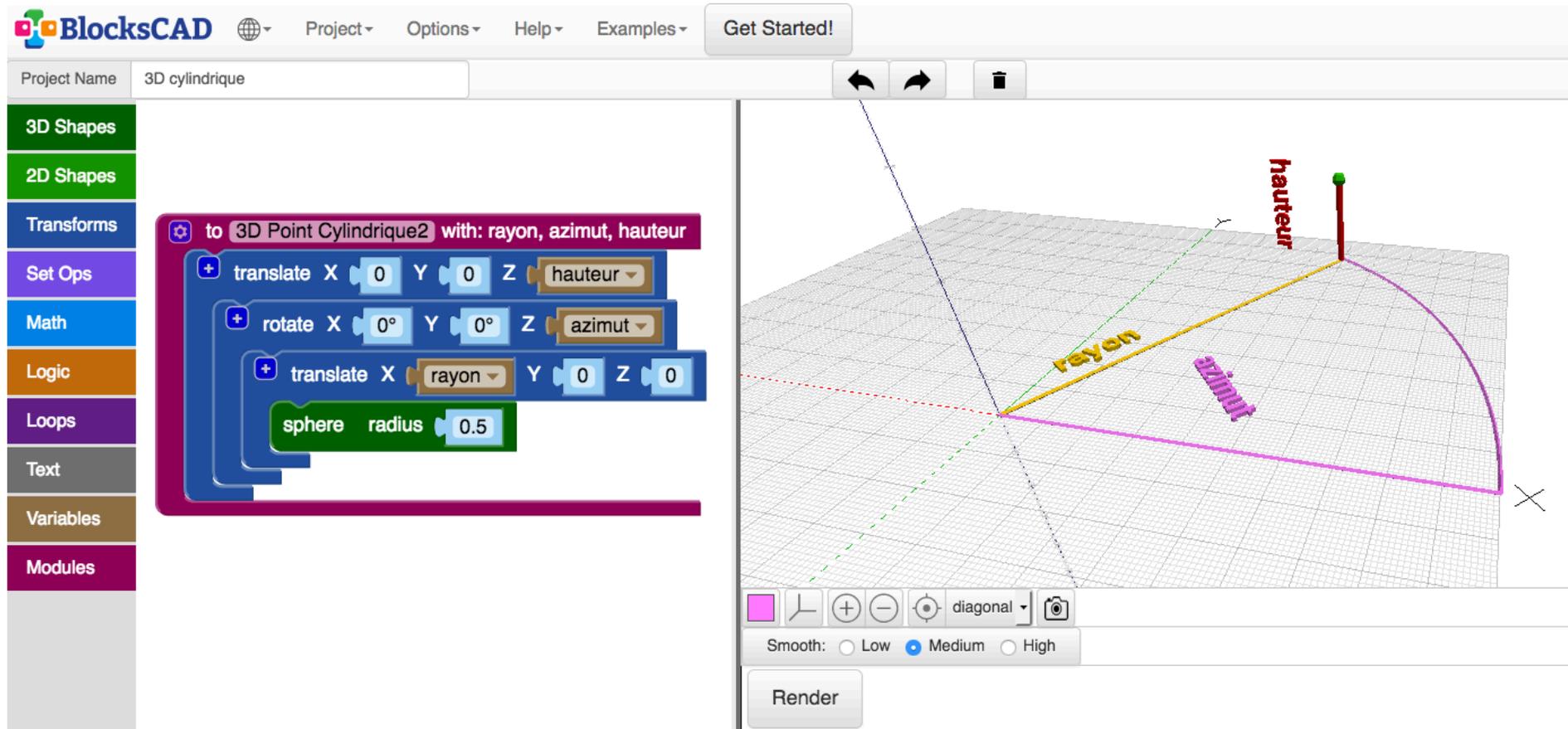
Nous avons déjà vu des formes en volumes et pas uniquement des profils en 2D, c'est à dire en surface sur un plan. Pour les formes en 2D, seuls les abscisses (X) et les ordonnées (Y) sont utilisées. Pour passer nos formules en 3D, il va falloir jouer avec la hauteur (Z).

The screenshot shows the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', 'Examples', and 'Get Started!'. The project name is '3D Point'. The left sidebar contains categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, and Variables. The main workspace displays a 3D grid with a red line and a yellow cube. The block-based programming environment shows several blocks: a 'translate X 10 Y 30 Z 20' block, a 'sphere radius 2' block, a 'translate X 10 Y 30 Z 0' block, a 'color' block, and a 'cube X 1 Y 1 Z 20 not centered' block. A tooltip for the 'translate' block reads: 'Translates (moves) one or more objects in specified dimensions x, y, and z.' The bottom right corner of the workspace shows a 'Smooth' control with options for Low, Medium (selected), and High.

Sur le schéma précédent, on voit bien en bleu, l'abscisse X qui vaut 10, ensuite, l'ordonnée Y qui vaut 30 et enfin la hauteur Z qui vaut 20. Ainsi, tout point dans l'espace est connu par trois coordonnées (et aussi une origine et trois axes).

Les coordonnées cylindriques

Comme nous l'avons déjà vu, le choix des coordonnées est important. Les coordonnées cylindriques sont un mélange des coordonnées polaires (azimut, rayon) et de coordonnée cartésienne (Axe Z) appelé Hauteur. Les coordonnées cylindriques sont très pratiques pour les objets de type cylindriques, comme par exemple des tubes, des pièces d'échecs, des trucs qui y ressemblent



Nous décomposons le block précédent : une petite sphère, un TRANSLATE du rayon pour courir le long de l'axe violet et arriver près du X, un ROTATE de l'azimut pour se placer au bout de l'axe jaune et un TRANSLATE suivant Z pour arriver à destination. Encore plus simplement, il est possible de faire le TRANSLATE en Z directement dans le premier TRANSLATE. Essayez donc !

Des Coordonnées cylindriques aux coordonnées cartésiennes

Quelles sont les formules qui permettent de passer des coordonnées cylindriques aux coordonnées cartésiennes ?

Pour un Point (RAYON, AZIMUT, HAUTEUR) nous connaissons déjà les formules pour X et Y. Celle pour Z est évidente :

- $X = \text{RAYON} * \text{COSINUS} (\text{AZIMUT})$
- $Y = \text{RAYON} * \text{SINUS} (\text{AZIMUT})$
- $Z = \text{HAUTEUR}$

The image shows two Scratch code blocks side-by-side, both titled "to 3D Point Cylindrique" with parameters "rayon", "azimut", and "hauteur".

The top block, titled "3D Point Cylindrique2", contains the following nested blocks:

- translate X: 0, Y: 0, Z: hauteur
- rotate X: 0°, Y: 0°, Z: azimut
- translate X: rayon, Y: 0, Z: 0
- sphere radius: 0.5

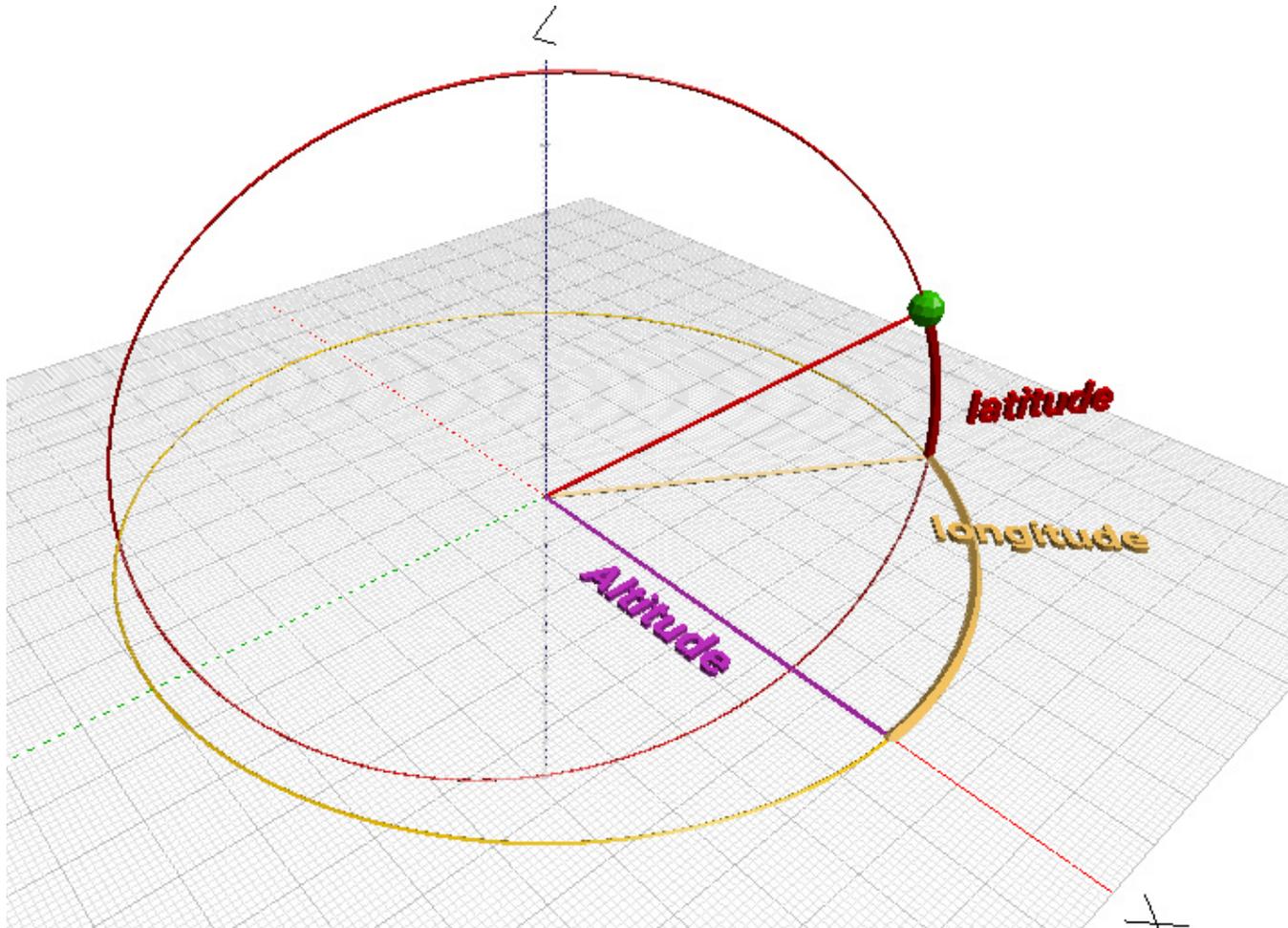
The bottom block, titled "3D Point Cylindrique", contains the following nested blocks:

- translate X: rayon * cos azimut, Y: rayon * sin azimut, Z: hauteur
- sphere radius: 0.5

Ces deux blocks sont totalement équivalents.

Les coordonnées sphériques

Les coordonnées sphériques sont vraiment l'extension des coordonnées polaires en 3 dimensions. Elles sont très utilisées pour les coordonnées terrestres et l'astronomie. Si vous n'avez pas encore visionné la vidéo sur Hipparque, alors stoppez votre lecture et allez-y : c'est le moment, trop génial!!



Maintenant que vous avez visionné la vidéo, nous allons pouvoir aller plus vite.

En jaune fin, nous avons tracé l'équateur de notre terre. Le trait en violet est le rayon de la terre, que nous allons appeler l'altitude. Et le point vert c'est nous. Yes !

Revoyons un peu nos définitions :

- ❑ L'altitude, c'est la distance du segment violet
- ❑ Longitude, c'est l'angle horizontal entre le rayon violet et le rayon jaune
- ❑ Latitude, c'est l'angle vertical entre le rayon jaune et le rayon rouge.

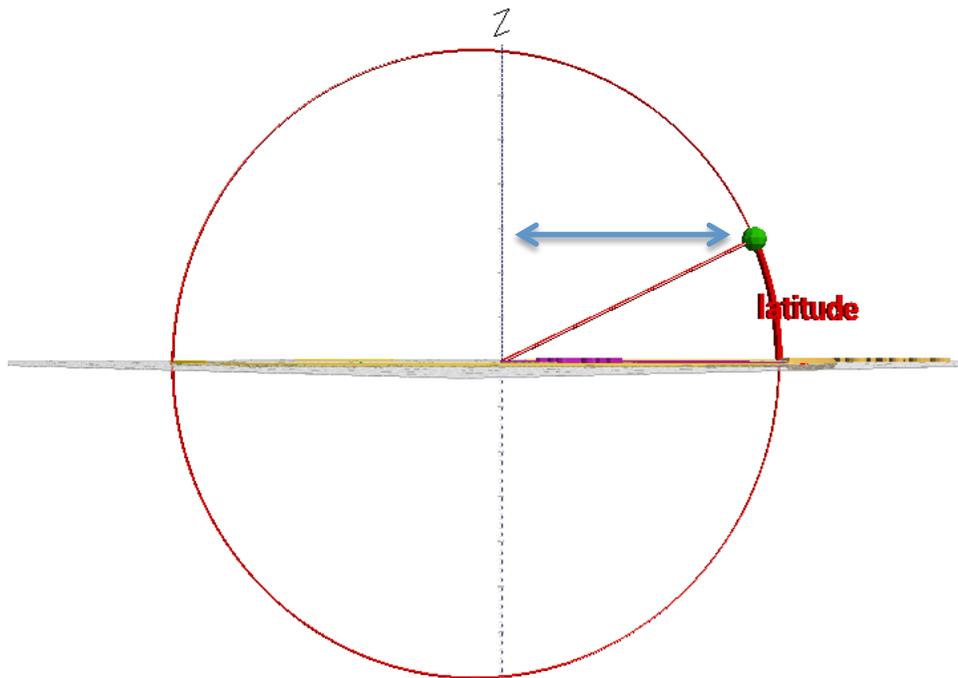
Remarque : sur notre bonne vieille Terre la notion d'altitude n'est pas prise au niveau du centre de la Terre mais de la surface.

Z en coordonnées sphériques

Je n'ai pas trouvé de TRANSFORMS qui permettent de tracer ce dessin facilement. On pourrait toujours faire un TRANSLATE suivant l'axe violet, un ROTATE suivant Z poursuivre l'arc jaune, mais le dernier ROTATE est plus dur à connaître car on ne sait pas sur quel axe nous devons le faire.

Maintenant, nous allons trouver comment on peut exprimer (X,Y,Z) en fonction du (Altitude, Longitude, Latitude).

Commençons par Z.



Le dessin ci-dessus est exactement le même que précédent, sauf que nous le regardons dans le plan du méridien rouge.

Reconnaissez vous la figure : c'est la même que celle que nous avons utilisée pour donner la définition du SINUS et du COSINUS.

$$Z = (\text{altitude}) * \text{SINUS} (\text{latitude}).$$

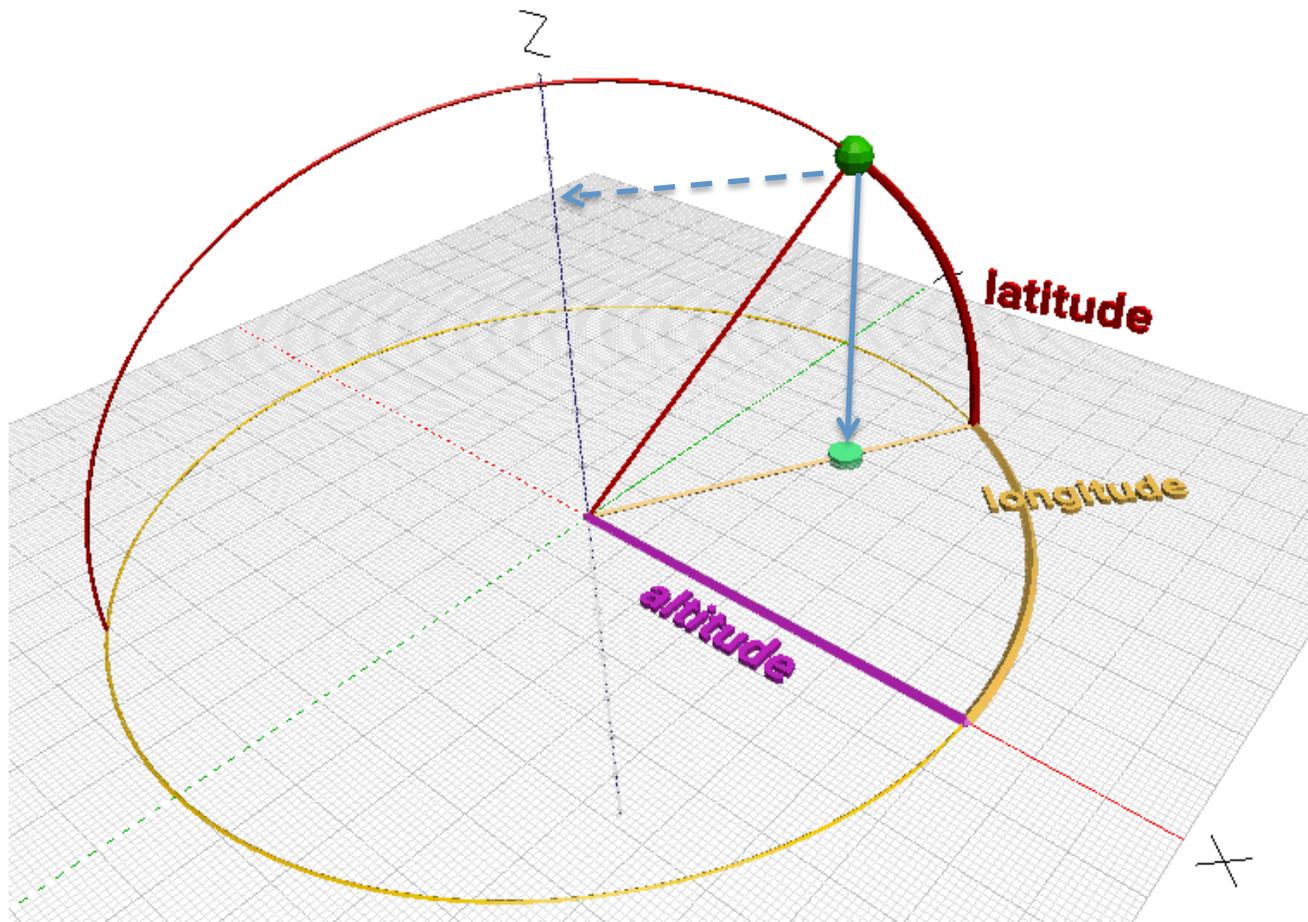
La latitude est bien un angle et le SINUS la valeur sur l'axe Z.

Trop facile !

Le cosinus de la latitude

Maintenant que nous avons trouvé le sinus de la latitude, nous pouvons nous poser la question du cosinus de la latitude. Hélas il ne passe pas ni par l'axe X ni par l'axe Y.

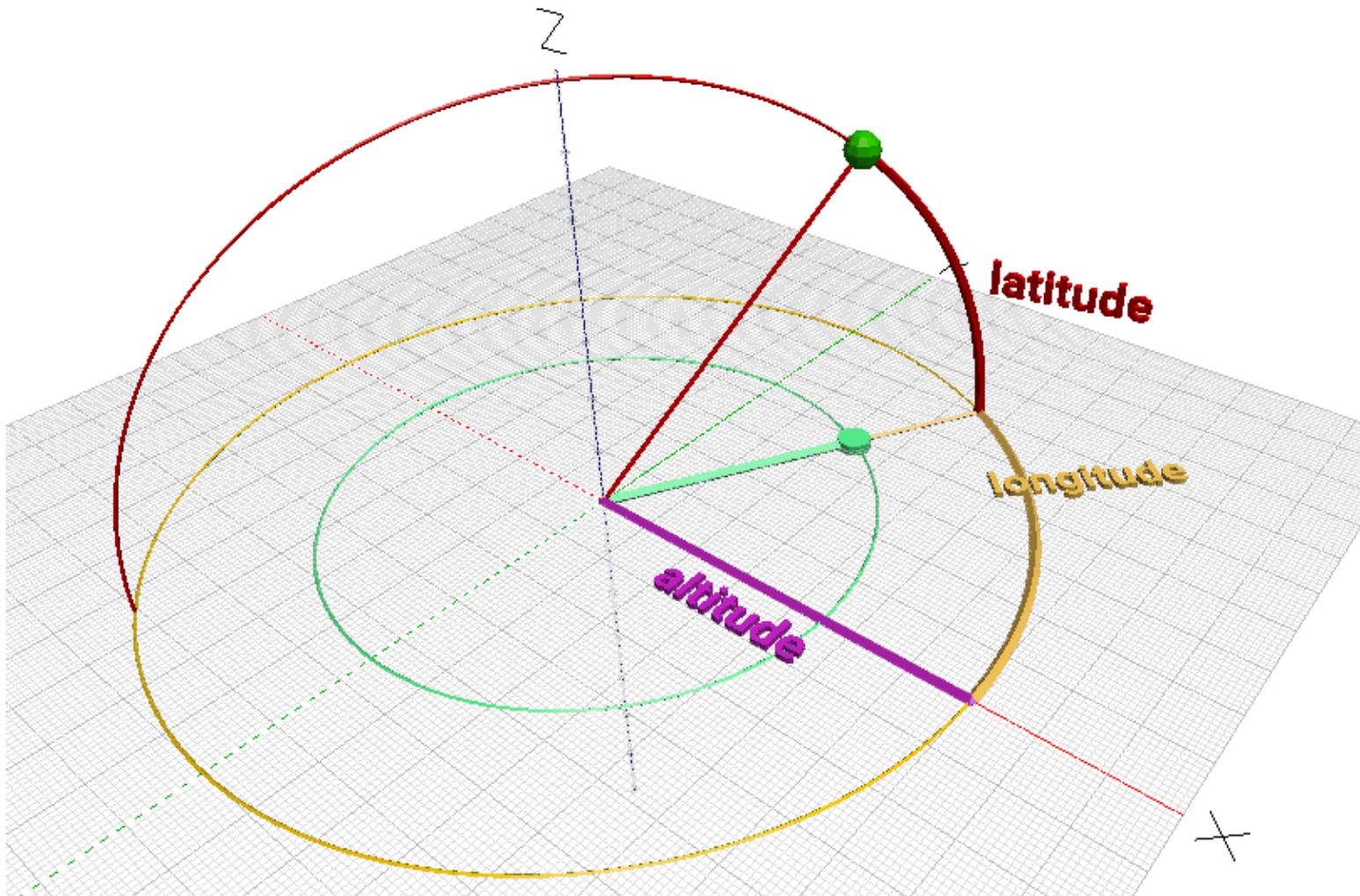
Regardez la figure ci-dessous.



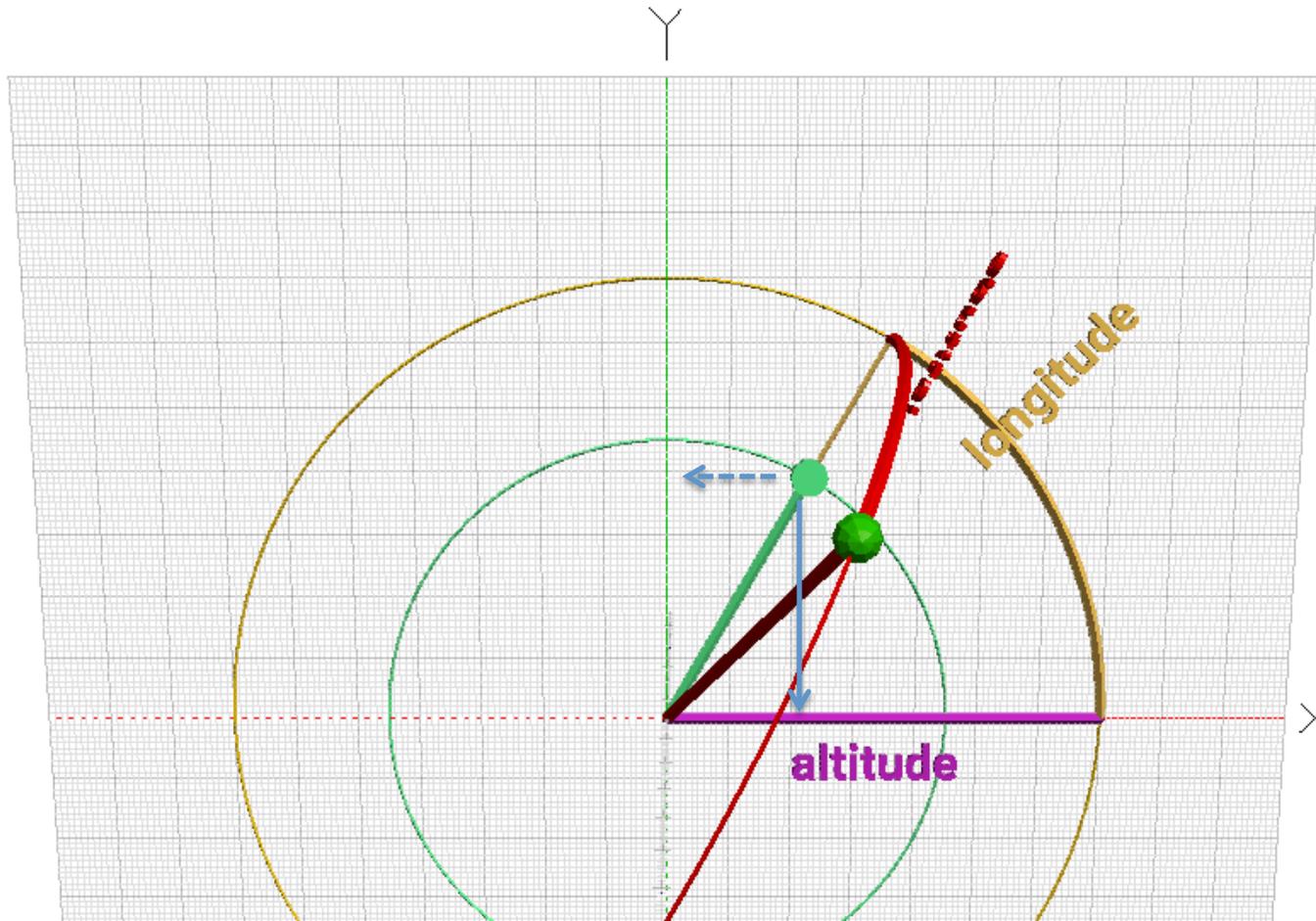
En pointillés, nous avons indiqué le SINUS. Le COSINUS est la distance entre l'origine et la pastille verte.

Distance à la Pastille verte =
 $(\text{altitude}) * \text{COSINUS}(\text{latitude})$

Maintenant, nous allons tracer un cercle qui passe par cette pastille verte.



et voilà, un joli cercle qui passe par la pastille verte. Et nous savons que la distance du segment vert est « $\text{altitude} \cdot \cos(\text{latitude})$ ». Ce segment vert est le rayon du cercle vert.
Faisons tourner notre objet pour voir le plan X-Y qui est aussi le plan de l'équateur.



Maintenant, on commence à voir apparaître la fin de la formule que nous cherchons. L'angle du segment vert est la longitude :

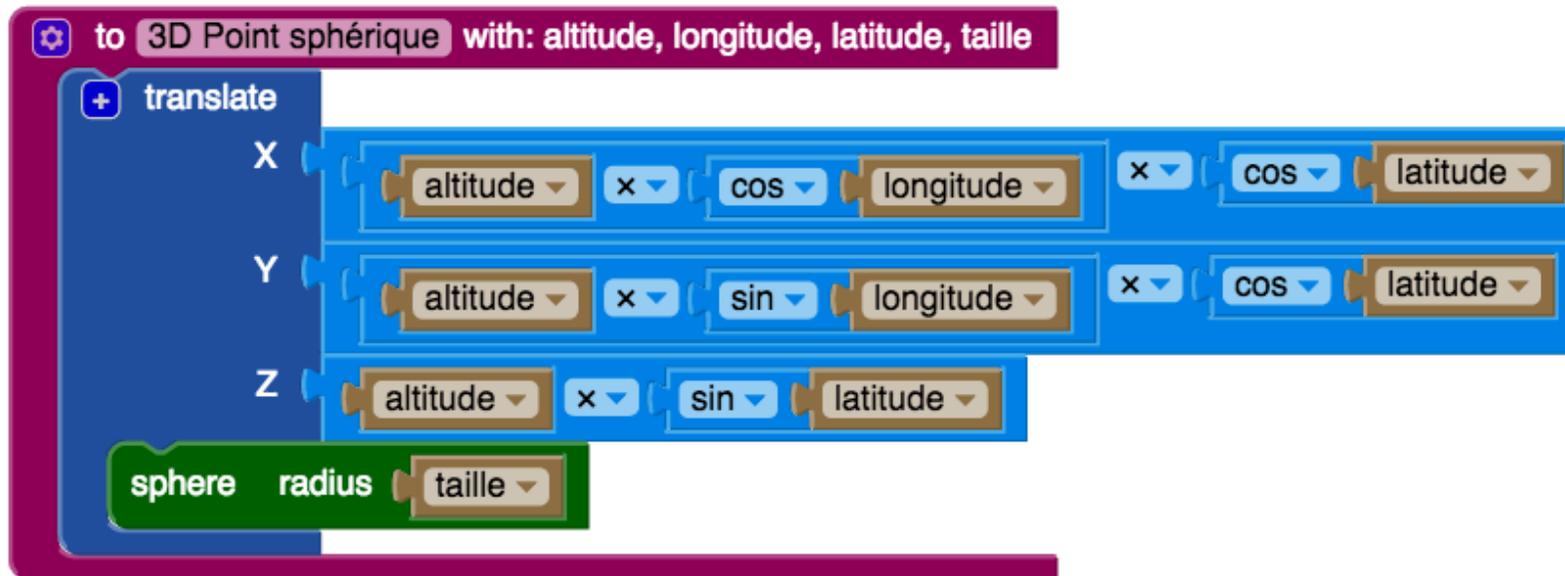
- ❑ $X = (\text{distance du segment vert}) \cdot \text{COSINUS} (\text{longitude})$ (flèche bleue pleine)
- ❑ $Y = (\text{distance du segment vert}) \cdot \text{SINUS} (\text{longitude})$ (flèche bleue pointillée).

Un block pour un point en coordonnées sphériques

Nous pouvons trouver maintenant les formules complètes en remplaçant la distance du segment vert par sa formule avec la latitude :

- $X = (\text{altitude}) * \text{COSINUS} (\text{latitude}) * \text{COSINUS} (\text{longitude})$
- $Y = (\text{altitude}) * \text{COSINUS} (\text{latitude}) * \text{SINUS} (\text{longitude})$
- $Z = (\text{altitude}) * \text{SINUS} (\text{latitude})$

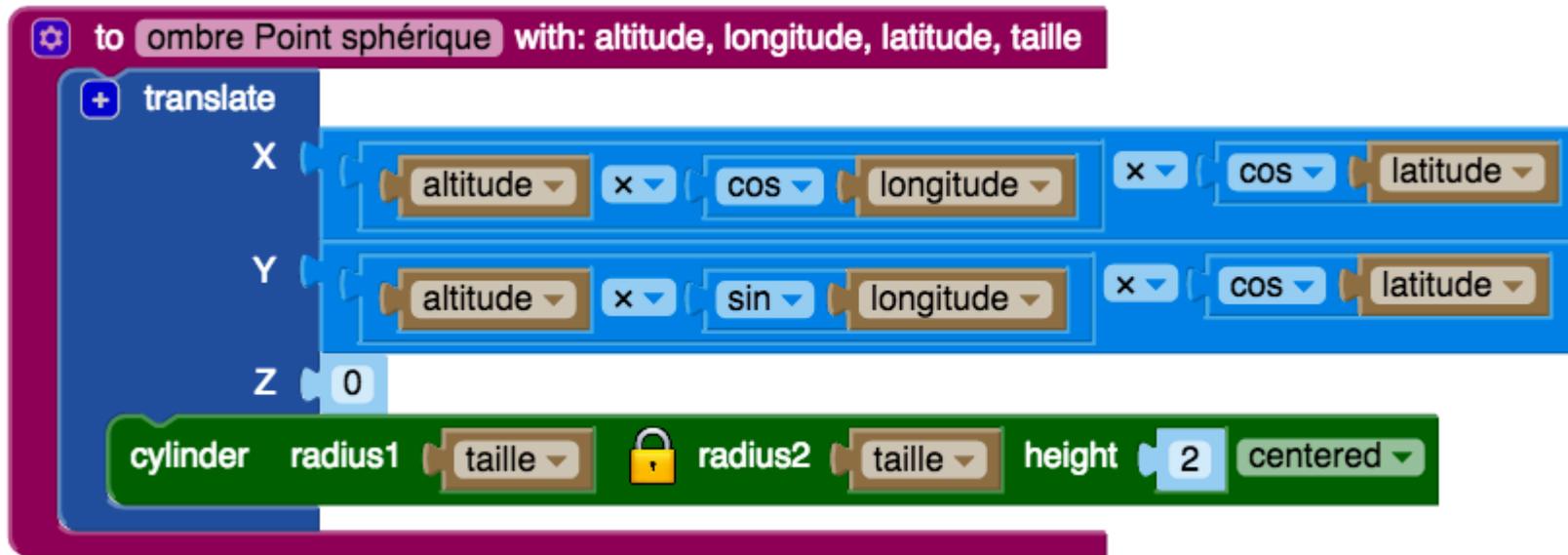
Ce qui donne en BlocksCAD.



Un module BlocksCAD avec les paramètres altitude, longitude et latitude et aussi un paramètre taille pour facilement changer le rayon du point à dessiner.

L'ombre du point en coordonnées sphérique

Un petit block à garder en mémoire qui nous a servi précédemment. Ce block permet de représenter l'ombre du point en coordonnée sphérique sur le plan X-Y ce qui peut aider.



C'était la fonction qui donnait la pastille verte.

Les méridiens en coordonnées sphérique

Un méridien est un cercle sur une sphère qui a toujours la même longitude. Pour ne réaliser un, il suffit donc de faire une boucle qui fait varier la latitude pour une longitude constante.

The screenshot displays the BlocksCAD software interface. The project name is "fonctions sphériques". The workspace contains several code blocks:

- A **color** block.
- A **count with** loop: from 0 to 360 by 45 (hull).
 - do** loop: **méridien with:** Altitude: 70, longitude: i, taille: 0.3.
- A **to** loop: **to méridien with: Altitude, longitude, taille**.
 - count with** loop: from 0 to 360 by 1 (hull).
 - do** loop: **3D Point sphérique with:** altitude, longitude, latitude, taille.
- A **to 3D Point sphérique with: altitude, longitude, latitude, taille** loop.
 - translate** block with X, Y, and Z coordinates:
 - X: altitude * cos(longitude) * cos(latitude)
 - Y: altitude * sin(longitude) * cos(latitude)
 - Z: altitude * sin(latitude)
 - sphere** block with radius: taille.

The 3D view shows a sphere with several colored meridian lines (yellow, purple, green) and a grid. The bottom right corner has a **Render** button and a **STL (Binary)** export option.

Le module MERIDIEN est donc une simple boucle sur la latitude. Dans le dessin, un méridien en jaune et une boucle pour faire un méridien tous les 45°. Il ne faut pas cependant en abuser car le temps de modélisation est long.

Les parallèles en coordonnées sphériques

Un parallèle est un cercle dont la latitude est constante. Le module PARALLELE réalise une boucle de 360 points en faisant donc varier uniquement la longitude et en gardant la latitude constante.

The screenshot displays the BlocksCAD environment. On the left, a sidebar lists various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains a script with the following logic:

- A loop **count with k** from 0 to 90 by 25 (hull) containing:
 - A **color** block.
 - A **parallèle with:** block with parameters: altitude (70), latitude (k), and taille (1).
 - A nested loop **to parallèle with: altitude, latitude, taille** containing:
 - A **count with i** from 0 to 360 by 1 (hull) loop.
 - A **do** block containing a **3D Point sphérique with:** block with parameters: altitude, longitude (i), latitude, and taille.
- A **to 3D Point sphérique with: altitude, longitude, latitude, taille** block containing:
 - A **translate** block with X, Y, and Z coordinates:
 - X: altitude × cos(longitude) × cos(latitude)
 - Y: altitude × sin(longitude) × cos(latitude)
 - Z: altitude × sin(latitude)
 - A **sphere** block with radius (taille).

On the right, a 3D view shows a grid with X, Y, and Z axes. Several yellow circular parallel lines are drawn on the surface of a sphere, representing the output of the script. The interface includes a **Render** button and a **Smooth** control set to Medium.

Nous avons réalisé plusieurs parallèles pour voir la facilité, une fois que le module est créé. Sur ce dessin, seuls les parallèles de l'hémisphère Nord.

Une droite en coordonnées sphériques

Pour finir nos fonctions en coordonnées sphériques, passons à une simple droite.

Le module de droite en coordonnées sphérique fait varier l'altitude pour une longitude et une latitude données.

The screenshot displays the BlocksCAD interface with a 3D model of a sphere. The sphere is rendered with a grid and features several yellow latitude lines and a purple meridian line. A green point is visible on the sphere's surface. The interface includes a menu bar with options like Project, Options, Help, and Examples. The sidebar on the left lists various categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains several code blocks: a 'to 3D droite sphérique' block with a 'count with i' loop, a '3D Point sphérique' block with parameters for altitude, longitude, latitude, and taille, and a '3D droite sphérique' block with parameters for longitude (60°), latitude (50°), début altitude (0), fin altitude (70), and épaisseur (0.5). The 3D view shows a sphere with a grid, a purple line representing a meridian, and yellow lines representing latitude. A green point is visible on the sphere's surface. The bottom right of the interface has a 'Render' button and a 'STL (Binary)' dropdown.

Sur les parallèles du schéma précédent, un point en vert et la droite en violet.

Maintenant, nous avons toutes les fonctions travailler avec ces coordonnées sphériques qui ne sont pas si difficiles que cela.

Les fonctions sphériques

Pour terminer, le block « fonctions Sphériques » reprend tous les modules. Il suffit de faire « import blocks into current project » ou l'équivalent français pour pouvoir en disposer.

The screenshot displays the BlocksCAD software interface. The top menu bar includes "Project", "Options", "Help", and "Examples". The project name is "fonctions sphériques". The left sidebar shows category tabs: "3D Shapes", "2D Shapes", "Transforms", "Set Ops", "Math", "Logic", "Loops", "Text", "Variables", and "Modules". The central workspace contains a script editor with the following blocks:

- A "color" block.
- A "3D Point sphérique with:" block with parameters: altitude: 70, longitude: 60°, latitude: 50, taille: 3.
- A "3D droite sphérique with:" block with parameters: longitude: 60°, latitude: 50°, début altitude: 0, fin altitude: 70, épaisseur: 0.5.
- A "count with k from 0 to 90 by 25 (hull)" block.
- A "do" loop containing a "color" block with a tooltip "Applies the color to the child object, which must be 3D." and a "3D droite sphérique with:" block with parameters: altitude: 70, latitude: k, taille: 1.
- A "color" block.
- A "count with i from 0 to 360 by 45 (hull)" block.
- A "do" loop containing a "méri dien with:" block with parameters: Altitude: 70, longitude: i, taille: 0.3.

The right sidebar shows a 3D view of a sphere with latitude and longitude lines. The view is labeled "front" and has a "Render" button. The "Smooth" setting is set to "Medium". The "STL (Binary)" format is selected.

Le passage des courbes 2D en 3D

Une tresse sinusoïdale

Pour passer assez facilement du 2D au 3D, il est nécessaire de changer la fonction CIRCLE en fonction SPHERE. Regardez le petit exemple ci-dessous.

The screenshot displays the BlocksCAD software interface. The top navigation bar includes the logo, a globe icon, and menu items: Project, Options, Help, Examples, and Get Started!. On the right, there are buttons for Save and cyv, and tabs for Blocks and Code. The Project Name field contains 'tresse sinusoïdale'. The left sidebar lists various categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D view of a sine wave braid with red, yellow, and green links. Below the 3D view is a toolbar with icons for color, zoom, and camera, along with a 'diagonal' dropdown and a 'Smooth' selector (Low, Medium, High). At the bottom, there are buttons for 'Render', 'STL (Binary)', and 'Generate STL'. The code editor on the left contains the following blocks:

```
count with i from -1 to 1 by 1 (hull )
do
  color HSV hue 120 saturation 100 value 100
  sinus2 with:
    periode 20
    amplitude 5
    phase 120°
    sens 0.4
to sinus2 with: periode, amplitude, phase, sens
count with i from -30 to 30 by 0.3 (hull )
do
  translate
  X 1
  Y amplitude x sin 360 periode x phase
  Z sens x amplitude x sin 360 periode x phase
  sphere radius 3
```

Les courbes de Bézier en 3D

La courbe de Bézier d'ordre 3 est la formule suivante :

$$P0*(1-t)^3 + 3*P1*t*(1-t)^2 + 3*P2*t^2*(1-t) + P3*t^3$$
 et cela pour la variable (t) qui varie entre 0 et 1.

Cette formule doit se calculer pour toutes les abscisses des Points, c'est à dire P0x, P1x, P2x, P3x et aussi pour les ordonnées de ces points, c'est à dire P0y, P1y, P2y, P3y **et enfin pour les hauteurs P0z, P1z, P2z, P3z.**

J'ai repris exactement le même texte que pour les courbes de Bézier en 2D et en rouge ce qu'il faut ajouter pour passer en 3D. Vraiment simple ! Voici le bloc 3D Bezier3.

The screenshot shows the BlocksCAD software interface. The project name is "3D BEZIER3". The workspace displays a 3D grid with a purple Bezier curve. The left sidebar contains various tool categories. The main workspace shows a complex block structure for the 3D Bezier curve, including a loop for t from 0 to 1, and a union of translated points and spheres. A "3D Bezier3" block is highlighted, showing its parameters:

Parameter	Value
P0x	0
P0y	0
P0z	0
P1x	60
P1y	30
P1z	20
P2x	30
P2y	50
P2z	10
P3x	50
P3y	10
P3z	50
Display	1

The bottom toolbar includes a "Render" button, a "Smooth" dropdown (set to Medium), and a "Generate STL" button.

Les Block 3D Bezier4

La même chose pour le block Bézier avec 5 points.

The screenshot displays the BlocksCAD interface. On the left, a sidebar lists categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a block-based assembly:

- A **union** block containing:
 - A **count with** block (0 to 11 by 0.01) with a tooltip "Return the sum of the two numbers."
 - A **do** loop with three **translate** blocks for X, Y, and Z axes, each followed by a **sphere** block (radius 0.5).
- A **plus** block containing:
 - A **display** block.
 - A **union** block with five **translate** blocks (X, Y, Z) and five **3D text** blocks (labeled 0 to 4).
- A **3D Bezier4** block with the following parameters:
 - P0x: 0, P0y: 0, P0z: 12
 - P1x: 75, P1y: 20, P1z: 0
 - P2x: 30, P2y: 50, P2z: 0
 - P3x: 50, P3y: 60, P3z: 0
 - P4x: 60, P4y: 0, P4z: 20
 - Display: 1

The right side of the interface shows a 3D grid with a purple Bezier curve passing through five control points (small purple cubes). The control points are located at (0,0,12), (75,20,0), (30,50,0), (50,60,0), and (60,0,20). The curve starts at (0,0,12) and ends at (60,0,20). Below the 3D view are controls for Smooth (Low, Medium, High) and buttons for Render and Generate STL.

3D Bézier 5

Maintenant la version 6 points (Bézier 5).

The screenshot shows the BlocksCAD software interface. The main workspace displays a 3D grid with a purple Bézier curve and several small purple cubes. A script on the left defines the curve using a '3D Bezier5' block with 6 control points (P0x-P0z to P5x-P5z) and a 'Display' block. A tooltip points to the '3D Bezier5' block with the text 'Run the user-defined function '3D Bezier5''. The bottom right of the workspace has a 'Render' button and a 'Generate STL' button.

C'est vrai, bouger les points pour régler la courbe n'est pas très pratique. Mais faire la courbe sous un autre logiciel (beaucoup de tutoriels proposent Inkscape) n'est guère plus pratique pour des courbes simples.

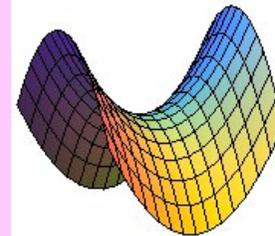
Les surfaces en 3D

Une paraboloid hyperbolique

Nous avons déjà vu que nous pouvons faire des surfaces en 3 dimensions avec LINEAR EXTRUDE et ROTATE EXTRUDE.
Pour clore ce chapitre, réaliser une surface directement à partir d'une formule.

Pour cela, encore faut-il connaître la formule. Nous allons en prendre une donnée par <http://www.mathcurve.com/surfaces/paraboloidhyperbolic/paraboloidhyperbolic.shtml> qui s'appelle communément la « selle de cheval ». Un cadeau pour Xavière et Quitrie !

$$\begin{cases} x = au \\ y = bv \\ z = h(u^2 - v^2) \end{cases}$$



L'équation cartésienne s'écrit :

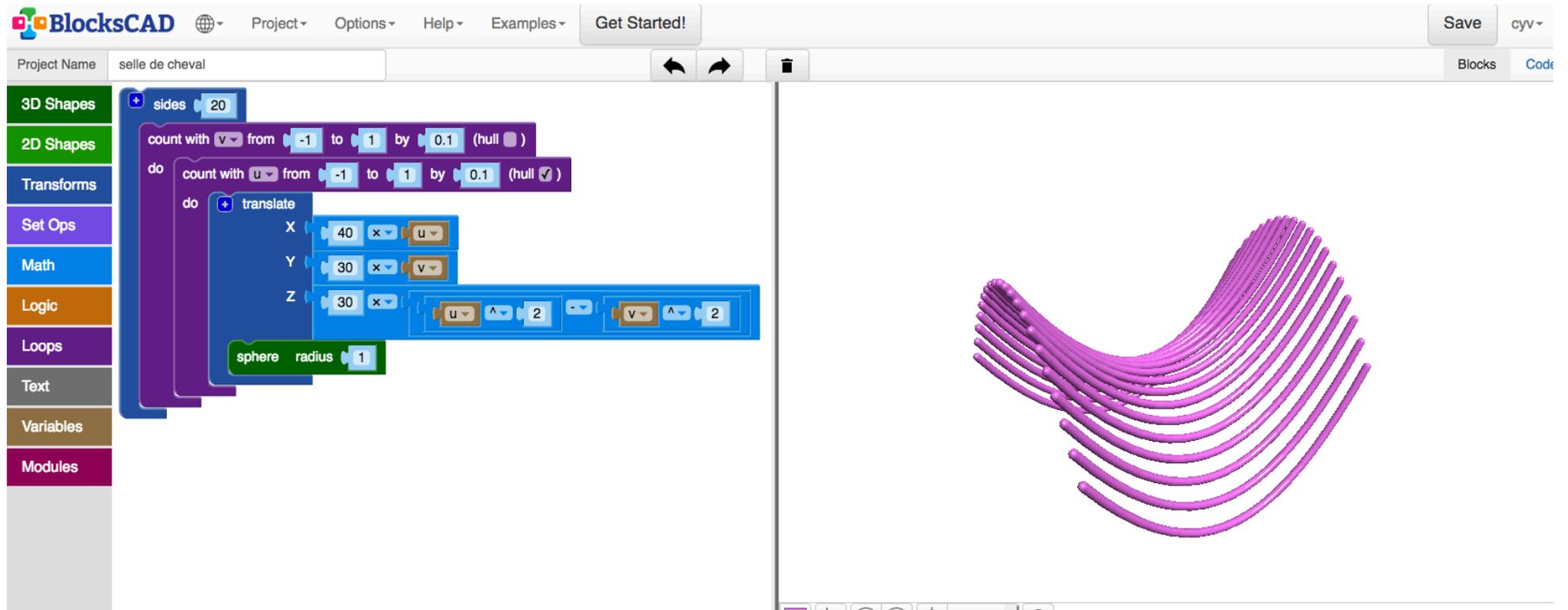
$$X = 40 * u$$

$$Y = 30 * v$$

$$Z = 30 * (u^2 - v^2)$$

Nous avons pris pour a, b, h les valeurs 40, 30,30 mais ces paramètres peuvent se changer en fonction de la taille de l'objet. Remarquez que pour une surface nous n'avons pas UNE variable (la variable « t ») mais DEUX variables (« u » et « v »).

Le block de selle de cheval



Comme nous l'avons fait déjà souvent, une petite sphère de base ; ensuite un TRANSLATE avec la formule en X, Y, et Z ; une LOOP sur « u » ; une deuxième LOOP sur « v ». Cette deuxième boucle (LOOP) est la nouveauté pour faire une surface par rapport aux courbes.

Vous avez remarqué ? La première LOOP contient l'option HULL mais pas la deuxième. Vous n'aurez qu'à essayer en cochant HULL pour les deux LOOP, le résultat est bizarre.

La fonction U^2-V^2 ou $U\wedge 2-V\wedge 2$

Pour avoir un objet en 3D et pas uniquement des tubes cotés à cotés, il faut diminuer le pas (BY) de la deuxième fonction LOOP, celle qui n'a pas l'option HULL cochée.

Au passage, nous avons un peu changé le block pour utiliser le module avec l'option RETURN. Cela permet de ne pas taper plusieurs fois une formule compliquée.

The screenshot displays the BlocksCAD interface for a project named "selle de cheval". The left sidebar shows various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains the following code blocks:

- A "sides" block set to 20.
- A "count with" block with "v" selected, ranging from -1 to 1 by 0.05, with the "hull" option unchecked.
- A "do" block containing:
 - A "count with" block with "u" selected, ranging from -1 to 1 by 0.1, with the "hull" option checked.
 - A "do" block containing:
 - A "translate" block with X=40, Y=30, and Z=30. The Z-axis uses a "U2-V2 with:" module.
 - A "sphere" block with a radius of 1.
- A "to U2-V2 with: u, v" block with a "return" block containing the formula $u^2 - v^2$.

The 3D view on the right shows a purple saddle-shaped surface composed of many thin tubes. The interface includes a "Render" button, a "Smooth" dropdown set to "Medium", and a "Generate STL" button.

Le souci principal de diminuer le pas, c'est que le programme est très long et que le lissage peut encore s'améliorer.

Pour améliorer, nous allons inverser les LOOP en commençant par celle de « u » puis celle de « v ». Cette dernière sera cochée HULL désormais.

The screenshot shows the BlocksCAD interface. On the left is a sidebar with categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains two identical code blocks for creating a hyperbolic paraboloid. Each block starts with a 'sides' block set to 20. It then uses a 'count with v' block (from -1 to 1 by 0.1) to loop over the vertical direction. Inside this loop is a 'do' block with a 'count with u' block (from -1 to 1 by 0.1) for the horizontal direction. Within the 'u' loop, there is a 'translate' block with X=40, Y=30, and Z=30, all multiplied by the 'u' variable. A 'U2-V2 with:' block is used to calculate the coordinates for a 'sphere' block with a radius of 1. The 'u' and 'v' variables are squared in the 'U2-V2' block. The 3D view on the right shows a purple mesh of the hyperbolic paraboloid. Below the 3D view are controls for rendering, including a 'Render' button, a 'Smooth' dropdown (set to Medium), and a 'Generate STL' button.

Je ne résiste pas plus longtemps pour vous donner le nom officiel de cette surface : la parabolôide hyperbolique. Trop stylés ces mathématiciens !

Et bien jouons à Star Wars dit la cigale !

Maintenant, nous pouvons jouer en changeant la formule de Z et même celles de X et Y et voir ce que cela donne.

Par exemple, nous avons simplement changer la formule (u^2-v^2) en (u^2+v^2) : un plus à la place du moins.

The image shows a Scratch code editor window titled "paraboloïde". The script is as follows:

```
when green flag clicked
  sides = 20
  loop from v = -1 to 1 by 0.1 (hull)
  do loop from u = -1 to 1 by 0.1 (hull)
    translate
      X = 40 * u
      Y = 30 * v
      Z = 30 * (u^2 + v^2)
    sphere radius = 1
```

The 3D rendering on the right shows a purple paraboloid shape, which is a surface of revolution. The shape is composed of many thin, curved lines that form a smooth, upward-curving surface. The color of the lines is a vibrant purple.

Et voici le résultat. Encore quelques modifications et nous aurons une coquille de Caliméro ou un casque de Darth Vader ?

Je vous laisse modifier et essayer des formules inventées ou vues sur des sites de mathématiques.

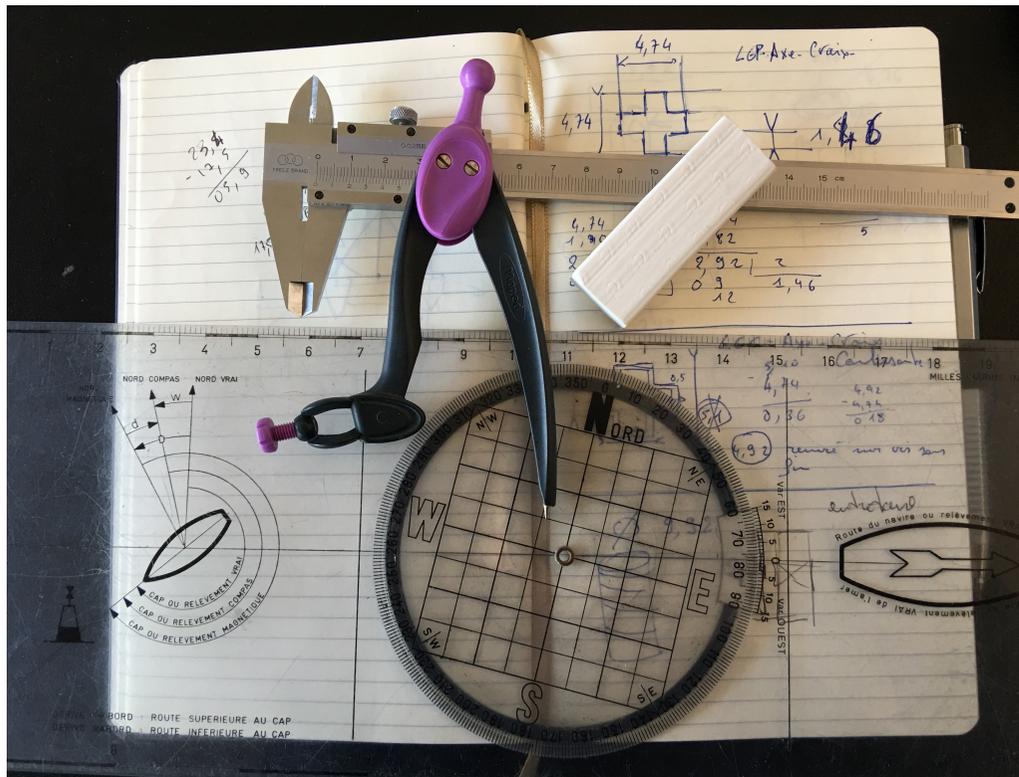
Jeux de construction

Le pied à coulisse

Avant de nous lancer dans les autres jeux de construction, j'aimerais vous parler du pied à coulisse. C'est un petit outil très utile dans le bricolage pour bien mesurer les pièces pas trop grandes.

De plus, pour un pied à coulisse à vernier, le système qui permet d'être précis est vraiment malin : une deuxième graduation légèrement décalé par rapport à la première. Vous trouverez facilement sur le Net comment vous en servir. https://www.youtube.com/watch?v=J0bYD3_n1Z4

J'en ai un depuis plus de 30 ans qui fonctionne parfaitement.



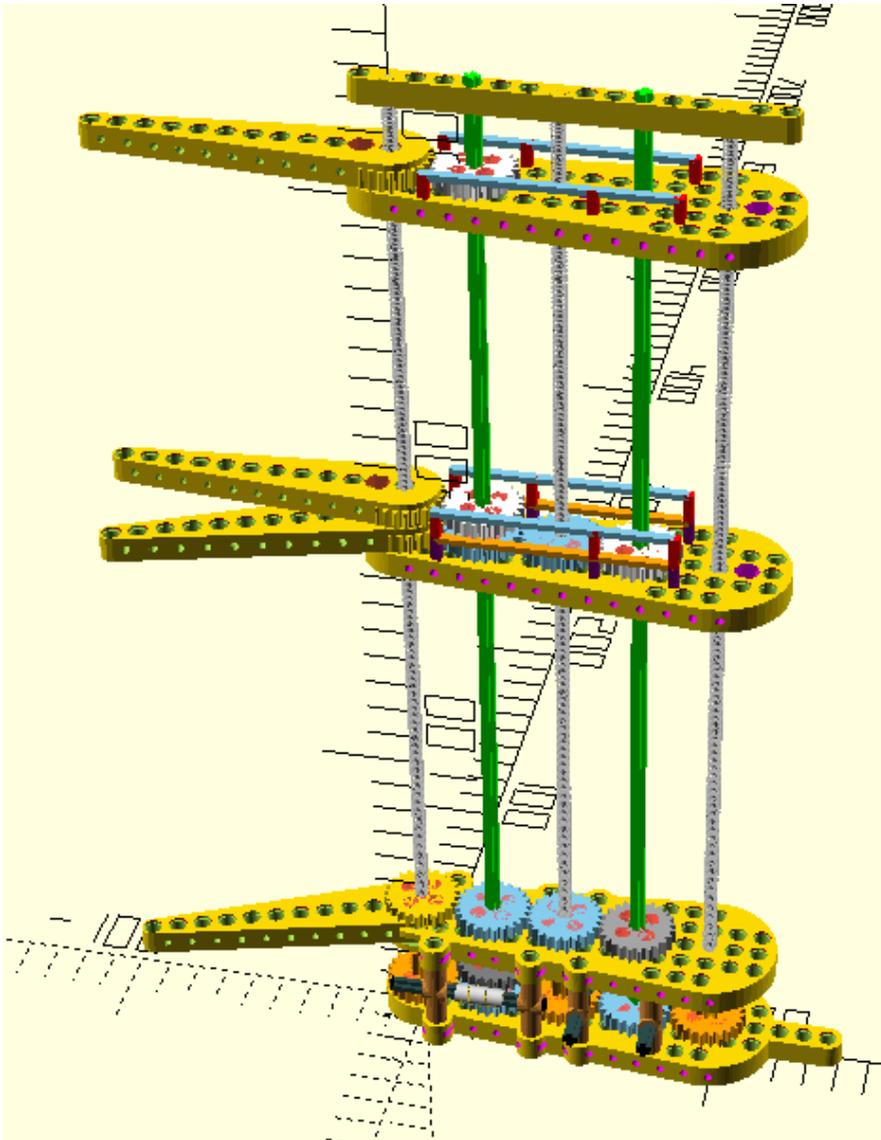
Et si vous rajouter un porte-mine, une gomme, un compas, une règle (j'utilise une règle de navigation), et un carnet de notes, le monde de votre imaginaire est à vos pieds.



Quittrie vous conseille de rajouter aussi de la colle et des ciseaux pour que votre imaginaire se réalise.

Lego®, Meccano® et tous les autres

Il existe de nombreuses boîtes de construction qui permettent presque tout. C'est ce presque qui justifie ce paragraphe. Il arrive que pour certains montages il faille des pièces spéciales. Alors pas de soucis, vous pouvez les faire sur mesure avec une imprimante 3D.



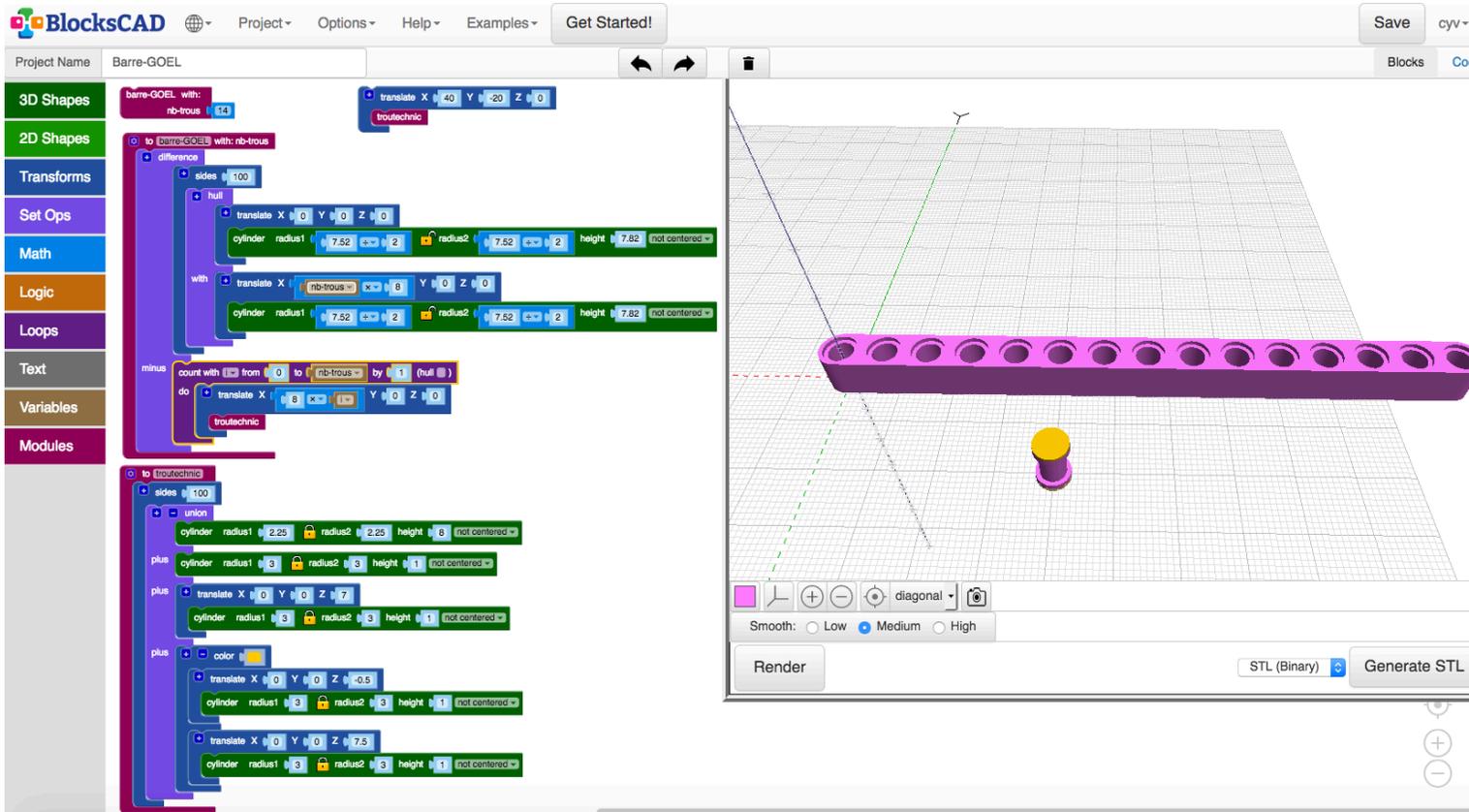
Le pied à coulisse est très important pour réaliser des pièces compatibles avec votre jeu de construction préféré car les mesures sont en dixièmes de millimètre et il est important d'être précis.

Nous n'allons pas rentrer dans le détail d'une telle construction, au demeurant fort peu utile, mais les grosses pièces jaunes sont toutes sur-mesure tout en restant compatible avec les dimensions LEGO®. Cela permet de ne réaliser que le strict nécessaire pour une maquette.

Une barre aux dimensions Lego Technic®

Armé du pied à coulisse, et d'une vraie pièce originale, nous avons pris les différentes mesures :

- Mesures externes de la barre,
- Mesures interne des trous.



Les mesures du trou permettent de réaliser le module TROUTECHNIC qui est encadré de deux cylindres, en jaune, car ce module sera utilisé dans une DIFFERENCE.

Le module Barre-GOEL qui réalise une coque (HULL) de deux cylindres en extrémités et puis une différence avec autant de module TROUTECHNIC que nécessaire.

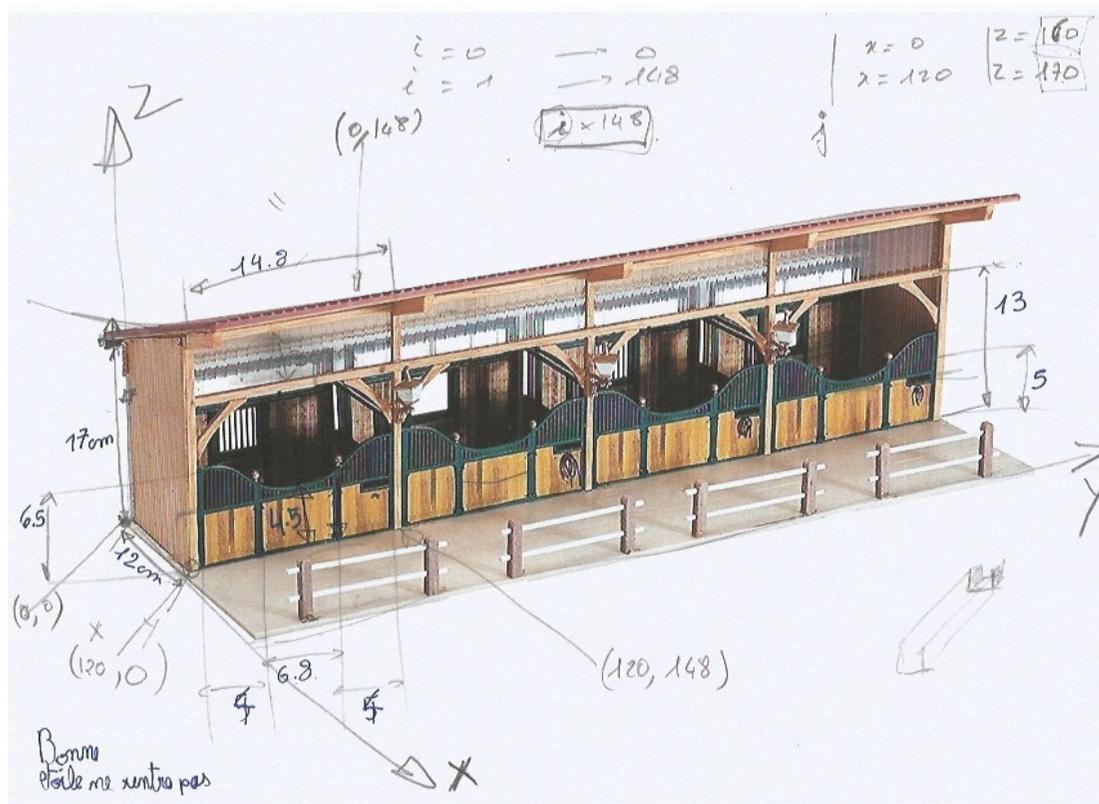
Initialement, nous avons repris des mesures dans des forums sur Internet. Nous conseillons vivement de reprendre chacun ses

propres mesures, de préférence avec un pied à coulisse au 50^{ème}.

Un projet plus complet : l'écurie

Démarrer un projet

Démarrer un projet commence par une idée, un besoin utile ou inutile, un dessin. En surfant sur Internet, nous sommes tombés sur une photo d'une écurie. Idéal pour en faire une sur mesure et plus adaptée aux besoins de rangement de la horde miniature de Quitrie et Xavière.



Donc voilà la photo trouvée sur Internet. Dans un premier temps, il a fallu trouver les côtes. C'est assez simple, quoique !

Quitrie, armée d'une règle et du pied à coulisse a défini les mesures pour que la plupart des chevaux puissent rentrer. A ce propos, regardez en bas à droite : Bonne Etoile, un alezan de mémoire, est un peu grand et ne rentrera pas. Pauvre bête ;-)

Ensuite, on trace les axes et l'origine. On les choisit comme on veut mais il faut le faire et s'y tenir. Et puis ensuite on peut commencer.

Premier pas : la structure

Voici un début de structure avec un plancher en gris, et diverses poutres.

The screenshot displays the BlocksCAD software interface. On the left, a sidebar lists various categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace is divided into two panes. The left pane shows a hierarchical tree of blocks under the name 'ecurie'. The right pane shows a 3D rendering of the structure, which consists of a grey floor and several vertical beams of different colors (blue, green, yellow, red) forming a frame. The 3D view includes a grid and a toolbar with options like 'diagonal', 'Smooth: Low Medium High', and 'Render'. At the bottom right, there are buttons for 'STL (Binary)' and 'Generate STL'.

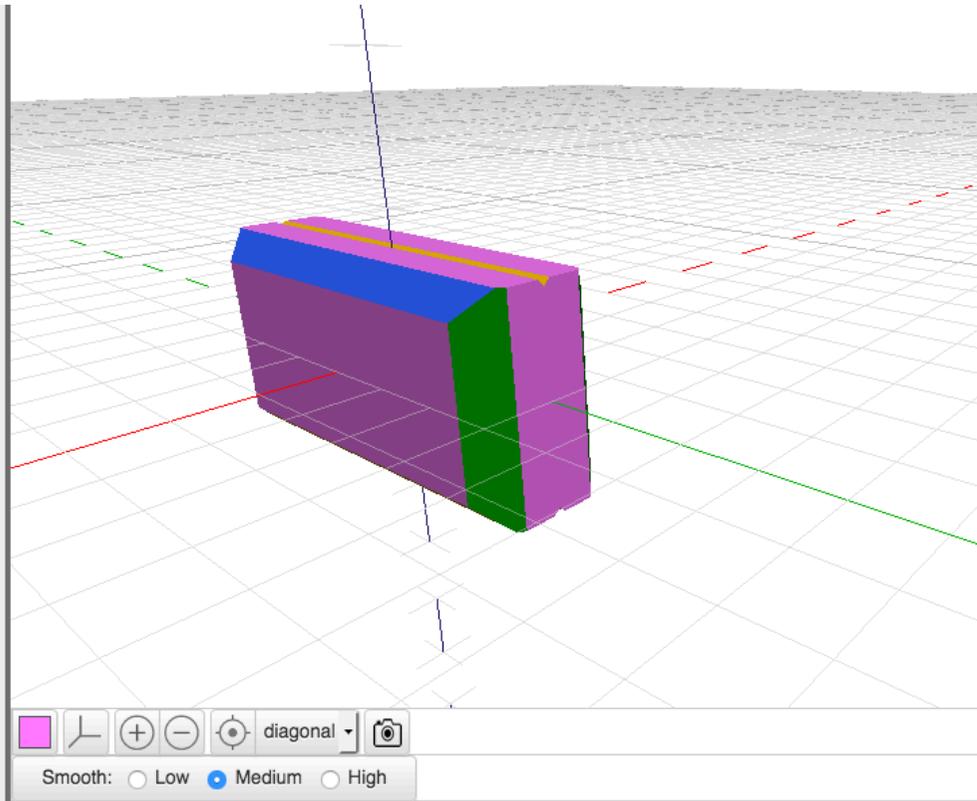
The block structure in the left pane is as follows:

- STRUCTURE
 - to STRUCTURE
 - union
 - count with [] from 0 to 1 by 1 (null)
 - do
 - union
 - color []
 - translate X [] Y [] Z []
 - cube X [] Y [] Z [] not centered
 - plus
 - color []
 - translate X [] Y [] Z []
 - cube X [] Y [] Z [] not centered
 - plus
 - color []
 - translate X [] Y [] Z []
 - cube X [] Y [] Z [] not centered
 - plus
 - color []
 - translate X [] Y [] Z []
 - cube X [] Y [] Z [] not centered
 - plus
 - color []
 - cube X [] Y [] Z [] not centered

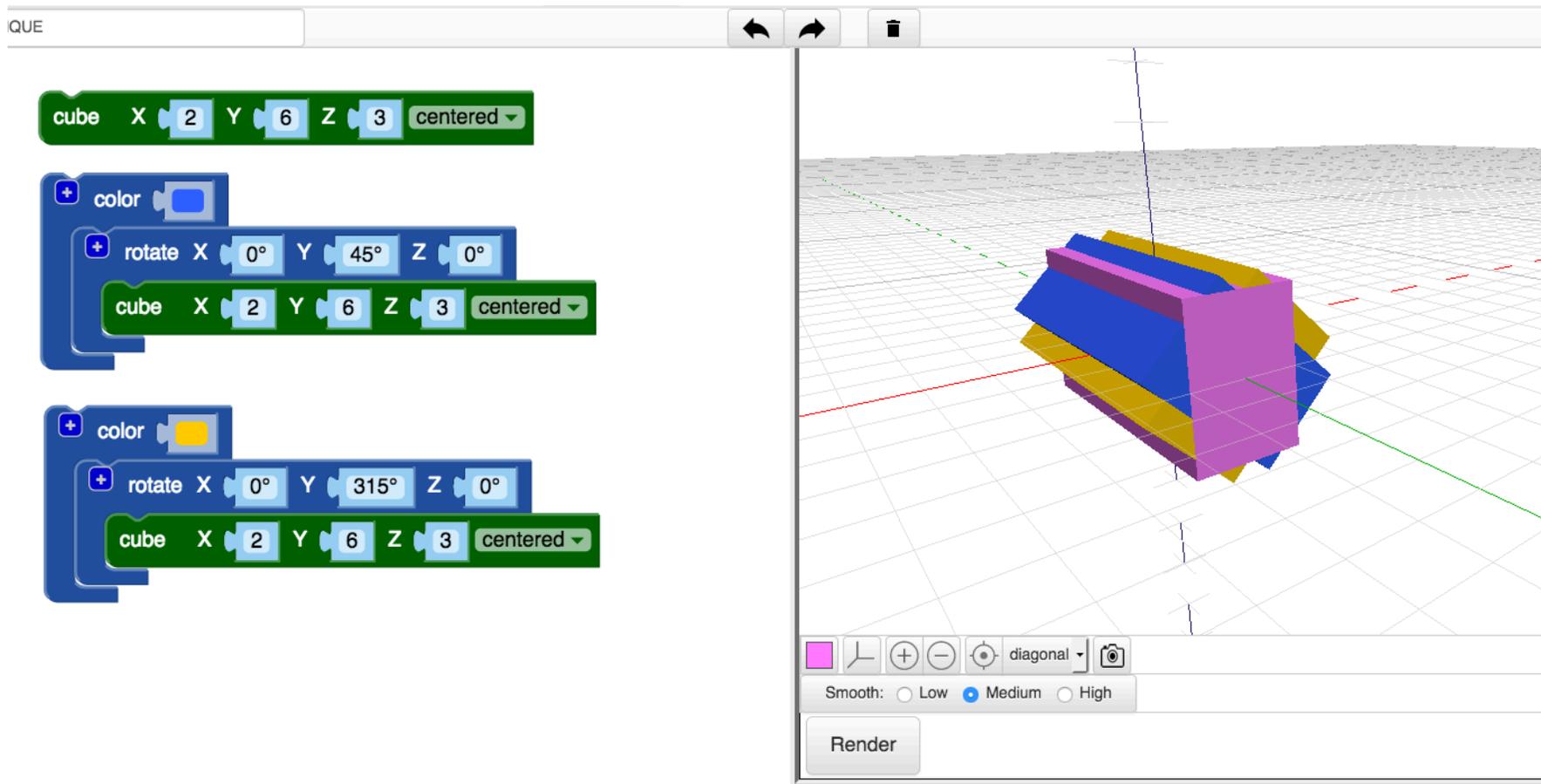
A chaque fois, nous avons 2 montants de mêmes couleurs. Bien sûr, il était facile de dupliquer les blocks mais, cette structure a été l'occasion de faire un petit jeu (trop horrible vous dirait Quitrie) : la boucle se fait uniquement avec une variable qui prend les valeurs 0 et 1. Le petit jeu est donc de trouver les formules qui correspondent aux mesures que l'on souhaite et qui sont sur le schéma.

Les briques

Pour faire un mur, nous avons besoin de briques. Nous allons dessiner des briques biseautées, c'est à dire avec des chanfreins (n'est ce pas un joli mot, bien que pas facile à utiliser sur les réseaux sociaux).

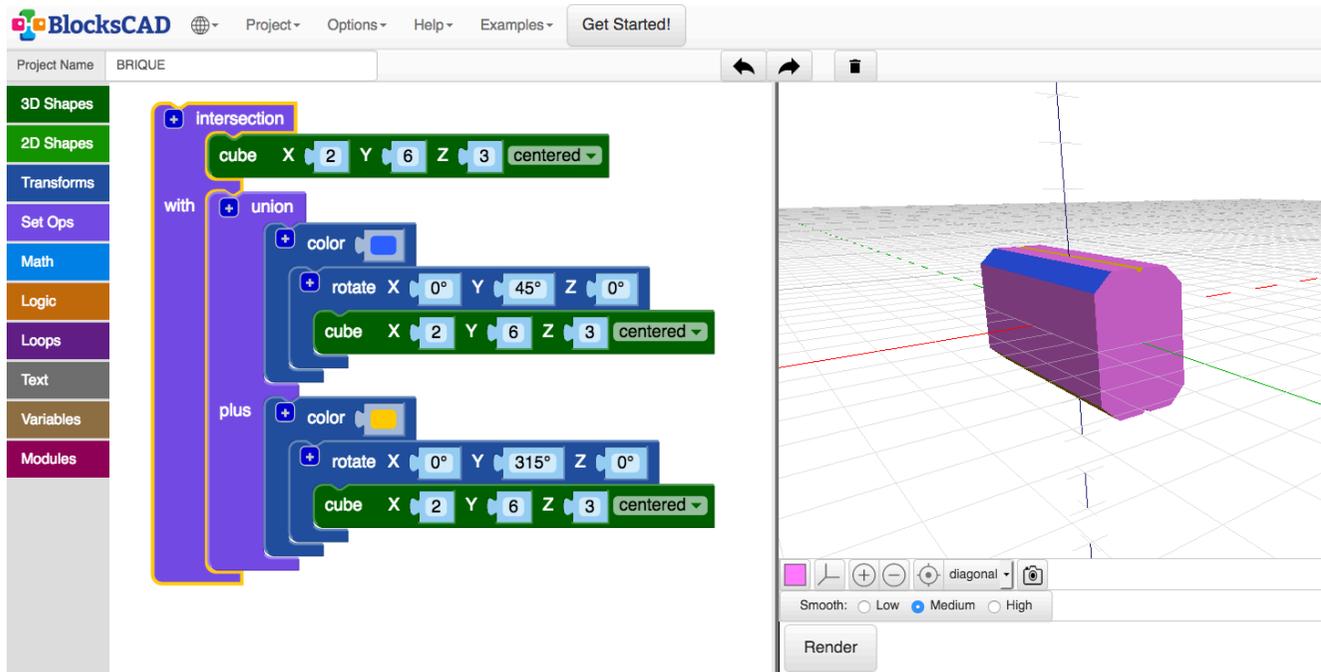


Voici la tête de notre brique en final. Pour arriver à ce résultat nous avons commencé par une brique parallélépipède rectangle, mot savant qui désigne un cube allongé.

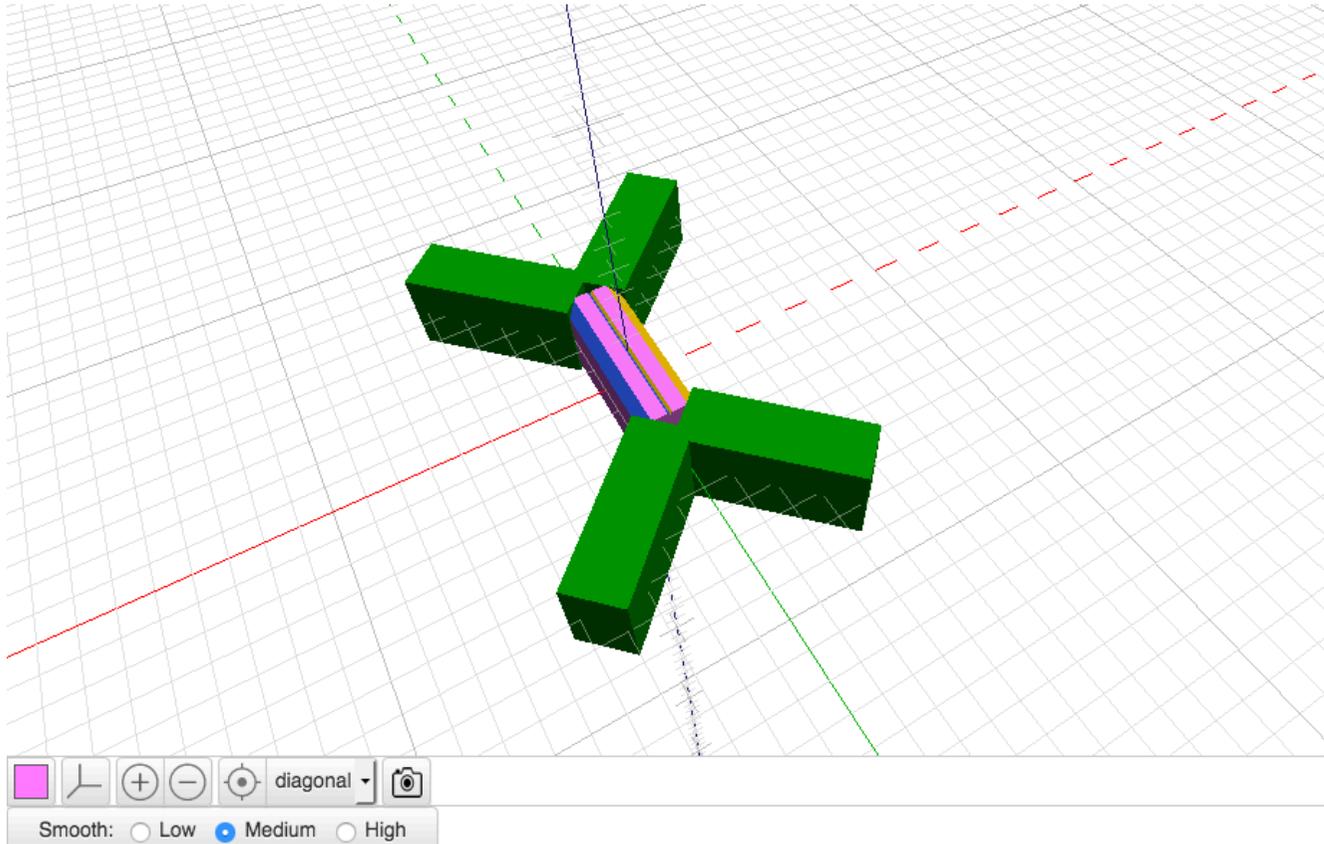


Donc en mauve, notre brique de base. En bleu, la même brique légèrement tournée et en jaune, encore la même chose mais tournée dans l'autre sens.

Il nous suffit de faire INTERSECTION pour supprimer les arrêtes mauves qui dépassent. Je vous laisse faire néanmoins quelques essais, par exemple, ne faites pas INTERSECTION en premier mais UNION des 2 briques en rotation auparavant : le résultat n'est pas le même !

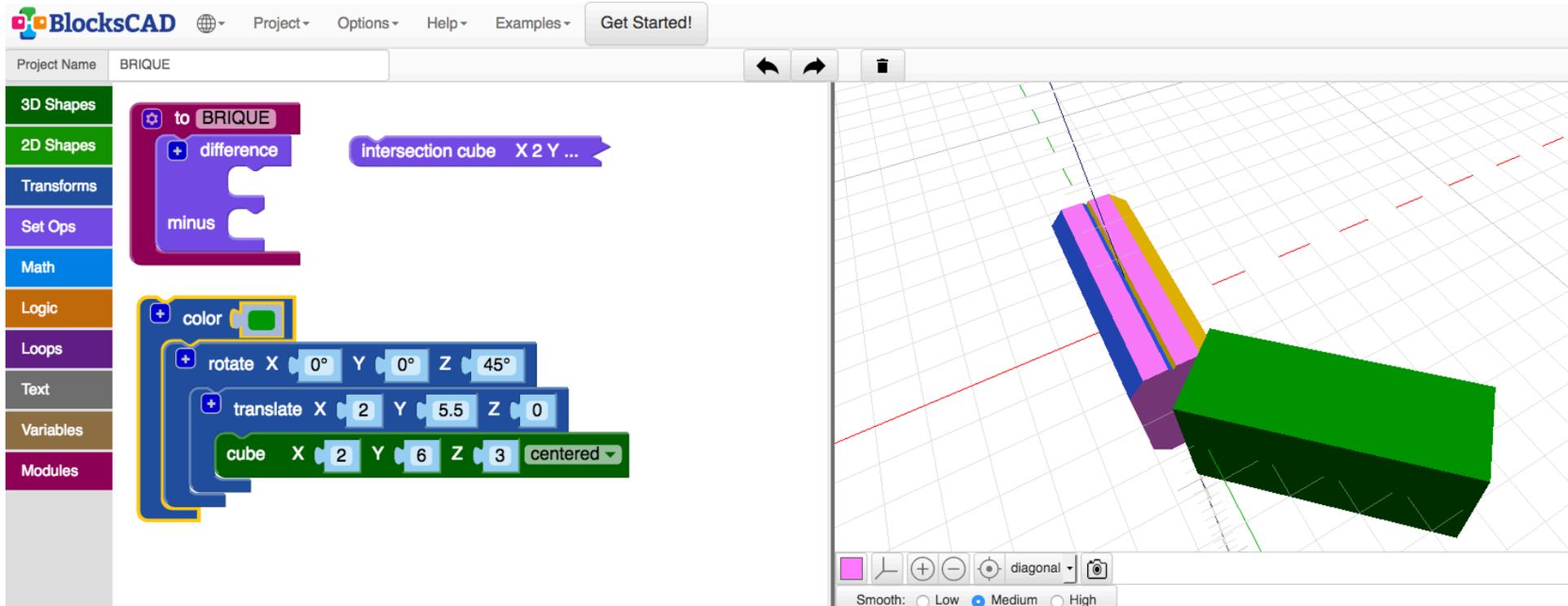


Nous avons nos chanfreins sur les arrêtes horizontales, réalisons ceux pour les 4 arrêtes verticales.
Le principe est simple, faire 4 blocs que nous allons retrancher (DIFFERENCE) de la brique mauve. Regarder dessous les 4 blocs en vert.



Bien sûr, il est assez simple de faire les 4 blocks, avec un ROTATE et un TRANSLATE pour chacun et de les positionner comme on veut. Pour nous amuser un peu, nous allons essayer de deviner une formule qui nous donne automatiquement ces 4 blocs : « Mais c'est trop horrible ». Surement, mais cela fait un peu travailler les neurones.

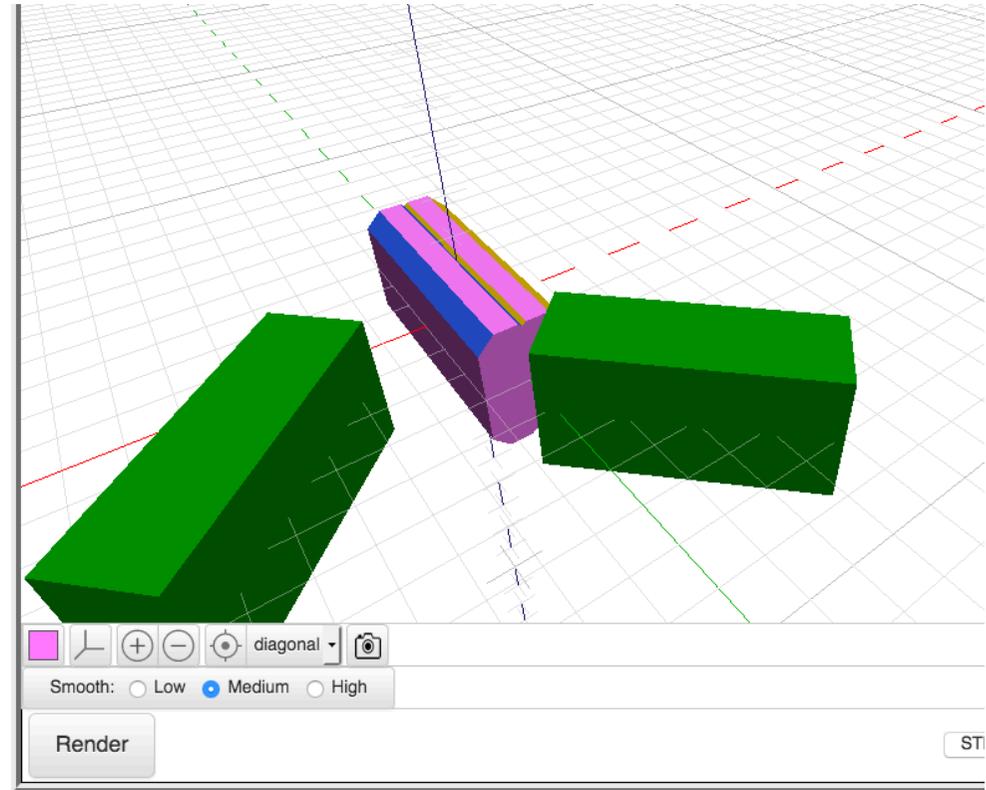
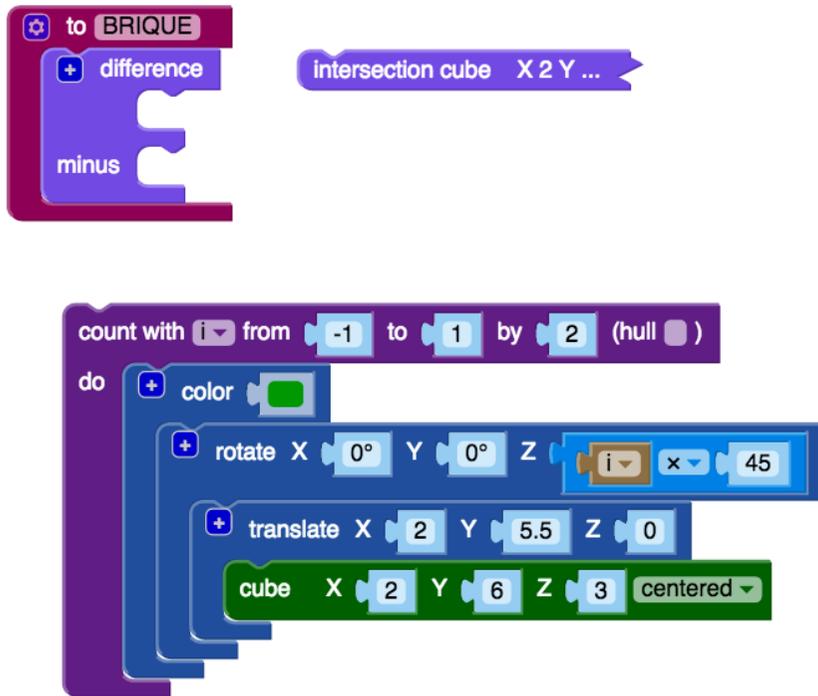
Commençons donc par placer un bloc, le tourner un peu (45 degrés) et le décaler pour qu'il masque une arête.



Les valeurs de Z (45 degrés), de Y (5.5) et de X (2) sont trouvées par tâtonnement.

Nous allons continuer par une boucle (LOOP) avec la variable (*i*) qui va prendre les valeurs suivantes : -1, +1. L'idée est de se dire que les valeurs 2, 5.5 et 45° sont le résultat de formule qui utilise la variable *i*.

Essayons déjà en multipliant 45° par i .



Cela marche pour l'angle mais il faut aussi changer X. Multiplions i par 2 pour le TRANSLATE.

The screenshot shows the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', 'Examples', and 'Get Started!'. The project name is 'BRIQUE'. The left sidebar lists categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains a 3D model of a brick on a grid. The code blocks are as follows:

```

to BRIQUE
  difference
    intersection cube X 2 Y ...
    minus
      count with i from -1 to 1 by 2 (hull )
      do
        color
        rotate X 0° Y 0° Z i x 45
        translate X i x 2 Y 5.5 Z 0
        cube X 2 Y 6 Z 3 centered
  
```

The 3D model shows a green brick with a purple and blue section on top. The code blocks are color-coded: purple for the 'to' block, blue for 'difference', 'count with', and 'do', green for 'color', 'rotate', and 'cube', and blue for 'intersection cube' and 'translate'. The 'translate' block's X coordinate is highlighted with a yellow box.

Et voilà, nous avons trouvé la première partie de notre formule. Passons à la deuxième partie.

Pour cela, nous allons faire une deuxième boucle avec la variable (j). Dans la formule précédente, nous voyons que Y ne bouge : essayons donc de modifier Y en fonction de la variable (j). Quand (j) prend la valeur +1, cela ne change pas la formule d'avant et quand (j) prend la valeur -1, et bien regardons !

The screenshot shows the BlocksCAD interface with a project named "BRIQUE". The left sidebar contains a menu with categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace displays a 3D model of a brick with a central hole, rendered in green. The code blocks are as follows:

- to BRIQUE** block containing:
 - difference** block with a sub-block **intersection cube X 2 Y ...**
 - minus** block
- count with** block: from **-1** to **1** by **2** (hull)
- do** block:
 - count with** block: from **-1** to **1** by **2** (hull)
 - do** block:
 - color** block: green
 - rotate** block: X **0°**, Y **0°**, Z **i** x **45**
 - translate** block: X **i** x **2**, Y **j** x **5.5**, Z **0**
 - cube** block: X **2**, Y **6**, Z **3**, centered **centered**

The 3D view shows the brick model with a central hole. The code blocks are color-coded: purple for loops, blue for math, and green for shapes. The **translate** block's X and Y fields are highlighted with a yellow box.

Pas loin, n'est ce pas ?

Il faut donc surement compléter la formule en déplaçant un peu X. Mettons la variable (j) dans le X du TRANSLATE et voyons le résultat.

$$X = (i)*(j)*2$$

The screenshot shows a 3D modeling software interface with a project named "BRIQUE". The left sidebar lists categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains a script with the following blocks:

- to BRIQUE** (purple block)
- difference** (purple block) with a sub-block **intersection cube X 2 Y ...** (purple block)
- minus** (purple block)
- count with i** from **-1** to **1** by **2** (hull) (purple block)
- do** (purple block) containing:
 - count with i** from **-1** to **1** by **2** (hull) (purple block)
 - do** (purple block) containing:
 - color** (green block)
 - rotate X** **0°** **Y** **0°** **Z** **i** **x** **45** (blue block)
 - translate X** **i** **x** **j** **x** **2** **Y** **j** **x** **5.5** **Z** **0** (blue block)
 - cube X** **2** **Y** **6** **Z** **3** **centered** (green block)

The 3D view on the right shows a green 3D model of the resulting shape, which is a central intersection of three cubes. The interface includes a "Render" button and a "Smooth" dropdown menu with options: Low, Medium (selected), and High. An "STL" button is also visible.

et voilà, c'est parfait. Bien sûr, nous aurions pu réfléchir dans nos têtes pour imaginer la formule avant de la coder mais pourquoi se compliquer la vie ?

En plus, je trouve l'exercice intéressant car il permet de découvrir la formule par étape.

Maintenant il suffit de faire une petite DIFFERENCE, et nous voilà prêts, tel le petit cochon de la fable, à construire notre mur.

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', and 'Examples', along with a 'Get Started!' button. The project name is 'BRIQUE'. On the left, a sidebar lists various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D view of a brick with a grid background. The brick is a rectangular prism with a notch on one side. The code for the brick is as follows:

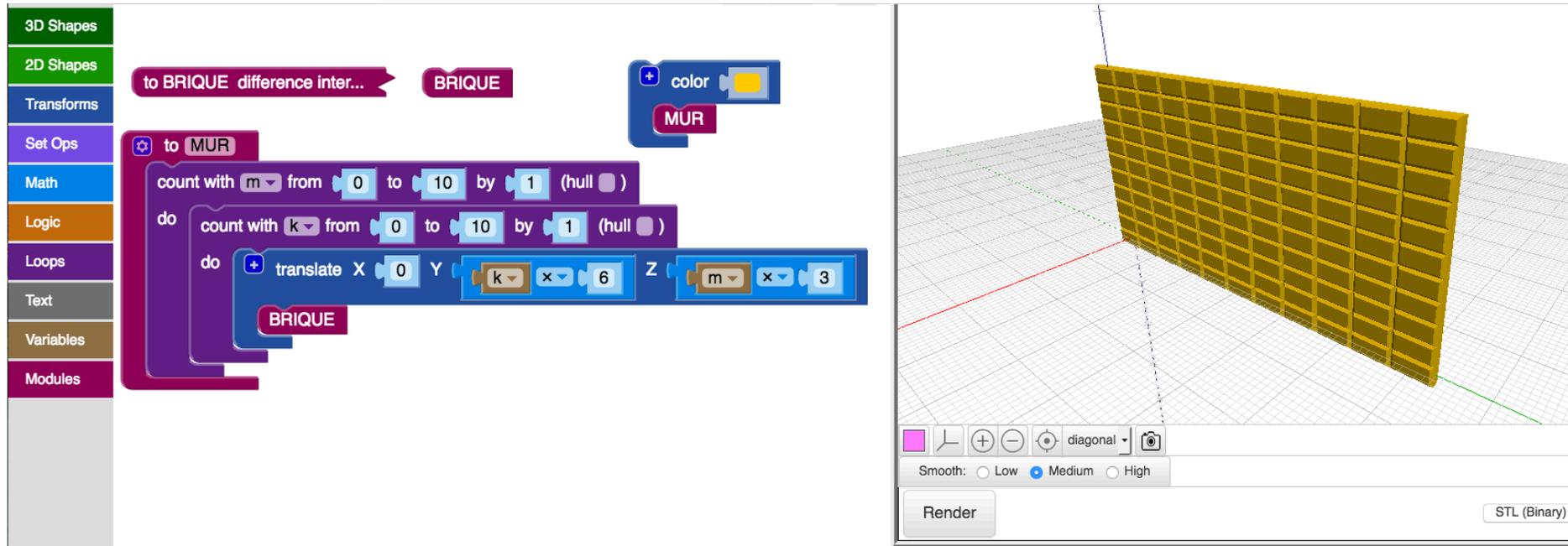
```
to BRIQUE
  difference
    intersection cube X 2 Y ...
  minus
    count with i from -1 to 1 by 2 (hull )
    do
      count with i from -1 to 1 by 2 (hull )
      do
        color
        rotate X 0° Y 0° Z i x 45
        translate X i x 2 Y i x 5.5 Z 0
        cube X 2 Y 6 Z 3 centered
```

The 3D view shows the brick with a grid background. The brick is a rectangular prism with a notch on one side. The color of the brick is purple. The grid is white with red and green axes. The 3D view includes a toolbar with icons for color, zoom, and rotation, and a 'Render' button.

Le mur de briques

Le mur se fait par empilage de briques :

- ❑ une première boucle avec la variable (k) qui translate de 6 (la taille d'une brique) chaque nouvelle brique dans l'axe des Y
- ❑ une deuxième boucle avec la variable (m) qui translate les briques de 3 (la hauteur d'une brique) sur l'axe Z



Un peu décevant, n'est pas ? nous voulons un mur et non pas une tablette de chocolat. Pour avoir un joli, mur il faudrait décaler les briques les rangées horizontales d'une demi-longueur de brique (c'est à dire de 3).

Donc, pour chaque ligne pour laquelle la variable (m) est paire, on va rajouter le chiffre 3 à Y. C'est à

Là qu'intervient la fonction mathématique REMAINDER OF. Cette fonction donne le reste d'une division entière (petit rappel, division entière signifie qu'il n'y a pas de virgule et qu'il peut y avoir un reste).

Pour savoir si la variable (m) est paire, il suffit de la diviser par 2 et de voir si le reste est zéro. Si m est impair, le reste sera de 1.

Donc à notre formule initiale $K+6$ on rajoute REAMINDER OF $m \div 2$ (qui fait 0 ou 1 suivant que le ligne est paire ou impaire) que l'on multiplie par 3 :

- si m est paire, la formule devient $K+6+0$
- si m est impaire, la formule devient $K+6+3$

Ce que nous cherchions.

REMAINDER OF

Juste pour terminer avec REMAINDER OF, voici un autre exemple, pour voir à quoi cela peut servir. Au lieu de diviser par 2 la variable (m), nous la divisons par 3. REMAINDER OF va prendre successivement les valeurs : 0, 1, 2, 0, 1, 2, 0,

Si en plus on décale de 2 au lieu de décaler chaque nouvelle brique de 3, cela donne un mur un peu différent du précédent.

The screenshot displays a 3D modeling software interface. On the left, a sidebar lists various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace shows a 3D view of a brick wall. The code blocks are as follows:

- Project Name:** BRIQUE
- to BRIQUE difference inter...:** BRIQUE
- color:** MUR
- to MUR:**
 - count with m from 0 to 10 by 1 (hull)
 - do
 - count with k from 0 to 10 by 1 (hull)
 - do
 - translate X 0 Y 0 Z remainder of m + 3 x 2
 - BRIQUE

The 3D view shows a brick wall with a staggered pattern. The wall is composed of grey bricks. The interface includes a 'Render' button and a 'Smooth' dropdown menu set to 'Medium'. The 'STL (Binary)' format is selected for export.

Le module MUR et ROUND

Le programme précédent suffit amplement pour soi-même. Cependant, quand on veut que d'autres personnes puisse l'utiliser facilement il va falloir un peu le modifier.

En effet, pour le rendre plus simple à utiliser nous allons construire un module ou il suffira de donner la longueur du mur et sa hauteur pour avoir un beau mur aux dimensions requises.

Abandonnons pour l'instant le block précédent, et commençons à la manière accoutumée des maçons à définir le cadre du mur.

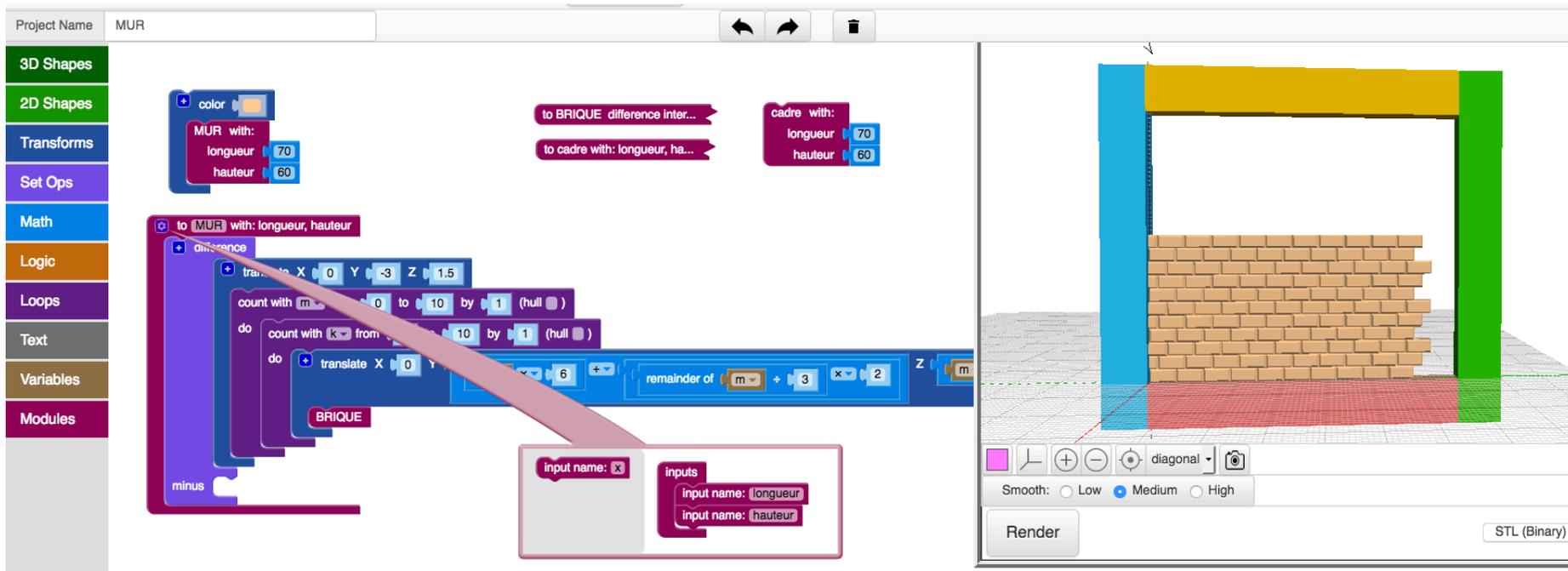
The screenshot shows the BlocksCAD interface with a project named 'MUR'. The block structure on the left is a 'to cadre' module with two inputs: 'longueur' and 'hauteur'. It contains four 'plus' blocks, each with a 'color' block, a 'translate' block, and a 'cube' block. The 3D model on the right shows a blue wall frame on a grid, with a yellow top bar, a green right side, and a red bottom bar. A pink callout box highlights the 'input name' fields for 'longueur' and 'hauteur'.

4 éléments de 4 couleurs pour bien les voir, regroupés au sein d'un module qui possède 2 variables (longueur) et (hauteur). Les TRANSLATE présents dans le module sont là pour que l'espace intérieur soit exactement.

Pour indiquer à un module qu'il doit avoir des variables, il suffit de :

- Cliquer le petit engrenage en haut à gauche du module
- Ecrire le nom de la variable : « hauteur » par exemple
- Glisser ce nom de la partie gauche de la fenêtre dans la partie droite à l'intérieur de la « machoire » INPUT.
- Recommencer autant de fois que nécessaire pour les variables suivantes.
- Cliquer à nouveau sur l'engrenage pour faire disparaître cette fenêtre de saisie..

Revenons à notre mur dans son cadre, et insérons pour le module MUR nos deux variables (longueur) et (hauteur).



Il reste maintenant à faire correspondre les variables (m) et (k) avec (longueur) et (hauteur) avec la bonne formule.

On sait que la brique vaut 6 de long, on sait que (k) est la variable du nombre de briques sur la longueur. Il faut donc faire la boucle (LOOP) « longueur ÷ 6 ». Cependant comme K doit être un entier, nous rajoutons la fonction ROUND UP qui donne l'entier immédiatement supérieur.

Pour la hauteur c'est le même principe, mais avec la variable (m) et la hauteur de brique 3. La formule est donc « hauteur ÷ 3 » avec la fonction ROUND UP.

The screenshot displays the BlocksCAD interface for a project named "MUR". The left sidebar shows various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains a 3D model of a brick wall with a blue left side, a yellow top, and a green right side. The wall is composed of brown bricks on a red grid base. The code blocks are as follows:

- A **color** block with a yellow swatch.
- A **MUR with:** block with **longueur** set to 70 and **hauteur** set to 60.
- A **to BRIQUE difference inter...** block.
- A **cadre with:** block with **longueur** set to 70 and **hauteur** set to 60.
- A **to [MUR] with: longueur, hauteur** block containing:
 - A **difference** block with a **translate** block (X: 0, Y: -3, Z: 1.5).
 - A **count with** block: **m** from 0 to **round up** (hauteur ÷ 3) by 1 (hull).
 - A **do** block: **k** from 0 to **round up** (longueur ÷ 6) by 1 (hull).
 - A **do** block: **translate** (X: 0, Y: **k** × 6, Z: **remainder of** (m ÷ 3) × 2) with a **BRIQUE** block.
- A **minus** block.

The 3D view includes a toolbar with navigation and rendering options (Smooth: Low, Medium, High) and a **Render** button.

Et voilà nos briques qui remplissent tout l'espace de notre cadre. Bien sûr, si vous changez la taille des briques, il faudra changer les formules, mais je vous laisse le soin de trouver comment.

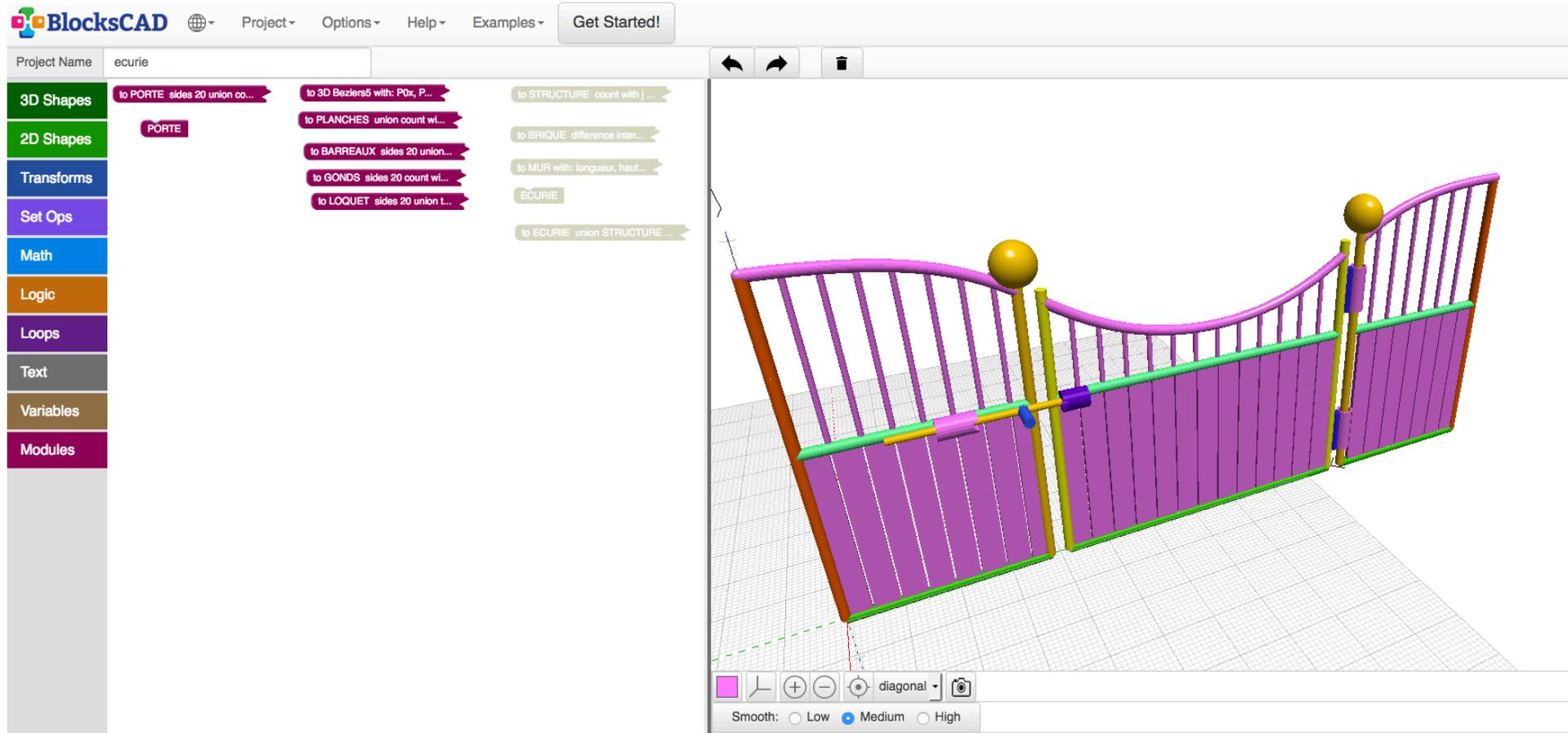
Pour finir, supprimons le cadre et au passage les briques qui dépassent du cadre. Ainsi va s'achever le module MUR.

The screenshot displays the BlocksCAD interface with a project named "MUR". The left sidebar shows various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace is filled with a complex script of code blocks. The script starts with a "to (MUR) with: longueur, hauteur" block, followed by a "difference" block. Inside the "difference" block, there is a "translate" block (X: 0, Y: -3, Z: 1.5), a "count with" block (from 0 to round up of hauteur / 3 by 1 (hull)), and a "do" block containing another "count with" block (from 0 to round up of longueur / 6 by 1 (hull)) and a "do" block with a "translate" block (X: 0, Y: k, Z: remainder of m / 3 * 2) and a "BRIQUE" block. Below the "difference" block, there are "union" and "plus" blocks. The "union" block contains a "color" block, a "translate" block (X: 0, Y: longueur / 2, Z: hauteur / 5), and a "cube" block (X: 4, Y: longueur, Z: 10, centered). The "plus" block contains a "color" block, a "translate" block (X: 0, Y: longueur / 5, Z: hauteur / 2), and a "cube" block (X: 4, Y: 10, Z: hauteur / 20, centered). A tooltip explains: "Subtracts one or more objects from the first object in the list." The right side of the interface shows a 3D view of a brick wall on a grid. Below the 3D view is a control panel with a "Render" button and a "Smooth" dropdown menu (Low, Medium, High). The "STL (E)" button is also visible.

Dans ce module, nous avons fait le choix de rester simple dans les paramètres, c'est à dire ce que l'on peut choisir : la longueur et la hauteur. Nous aurions pu aussi mettre sous forme de paramètres : la taille des briques et le décalage d'une rangée sur l'autre. C'es surement un exercice intéressant mais n'oubliez pas que plus on met de possibilités, plus cela devient peu compréhensible pour le suivant.

Le portail et les montants

Voici la belle porte de box que Quitrie et Xavière vous ont préparée.



Ce module est constitué de plusieurs sous-modules. Le premier est la courbe ondulée du portail. Ce n'est pas pour rien que nous avons codé les courbes de Monsieur Bézier.

The screenshot displays the BlocksCAD environment. On the left, a sidebar lists various modules, with 'Modules' selected. The main workspace is split into a code editor and a 3D view. The code editor shows a sequence of blocks: a 'to B&B (Beziers)' block, a 'union' block, and several 'translate' and 'rotate' blocks. The 3D view shows a purple curved beam on a grid, with points 0, 1, 2, 3, 4, and 5 marked along its length. A 'Render' button and a 'Smooth' dropdown are visible at the bottom of the 3D view.

Le module d'appel à gauche et le module Bézier avec le paramètre DISPLAY à 1. Ce paramètre sera bien entendu mis à zéro pour que les chiffres qui indiquent la place des points de Bézier n'apparaissent plus.

Avouons le, il faut un certain temps pour bien placer les points. Ensuite, les formules font le reste.

La suite du programme, ne fait que rajouter la structure de la porte. En regardant les couleurs sur le dessin et les couleurs dans le code informatique, il est facile de comprendre cette partie. Un petit truc sur la courbe de Bézier : la droite entre le point 0, celui du départ, et le point 1 est la tangente au point 0. Cela permet de placer le point 1. C'est aussi vrai le point 5 : la tangente à ce point est la droite entre point 5 et point 4.

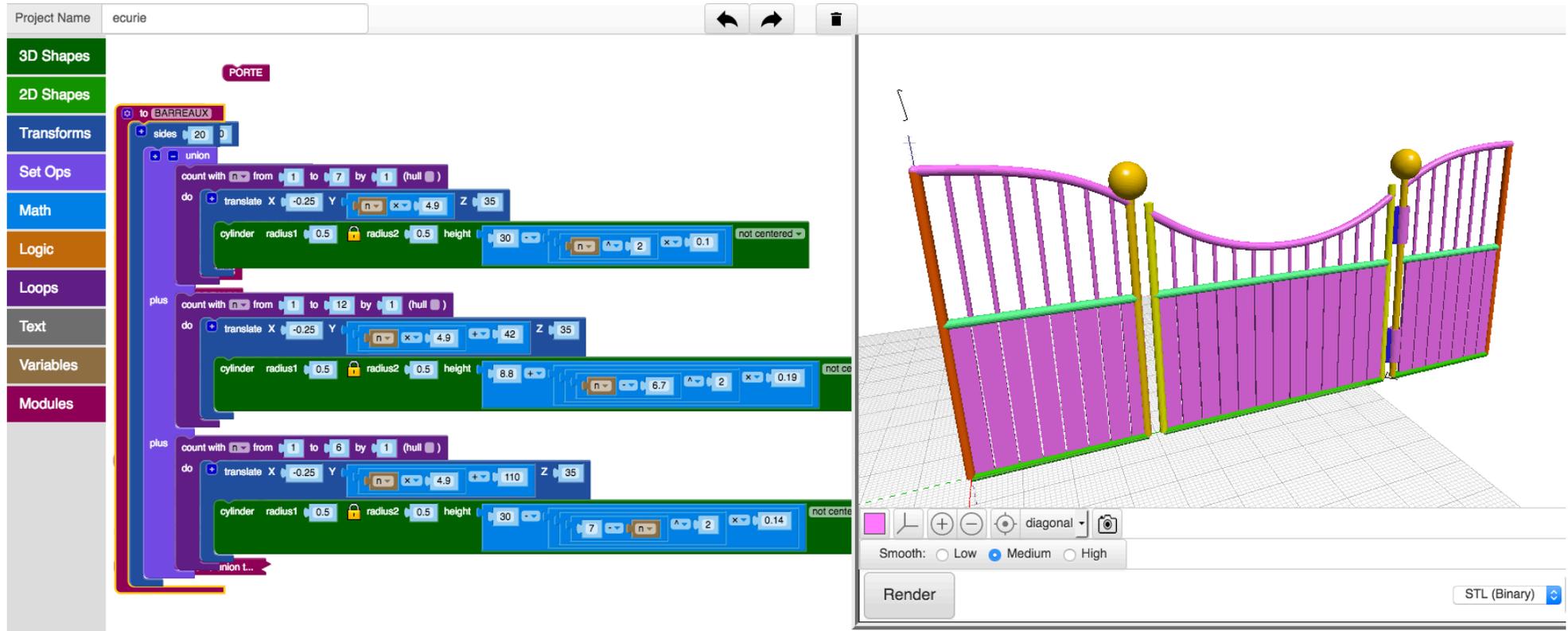
Rajoutons les gonds pour faire tourner la porte : trivial comme aurait dit mon professeur de mathématique du lycée Montaigne.

The screenshot displays a 3D modeling software interface with a code editor on the left and a 3D view on the right. The code editor shows two main blocks: 'to PORTE' and 'to GONDS'. The 'to PORTE' block includes a 'sides' block set to 20, followed by a 'union' block containing several sub-blocks: 'color #f0 count with k fro...', 'plus color #f3 count with k fro...', 'plus 3D Beziers5 - difference union...', 'plus GONDS', 'plus PLANCHES', and 'plus BARREAUX'. Below these are 'translate X' and 'rotate X' blocks. The 'to GONDS' block starts with a 'count with' block from 5 to 40 by 35, followed by a 'do' loop containing 'translate X' blocks, 'difference' and 'minus' blocks with cylinder parameters, and 'color' and 'cylinder' blocks. The 3D view shows a rendered model of a door with a pink frame, yellow hinges, and a purple handle, set against a grid background. The interface includes a 'Render' button and an 'STL (Binary)' export option.

Au tour des planches de faire leur apparition. Pas de souci, une simple révision des LOOP.

The screenshot displays the BlocksCAD software interface. The top menu bar includes 'Project', 'Options', 'Help', and 'Examples'. The 'Project Name' field is set to 'ecurie'. The left sidebar contains a vertical menu with categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace is divided into two panels. The left panel shows a tree view of the model's components: 'PORTE' (containing 'sides 20', 'union', 'color #f00 count with k fro...', 'plus color #ff3 count with k fro...', 'plus 3D Bezers5 - difference union...', 'plus GONDS', 'plus PLANCHES', 'plus BARREAUX', 'translate X 2 Y 28 Z 35', 'rotate X 270° Y 0° Z 0°', and 'LOQUET') and 'PLANCHES' (containing 'union', 'count with n from 0 to 7 by 1 (hull)', 'do translate X -0.5 Y [n] x -4.9 Z 0', 'cube X 1 Y 4.6 Z 35 not centered', 'plus count with n from 0 to 12 by 1 (hull)', 'do translate X -0.5 Y [n] x -4.9 [] 42 Z 0', 'cube X 1 Y 4.6 Z 35 not centered', 'plus count with n from 0 to 6 by 1 (hull)', 'do translate X -0.5 Y [n] x -4.9 [] 112 Z 0', 'cube X 1 Y 4.6 Z 35 not centered'). The right panel shows a 3D rendering of a stable door with a pink frame, purple panels, and yellow posts. The bottom right corner features a 'Render' button and a 'STL (Binary)' dropdown menu.

Maintenant, nous allons passer aux barreaux.

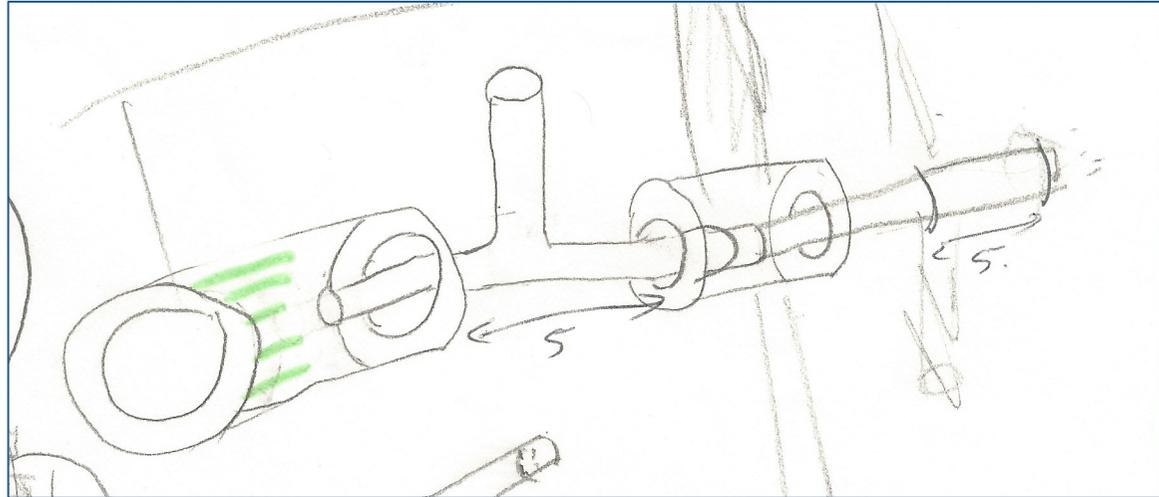


Les barreaux sont un peu plus délicats à disposer. On pourrait les faire les uns après les autres mais ce serait peu élégant. Pour chaque élément du portail, nous réalisons une boucle et la hauteur de chaque barre est une formule. L'idéal serait que cette formule soit de type Bézier mais le module Bézier n'est pas conçu pour cela. Pour l'instant ! Peut être un lecteur de ce livre fera un module avec la bonne formule ?

Nous avons préféré être rapide, et utiliser une formule simple de type $N \wedge 2$ (N puissance 2) avec quelques coefficients choisis par essais et re-essais. Pour expliquer, le coefficient est un nombre qui est soit multiplié soit additionné dans la formule.

Le Loquet

Pour finir, notre portail, il faut un petit loquet. Nous avons fait un concours : chacun devait imaginer un loquet. En combinant les idées de Xavière et les miennes, nous avons un vrai loquet. Mais comme celui de Quitrie était le plus efficace, elle a gagné. Voici son dessin.



Et une fois le block réalisé, le portail est fini.

Project Name ecurie

3D Shapes
2D Shapes
Transforms
Set Ops
Math
Logic
Loops
Text
Variables
Modules

PORTE

LOQUET

to PORTE

- sides 20
- union
 - color #f00 count with k fro...
 - plus color #ff3 count with k fro...
 - plus 3D Beziers5 - difference unic
 - plus GONDS
 - plus PLANCHES
 - plus BARREAUX
- translate X 2 Y 28
- rotate X 270° Y 0°
- LOQUET

to LOQUET

- sides 20
- union
 - translate X 0 Y 0 Z -7
 - difference
 - cylinder radius1 2 radius2 2 height 7 not centered
 - minus cylinder radius1 1 radius2 1 height 10 not centered
 - minus
 - translate X 0 Y 0 Z 7
 - rotate X 0° Y 90° Z 0°
 - cylinder radius1 1 radius2 1 height 5 not centered
- plus translate X 0 Y 0 Z 15
- color
- difference
 - cylinder radius1 2 radius2 2 height 5 not centered
- minus cylinder radius1 1 radius2 1 height 10 not centered
- minus
 - translate X 0 Y 0 Z 0
 - rotate X 0° Y 90° Z 0°
 - cylinder radius1 1 radius2 1 height 5 not centered

- plus color
- cylinder radius1 0.7 radius2 0.7 height 30 centered
- plus translate X 0 Y 0 Z 8
- rotate X 0° Y 90° Z 0°
- color
- cylinder radius1 1 radius2 1 height 5 not centered

diagonal

Smooth: Low Medium High

Render

STL (Binary)

Le toit de tuile

Le toit de tuile ne fait que répéter ce que nous savons déjà :

- ❑ Un module TUILE qui réalise une tuile
 - Avec un cylindre à peine conique
 - Une différence pour le rendre creux
 - Une différence pour le couper en deux
- ❑ Un module TOITURE avec 2 boucles :
 - Une pour mettre les tuiles en rangées
 - L'autre pour additionner les rangées

The screenshot displays a 3D modeling software interface. On the left, a sidebar lists various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace contains a block-based script for a project named 'ecurie'. The script is organized into several modules: 'ECURIE', 'TOITURE', and 'TUILES'. The 'TOITURE' module uses a 'do' loop to iterate over rows, with a 'count with' block and 'translate' blocks for positioning. The 'TUILES' module uses a 'do' loop to iterate over tiles in a row, with a 'scale' block, a 'difference' block (combining a cylinder and another cylinder to create a hollow shape), and a 'minus' block (combining the result with a cube to create the final tile shape). The 3D view on the right shows a purple tiled roof structure with a grid floor and a coordinate system. The interface includes navigation controls like 'diagonal' and a camera icon.

Le box enfin terminé

Voici, le grand jour est arrivé.

The screenshot displays the BlocksCAD software interface. On the left, a vertical sidebar lists various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace is divided into two panes. The left pane shows a hierarchical tree of blocks for a project named 'écurie'. The root block is 'to ECURIE', which contains a 'difference' block. This 'difference' block is composed of several 'plus' blocks and one 'minus' block. The 'plus' blocks include: a 'union' block containing 'STRUCTURE' and 'PORTE'; a 'color' block; a 'translate' block (X: 120, Y: 2, Z: 0); a 'rotate' block (X: 0°, Y: 0°, Z: 270°); a 'color' block; a 'MUR with:' block (longueur: 120, hauteur: 65); another 'translate' block (X: 0, Y: 148, Z: 0); another 'rotate' block (X: 0°, Y: 0°, Z: 270°); a 'color' block; another 'MUR with:' block (longueur: 120, hauteur: 65); another 'translate' block (X: -10, Y: -5, Z: 155.5); another 'rotate' block (X: 0°, Y: 65°, Z: 0°); and a 'TOITURE' block. The 'minus' block at the bottom is a 'translate' block (X: 130, Y: -5, Z: 0) applied to a 'cube' block (X: 40, Y: 170, Z: 70, not centered). The right pane shows a 3D rendered view of the stable box, featuring a brick wall, a green roof, and a green fence. Below the 3D view is a control panel with a 'Render' button, a 'Smooth' dropdown (set to Medium), and a 'Generate STL' button. The interface also includes a top menu bar with 'Project', 'Options', 'Help', and 'Examples', and a 'Get Started!' button. The bottom right corner of the 3D view has zoom controls (+, -).

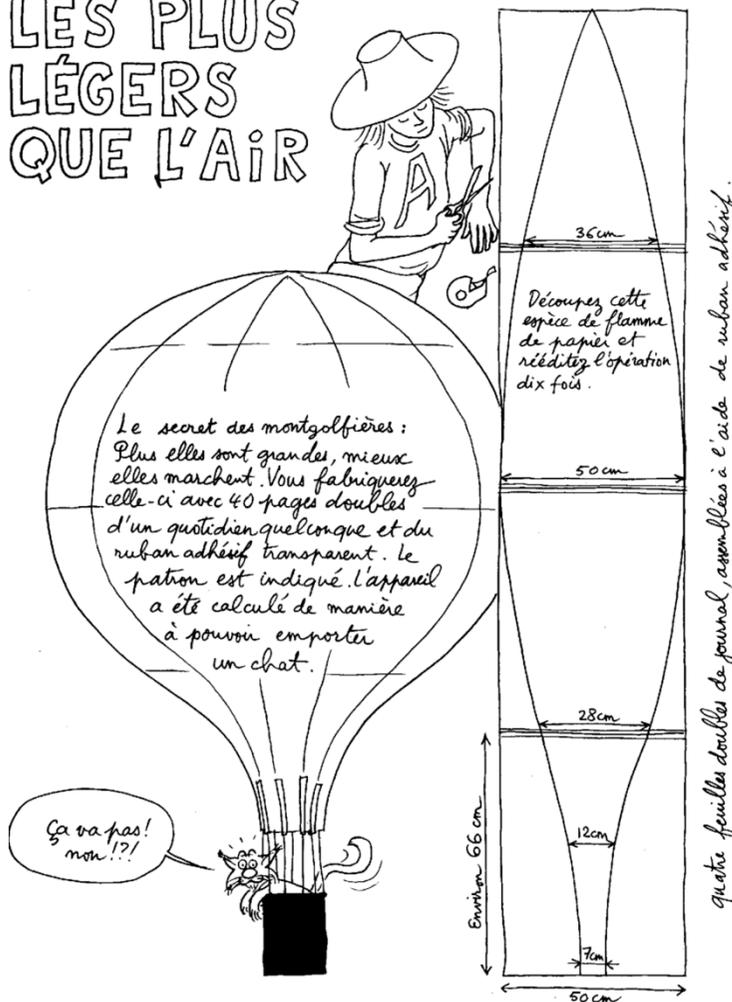
Nous pourrions encore le compléter avec des accessoires, des fenêtres à l'arrière et plein d'autres choses. Et pour être plus près de la photo prise sur Internet, il suffit de faire une boucle qui dessine et translate l'écurie. Prévoir d'aller goûter une fois RENDER lancé !

La montgolfière solaire

Le projet fou de Guillaume d'Echateau

Vous ne le connaissez sûrement pas mais Guillaume est un cousin éloigné d'Anselme Lanturlu. Les coordonnées d'Anselme sont en fin de document et je vous conseille de lire ses savantures (oui, Anselme est un savanturier) qui sont passionnantes.

LES PLUS
LÉGERS
QUE L'AIR



quatre feuilles doubles de journal, assemblées à l'aide de ruban adhésif.

Revenons à Guillaume. Garçon sérieux au demeurant, il est néanmoins passionné de Montgolfière. Objets volants, certes intéressants, mais hélas assez peu dans le développement durable car il faut un bruleur à gaz. Pour ceux qui ne connaissent pas les montgolfière, le bruleur à gaz (à gaz car c'est le plus simple dans ce cas) sert à chauffer l'air à l'intérieur du ballon vers 100 °Celsius. La différence de poids entre l'air très chaud du ballon et l'air ambiant permet au ballon de voler selon un principe attribué à Archimède.

Guillaume veut donc faire une montgolfière solaire dans laquelle le bruleur sera remplacé par une espèce de four solaire.

Fou mais pas inconscient, Guillaume nous a demandé de l'aider pour faire une maquette de son bolide.

La montgolfière

Pour la montgolfière, reprenons les plans donnés par Anselme dans l'Aspirisoufle.

Le dessin ci-contre est assez clair. Il faut piquer les journaux de la semaine dernière au grand-père et le tour est joué.

Pour un premier essai, nous avons privilégié le pistolet à colle mais les joints sont ensuite un peu durs. Rempli d'air chaud avec un sèche cheveux, la montgolfière a fait quelques mètres avant que le vent (il y en avait) ne la déchire.

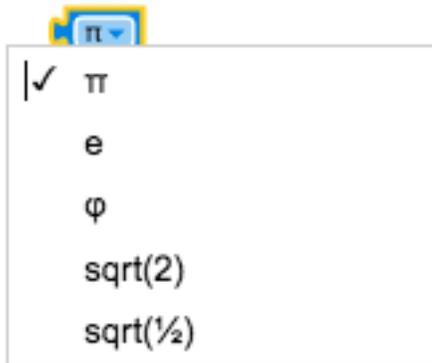
Mais que vient faire PI dans cette histoire ?

Π , lettre en alphabet grec, qui se prononce « PI » est un nombre un peu magique pour les mathématiciens. Il sert à calculer le diamètre d'un cercle quand on connaît le périmètre, ou le contraire.

Périmètre du cercle = Π x diamètre du cercle.

Quand on ne possède pas d'ordinateur, il suffit de se rappeler que PI vaut environ 3,14. Je dis vaut environ car PI ne se termine jamais mais cela ne nous intéresse que peu pour nos bricolages.

Et quand on possède un ordinateur, PI est en général donné. Dans BlocksCAD, menu gauche MATH, vous pouvez y accéder par un block qui permet de choisir π , mais aussi « e », « φ », $\sqrt{2}$, $\sqrt{1/2}$. L'espèce de « V » bizarre se dit « racine carré » en français et SquareRoot en anglais.



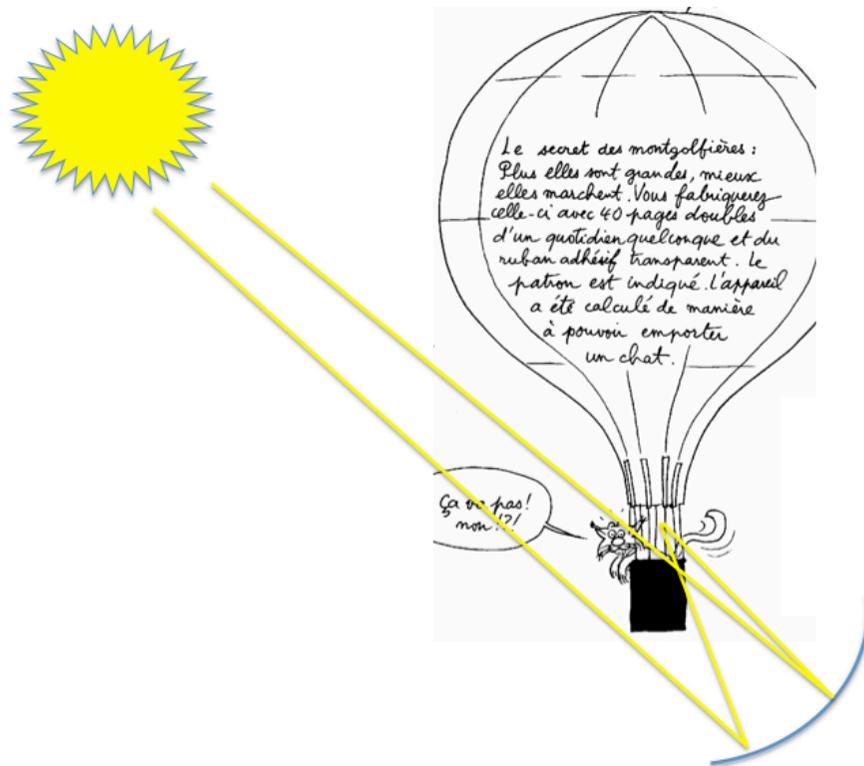
Revenons à notre montgolfière. Si nous avons 10 grandes flammes à assembler, le périmètre du cercle tout en bas sera de $10 \times 7 \text{ cm} = 70 \text{ cm}$. Division par PI pour obtenir le diamètre : environ 22 cm.

Utilisons encore PI. La partie la plus larde de la flamme est de 50 cm. On peut approximer le périmètre de la sphère à $50 \times 10 = 500 \text{ cm}$, soit 5 mètres. Son rayon est donc de ... de 80 cm (bien sûr petit piège, le diamètre est de 159 cm donc le rayon la moitié).

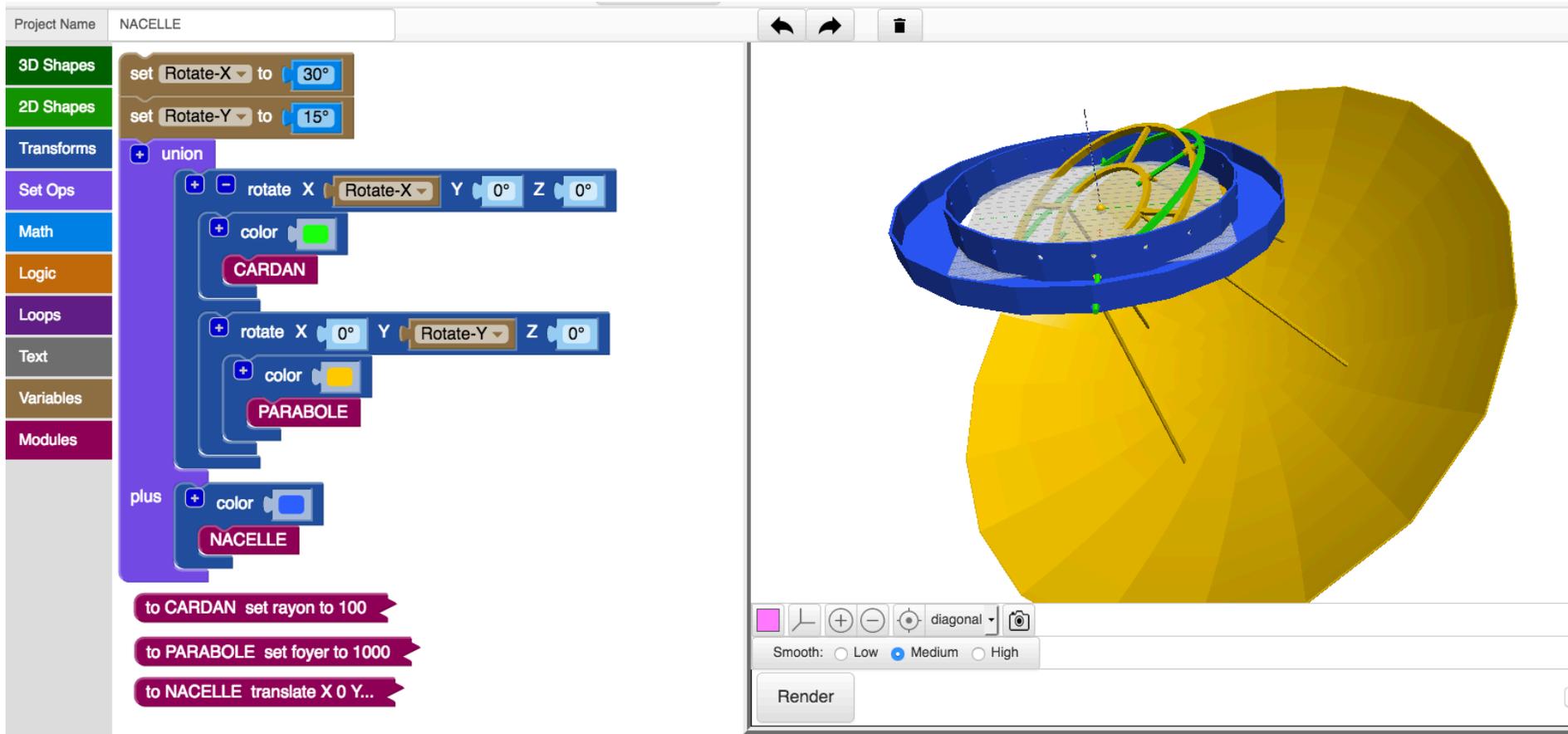
A quoi sert d'avoir ce rayon. Pas grand chose hormis de connaître le volume de cette montgolfière. Autre formule autant magique qu'inutile à connaître par cœur. Le volume de la sphère est égale à $(4/3) \times \pi \times \text{RAYON} \wedge 3$ (c'est à dire rayon au cube). Soit avec 80 cm, cela donne $2\,143\,573 \text{ cm}^3$. Sachant qu'un litre vaut 1000 cm^3 , cela nous fait 2 143 litres et sachant qu'un m^3 fait 1000 litres, notre montgolfière fait plus de 2 m^3 .

Le bruleur solaire

Le principe est de remplacer le bruleur à gaz par une parabole concentrant la lumière du soleil pour faire chauffer un bruleur. Mon pauvre chat, Guillaume ne vaut guère mieux qu'Anselme !



Voici le résultat que nous vous proposons.



le mécanisme est composé de trois éléments :

- La parabole avec son support, en jaune
- Un cardan, en vert, en liaison avec le support de la parabole et la nacelle
- La nacelle, en bleu, qui doit supporter la toile de la montgolfière.

La nacelle de la montgolfière

Elle est assez simple, ressemblant un peu à la nacelle des frères Mongolfier ou, plus actuel, à celle de la charlière du Parc André Citroën à Paris.

Elle est réalisée avec une forme en 2D et un ROTATE EXTRUDE.

The screenshot displays a 3D modeling software interface for creating a hot air balloon basket (nacelle). The interface is divided into several sections:

- Project Name:** NACELLE
- Left Sidebar:** A vertical menu with categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules.
- Main Workspace:** A complex assembly tree for the 'NACELLE' object. The tree starts with a 'translate' block (X: 0, Y: 0, Z: -4). This is followed by a 'difference' block that combines a 'rotate extrude' (sides: 20) of a square (X: 2, Y: 30, not centered) with other squares (X: 40, Y: 20, not centered) and a 'translate' block (X: 2, Y: 2). A 'do' loop with 'count with' (from 1 to 20 by 1) contains a 'rotate' block (X: 0°, Y: 0°, Z: 18°), a 'translate' block (X: 0, Y: 0, Z: 20), a 'rotate' block (X: 0°, Y: 90°, Z: 0°), and a 'cylinder' block (radius1: 2, radius2: 2, height: 120, not centered). The final 'minus' block adds a base using 'translate' (X: -99, Y: 0, Z: 4), 'rotate' (X: 0°, Y: 270°, Z: 0°), and 'cylinder' (radius1: 3, radius2: 3, height: 60, not centered) blocks.
- Right Panel:** A 3D view of the completed blue basket on a grid. The basket has a circular base and a rim with several holes. Below the view are controls for 'Smooth' (Low, Medium, High) and a 'Generate STL' button.

Nous avons rajouté quelques trous pour attacher la toile de la montgolfière. Les côtes de la nacelle sont déduites des plans d'Anselme.

La parabole

La parabole est faite suivant la formule déjà donnée pour une parabole de révolution. Ensuite, le plus long est de faire le support en plusieurs parties : des jambes pour accrocher la parabole, et deux anneaux pour accueillir le joint de cardan.

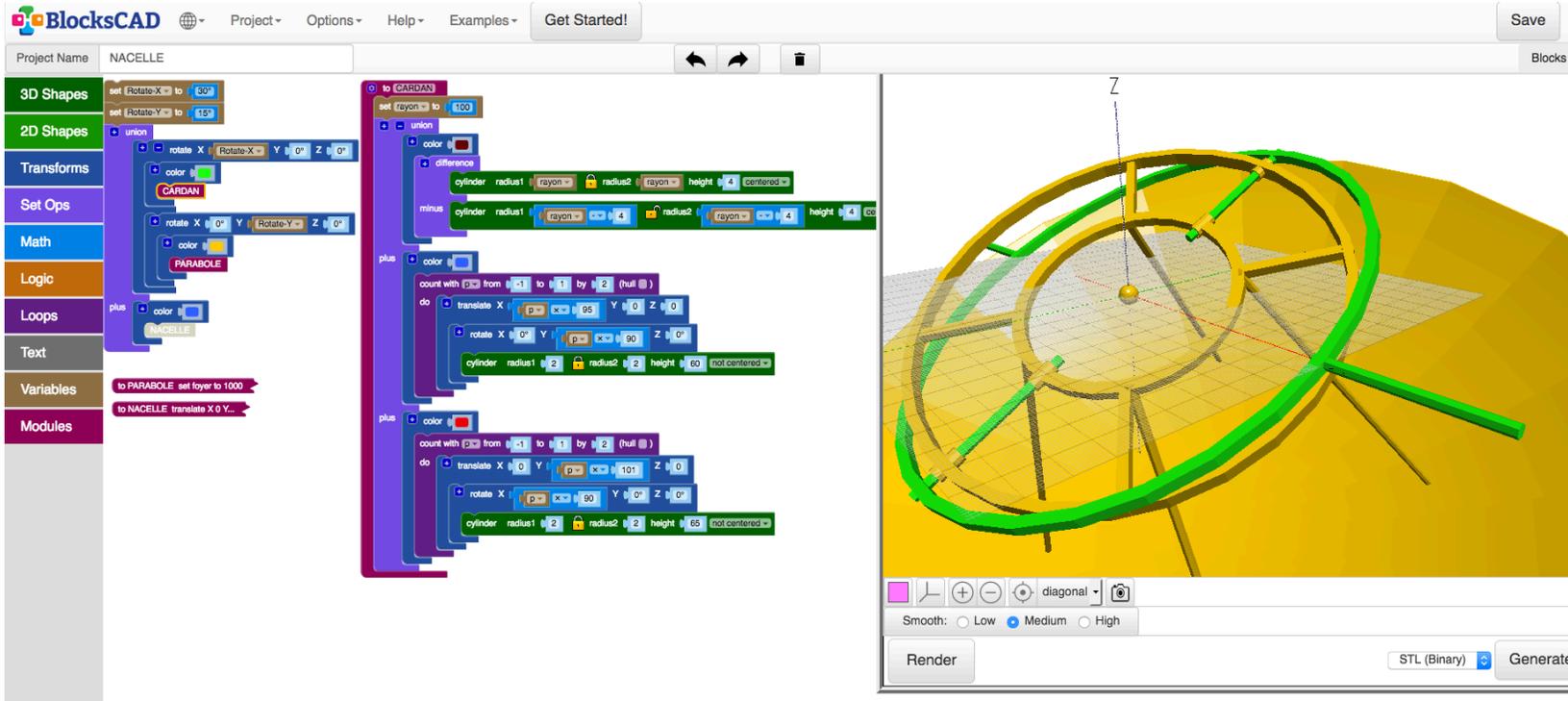
The screenshot displays the BlocksCAD software interface. On the left, a sidebar lists various tool categories: 3D Shapes, 2D Shapes, Transforms, Set Ops, Math, Logic, Loops, Text, Variables, and Modules. The main workspace is filled with a complex assembly of colored blocks representing the CAD model's construction. The blocks include operations like 'union', 'difference', 'minus', 'translate X', 'rotate X', 'rotate Y', 'rotate Z', 'cylinder', 'square', 'circle', 'count with', and 'do'. The 3D view on the right shows a yellow parabolic dish with a grey support structure consisting of two rings and four legs. A small sphere is visible at the origin (0,0) of the model. The interface includes a top menu bar with 'Project', 'Options', 'Help', and 'Examples', and a 'Get Started!' button. On the right side of the interface, there are 'Save' and 'CYV' buttons, and a 'Render' button with a 'Smooth' dropdown set to 'Medium'. At the bottom right, there are 'STL (Binary)' and 'Generate STL' buttons.

Tout le programme n'est pas visible sur l'image (un peu long). La boule au milieu est là uniquement pour montrer le foyer de la parabole mais ne servira pas à la fabrication. Ce foyer est positionné à l'origine (0,0) du modèle.

Le joint de Cardan

Pour tout savoir sur le joint de Cardan, il va falloir soit voir sur Wikipédia soit demander à votre professeur de Technologies. Il est utilisé ici pour orienter la parabole tout en gardant le foyer sur l'origine. Ce joint est composé d'un anneau en vert :

- ❑ Dont l'axe des X s'insère dans la nacelle (les 2 barres qui vont vers l'extérieur)
- ❑ Dont l'axe des Y s'insère dans le support de la parabole (les 2 axes qui vont vers l'intérieur).



Il suffit pour tourner la parabole des 2 variables (Rotate-Y) et (Rotate-X) ; regardez bien le bloc :

- ❑ (Rotate-Y) fait tourner suivant l'axe Y toute la parabole avec son support
- ❑ (Rotate-X) fait tourner le cardan vert ET AUSSI la parabole.

C'es tassez primaire, mais cela

éclaire bien (du moins je crois) le fonctionnement d'un cardan. Celui-ci est présent un peu partout dans les objets mécaniques, dans tous les jeux de construction et, très important, dans les voiliers pour que le réchaud à gaz reste horizontale même quand le bateau gite et roule.

Il n'est pas sûr que ce projet soit l'invention du siècle pour les aérostiers. Il nous est néanmoins parut intéressant pour fabriquer ce joint de Cardan et parler un peu de PI.

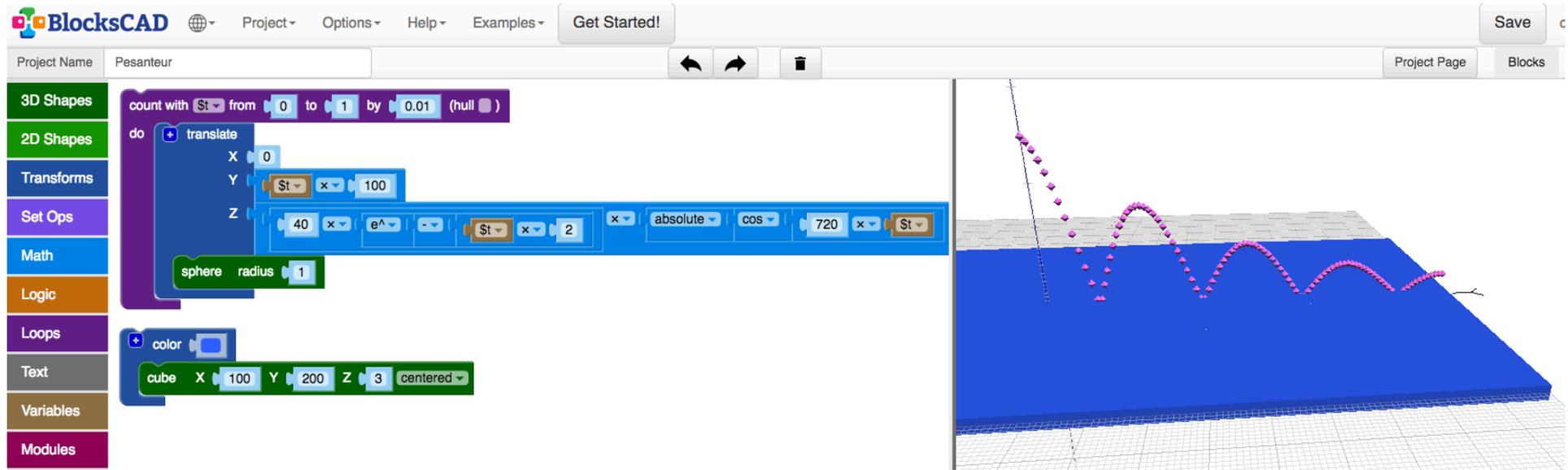
Faisons notre cinéma

Avant de nous quitter, laissons entrevoir les possibilités de l'animation, c'est à dire de faire bouger les objets dans le temps et pas uniquement dans l'espace.

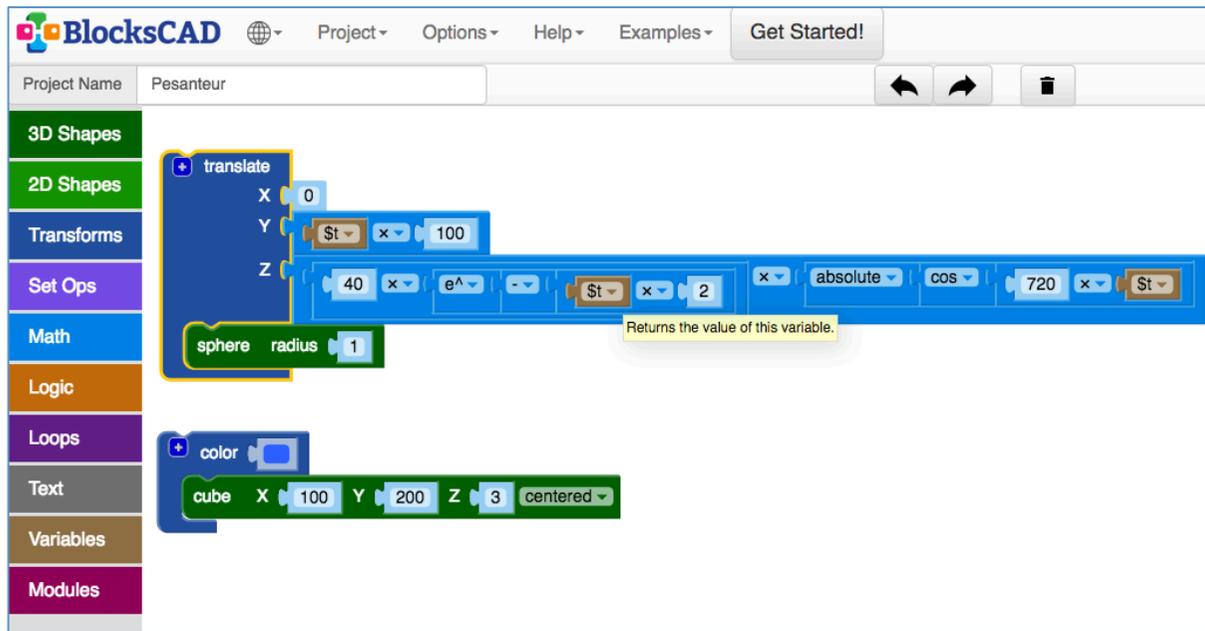
Une courbe rebondissant

Premièrement, réalisons une courbe qui ressemble à celle de l'exponentielle amortie que nous avons appelée « courbe d'amortissement ». Nous avons apporté néanmoins 2 modifications :

- Nous avons changé la variable utilisée « i » en « \$t », nous verrons plus tard pourquoi.
- Nous avons changé le sinus en cosinus et rajouté la fonction la fonction ABSOLUTE devant. Cette fonction rend le résultat du cosinus toujours positif
- Enfin, nous faisons varier la variable « \$t » entre 0 et 1, ce qui a obligé à changer les paramètres pour faire une jolie courbe que vous pouvez voir ci-dessous.



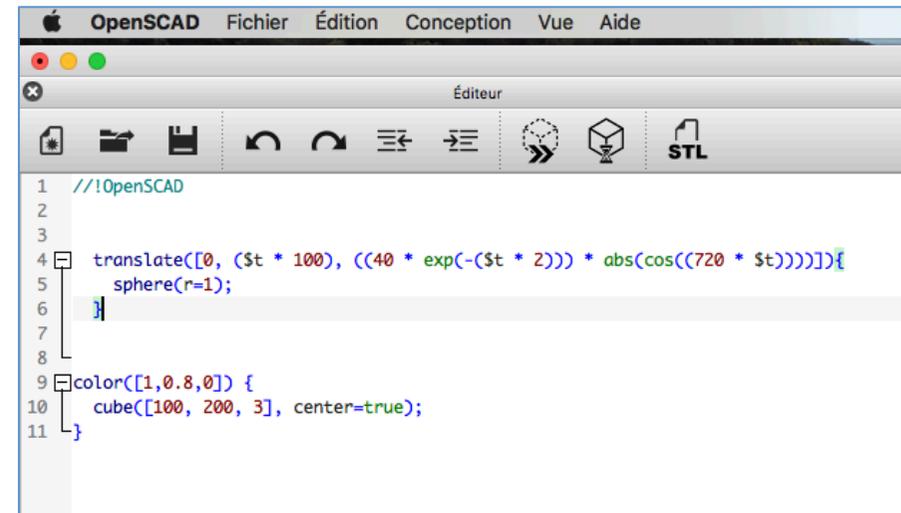
Préparation de la courbe pour passer en Openscad



Maintenant, nous allons supprimer la boucle sans pour autant toucher au reste.

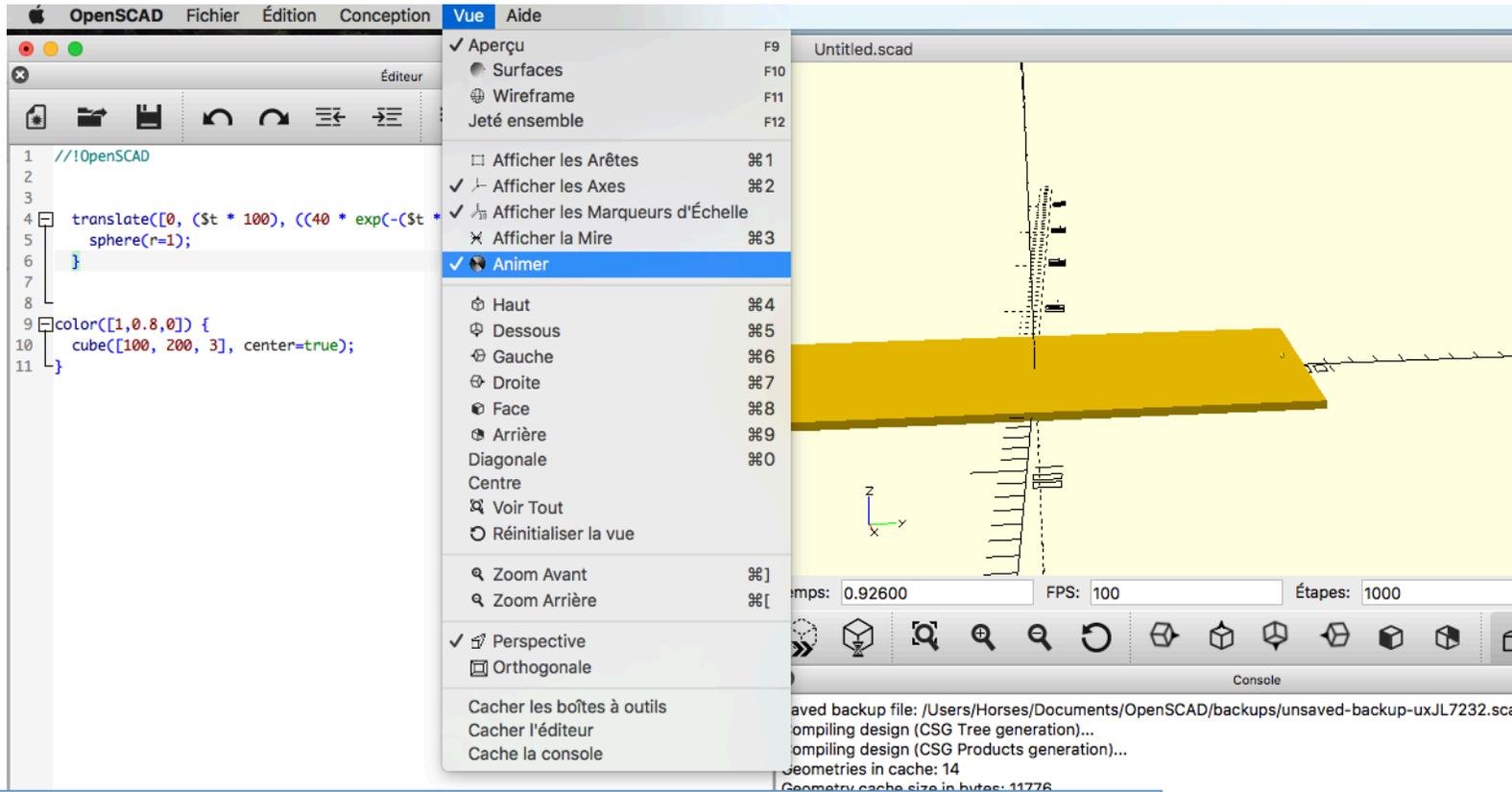
Maintenant, il ne reste qu'à aller sous l'onglet CODE et copier le code pour le passer dans Openscad.

Sur l'image ci-contre, nous avons donc le code en format Openscad dans ce même logiciel. Il arrive que le symbole « \$ » devant le « t » ne passe pas dans le copier-coller. Il suffit alors de le remettre à la main.



L'animation avec Openscad

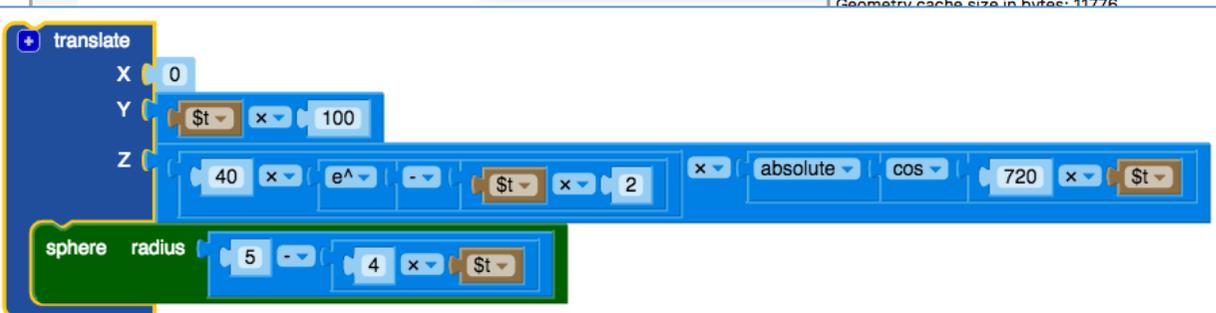
Dans Openscad, la variable « \$t » est une variable spéciale qui réagit avec la fonction ANIMER que l'on active à partir du menu VUE.



Cette fonction ANIMER fait apparaître 3 zones dont deux pour la saisie :

- ❑ FPS, qui doit vouloir dire Frame Per Second et qui va permettre de dire au logiciel combien d'images il doit réaliser en une seconde.

- ❑ Étapes, qui va permettre de régler la vitesse de la sphère qui bouge, c'est à dire le BY de la LOOP dans BlocksCAD.



En complément un petit block où la sphère change de taille en même temps qu'elle tombe.

Autres sources de jeux

Dans ce chapitre, nous avons mis les liens pour découvrir de nouvelles choses amusantes, intéressantes et éducatives.

BLOCKLY

Blockly est l'interface réalisée par GOOGLE qui est utilisé dans BlocksCAD. Blockly peut aussi être utilisé pour programmer des Arduino (petit contrôleur pour robot) ou des Thymio (petits robots sympathiques). <https://developers.google.com/blockly/>

SNAP! Build You Own Blocks

SNAP est un logiciel de programmation généraliste et récursif (parfait pour les fractals) qui utilise le même style d'interface de blocks. <http://snap.berkeley.edu/index.html>

SCRATCH

Initiateur de cette démarche très graphique de programmation, SCRATCH est principalement destiné aux enfants. Il est néanmoins très puissant. <https://scratch.mit.edu/>

OPENSCAD

OPENSCAD est un modèleur qui a servi de référence à BlocksCAD. <http://www.openscad.org/>

Mathématique magique

Le site personnel d'une agrégée de Mathématiques, site plein de ressources, d'histoires et de merveilles. <http://therese.eveilleau.pagesperso-orange.fr/>

Maths et Tiques

<http://www.maths-et-tiques.fr/> un site avec de nombreuses ressources sur les mathématiques et la géométrie.

Anselme Lanturlu

https://www.jp-petit.org/TELECHARGEABLES/livres_telechargeables.htm De nombreux aspects de la science abordés de façon originale voire révolutionnaire dans des bandes dessinées qui ont fait rêver mon adolescence.

HIPARQUE

Pour aller (beaucoup) plus loin dans la notion de représentation en trois dimensions un excellent document http://www.dimensions-math.org/Dim_regarder.htm

MATHCURVE

<http://www.mathcurve.com/>
De nombreuses ressources et formules

Pour ceux qui aiment l'air

Le brevet d'initiation à l'aéronautique est un examen pour les jeunes qui veulent faire du vol libre et plus tard du vol moteur. Passionné d'avion dans mon adolescence, j'ai travaillé cet examen (sans jamais le passer finalement) et cela m'a donné beaucoup de culture générale sur la physique, le vol des avions, les mathématiques, la navigation, la météorologie. Ce qui m'a beaucoup servi dans mes activités nautiques plus tard.

On ne peut que remercier ceux qui avaient écrit le cours de préparation (il y a bien longtemps) et surtout que ce cours soit disponible en téléchargement gratuitement aujourd'hui. <http://federation.ffvl.fr/pages/brevet-dinitiation-aeronautique-bia>.

Pour ceux qui aiment l'eau

N'en ayant pas l'air, j'ai ensuite préféré l'eau, mais il n'existe pas d'équivalent au BIA pour les marins. C'est un peu dommage, mais comme un avion et un bateau marchent sensiblement de la même façon à « ROTATE 90° » près, vous pouvez toujours regarder le site de ceux qui aiment l'air.

A bientôt

Ce voyage au pays des nombres se termine. Ces nombres sont là pour compter mais aussi pour faire. En espérant que vous allez prendre autant de plaisir et d'acharnement à faire et à réaliser vos objets que nous en avons eu pour écrire le livre et les blocks.

Sommaire

Bienvenue.....	2
Premiers contacts.....	3
Le bureau de travail.....	3
Une boule avec SPHERE.....	4
La représentation graphique.....	5
La fonction CUBE.....	6
L'option CENTERED.....	7
La fonction TRANSLATE.....	8
Un peu de COLOR.....	9
Un triangle rectangle avec l'option ROTATE.....	10
ROTATE et TRANSLATE.....	11
La mesure des angles : degré, radian ou grade.....	12
Le Tire-bouchon de Maxwell.....	13
Cylindres et cônes.....	14
Les tores avec TORUS.....	15
Les options SIDES et FACES.....	16
La réunion de différentes formes avec UNION.....	17
La différence de différentes formes avec DIFFERENCE.....	18
Bien gérer la DIFFERENCE.....	19



Un machin planétaire	20
L'intersection de formes avec INTERSECTION	21
HULL	22
Une roue (simple) de charrette	23
LOOP, une boucle informatique	28
Le module « faire une roue »	32
Un champignon	34
Un peu d'organisation	35
Programmez en COLOR	36
Commentez vos programmes	36
Le tire bouchon de Maxwell	37
Réalisons un profil pour une extrusion	38
La fonction EXTRUDE sur un profil 2D	39
L'option TWIST de la fonction EXTRUDE	40
Notre première formule mathématique	41
L'option SCALE de la fonction EXTRUDE	42
Un manche conique avec FANCY MIRROR	43
FANCY MIRROR ou MIRROR	44
La poignée du tire-bouchon	45
Une flèche positive avec 3D TEXT	46
Le programme complet du tire-bouchon de Maxwell	48
Dégustons une glace cassis-vanille	49
Retour sur la fonction MIRROR	50
Un Baril pour Barrel Race	51
Profils pour les fonctions LINEAR EXTRUDE et ROTATE EXTRUDE	52
ROTATE EXTRUDE	52
L'option SIDE de ROTATE-EXTRUDE	56
Les profils	58
Courbes et comment les dessiner pour en faire des profils	59
Premiers pas	59
Les fonctions mathématiques	61
L'équation paramétrique	62
Une courbe lissée avec HULL	63
Ecrire des formules avec BlocksCAD	64
Formule de la parabole avec les « puissances »	65



Une antenne Satellite	66
Un four solaire à hot dog	67
Une autre formule pour réaliser une hyperbole.....	68
Mais à quoi servent les hyperboles	70
Les coordonnées Polaires	71
Les quadrillages	72
Créer un point en coordonnées polaires	73
Le module simple pour créer un point en coordonnées polaires	74
Créer un point en coordonnées polaires uniquement avec TRANSLATE	75
Le block Point Polaire mathématique	77
La fonction IF (« si » en anglais).....	78
Une courbe en coordonnées polaires	79
Le cercle polaire.....	80
L'arc de cercle	81
La part de tarte	82
Une rosace (2D Rosace).....	83
Les paramètres de la Rosace.....	84
La Spirale d'Archimède.....	85
Les courbes périodiques	86
Une onde sinusoïdale	86
Amplitude et période	87
Un peu d'osier	88
Restons en phase	89
Mieux gérer les extrémités avec la phase.....	90
REMAINDER	91
HSV ou TSV	92
Un cercle ondulé	93
Une formule d'ellipse	95
Les coniques : Ellipse, hyperbole et parabole	96
Le cercle est une conique	97
L'ellipse est une conique	98
La parabole est une conique	99
l'hyperbole est une conique	101
L'hyperbole des mathématiciens	103
Les équations des différentes coniques	104



Encore plus de courbes : Pierre Bézier et Cyril Grandpierre	105
Les courbes de Bézièrs.....	105
Le block 2D Bezièrs3	106
Le Bloc 2D Bezièrs4.....	108
Le Bloc 2D Bezièrs5.....	109
Un profil d'aile	110
Le triangle de Pascal.....	111
La méthode C. Grandpierre	112
BlocksCAD versus Openscad	113
Un sulky pour Duchesse des Myrtilles.....	118
Le siège du sulky	119
Les roues.....	120
Essieux.....	121
Les pédales.....	122
Les barres de harnais	123
Harnais et boucles	124
Un peu de géométrie.....	126
Le module REGLE	126
Le module REGLE GRADUEE : le théorème de Pythagore.....	127
Racine carré ou Square Root.....	128
2 chiffres après la virgule	129
Le module COMPAS	130
Une première rosace	131
Le module ROSACE	133
Les fonctions mathématiques de BlocksCAD	135
Racines carrées et valeur absolue	136
Exponentiel.....	137
Exponentiel décroissante	138
Une courbe d'amortissement	139
Logarithme et exponentiel	140
Logarithme.....	141
Prenons la tangente	142
Encore une tangente !.....	148
Combinons les fonctions	148
Un coque de bateau avec SCALE.....	150



La formule mathématique de l'accolade.....	151
Le Maître-Couple.....	152
Le liston et la quille.....	153
Les couples.....	154
La coque complète.....	157
Le sapin de Noël	158
Un feuillage de tores.....	159
La spirale d'Archimède pour guirlande	160
Les Boules de Noël réparties sur la guirlande	162
Les courbes en trois dimensions	163
Passons dans la troisième dimension : les coordonnées cartésiennes.....	163
Les coordonnées cylindriques	164
Des Coordonnées cylindriques aux coordonnées cartésiennes	165
Les coordonnées sphériques	166
Z en coordonnées sphériques.....	167
Le cosinus de la latitude	168
Un block pour un point en coordonnées sphériques	171
L'ombre du point en coordonnées sphérique.....	172
Les méridiens en coordonnées sphérique	173
Les parallèles en coordonnées sphériques.....	174
Une droite en coordonnées sphériques.....	175
Les fonctions sphériques	176
Le passage des courbes 2D en 3D.....	177
Une tresse sinusoïdale.....	177
Les courbes de Bézier en 3D	178
Les Block 3D Bezier4.....	179
3D Bézier 5.....	180
Les surfaces en 3D	181
Une parabololoïde hyperbolique.....	181
Le block de selle de cheval.....	182
La fonction U^2-V^2 ou $U \wedge 2-V \wedge 2$	183
Et bien jouons à Star Wars dit la cigale !.....	185
Jeux de construction	186
Le pied à coulisse.....	186



Lego®, Meccano® et tous les autres	187
Une barre aux dimensions Lego Technic®	188
Un projet plus complet : l'écurie.....	189
Démarrer un projet.....	189
Premier pas : la structure	190
Les briques.....	191
Le mur de briques.....	201
REMAINDER OF.....	203
Le module MUR et ROUND	204
Le portail et les montants	208
Le Loquet	213
Le toit de tuile	215
Le box enfin terminé.....	216
La montgolfière solaire.....	217
Le projet fou de Guillaume d'Echateau.....	217
La montgolfière	217
Mais que vient faire PI dans cette histoire ?.....	218
Le bruleur solaire	219
La nacelle de la montgolfière.....	221
La parabole	222
Le joint de Cardan.....	223
Faisons notre cinéma	224
Une courbe rebondissant.....	224
Préparation de la courbe pour passer en Openscad.....	225
L'animation avec Openscad.....	226
Autres sources de jeux	227
BLOCKY	227
SNAP! Build You Own Blocks	227
SCRATCH.....	227
OPENSCAD	227
Mathématique magique.....	227
Maths et Tiques	228
Anselme Lanturlu	228
HIPARQUE	228
MATHCURVE.....	228



Pour ceux qui aiment l'air..... 228

Pour ceux qui aiment l'eau 229

A bientôt..... 229

Remerciements 236

Remerciements

Special thanks to Jennie Yoder, BlocksCAD developer, for so many good advices about this book.

Remerciements à Isabelle de J, Florence F, Manfred O, Clément E, Nicolas L, Marcel O, Bernard M, pour leurs supports à mes sollicitations et leurs encouragements, petits et grands. Merci aussi à Caroline, Jeanne et Guillaume de ViletteMakerz pour l'anticipation de la suite ...

Merci aussi à Pierre Bézier et Cyril Grandpierre pour leurs courbes et à Jean-Pierre Petit, le papa du premier des savanturiers, notre guide à tous : Anselme Lanturlu.



Ce livre n'aurait pas vu le jour sans la passion de Quitrie pour les grandes constructions en carton et son exploration avec moi de BlockcCAD, vite accompagnée de sa sœur Xavière qui nous propose aussi ce dessin de fin de livre.