****

**UNIVERSITE DE CARTHAGE**

**FACULTE DES SCIENCES DE BIZERTE**

**Département physique.**

PROJET

**Electronique, Electrotechnique et Automatique**

Créer une jeux de PING--PONG

**Présenté par :**

**Khalil Chahid**

**Année universitaire : 2022 / 2023**

**Table des matières**

[Introduction 1](#_Toc92727195)

[Chapitre 1 : Cadre de projet 2](#_Toc92727196)

[1.Le Ping Pong : 2](#_Toc92727197)

[2.Cadre général de projet : 3](#_Toc92727201)

[Chapitre 2 : Création du jeu 4](#_Toc92727202)

[1.JAVA Processing : 4](#_Toc92727203)

[2.P5.JS : 5](#_Toc92727204)

[3.Création du code : 6](#_Toc92727205)

[3.1. Création du Background : 6](#_Toc92727206)

[3.2. Création du Main : 8](#_Toc92727207)

[3.3. Les joueurs : 10](#_Toc92727208)

[3.4. La balle : 11](#_Toc92727209)

[Conclusion 16](#_Toc92727212)

[Références bibliographiques 16](#_Toc92727213)

# Introduction

Dans notre monde moderne, les logiciels numériques est devenu un pilier essentiel de la façon dont les choses fonctionnent. Les langages de programmation sont devenus les mêmes que les laquages parlés au quotidien.

Ces types de langues varient, mais d’autre part, ils sont tous impliqué dans la création de contenu numérique, talques la science-fiction, l’animation, les effets d’art, les jeux vidéo et des nouvelles technologies qui inaugurent une nouvelle ère de progrès technologiques.

# Chapitre 1 : Cadre de projet

# Le Ping Pong :

## Le tennis de table, également appelé Ping Pong, jeu de balle similaire en principe au tennis et joué sur une table plate divisée en deux terrains égaux par un filet fixé sur toute sa largeur au milieu.



Figure 1 : Une image du jeux paralympiques Tokyo 2021

## L’objectif est de frapper la balle de sorte qu’elle passe par-dessus le filet et rebondit sur la moitié de la table de l’adversaire de telle sorte que l’adversaire ne peut pas l’atteindre ou le retourner correctement. La balle creuse légère est propulsée d’avant en arrière à travers le filet par de petites raquettes (paddles) tenues par les joueurs.

## Le jeu est populaire dans le monde entier. Dans la plupart des pays, il est très bien organisé en tant que sport de compétition, en particulier en Europe et en Asie, en particulier en Chine et au Japon.

# Cadre général de projet :

Mon objectif est de développer une jeux simple similaire au **Ping Pong** (bien sûr on n’examine pas tous les règles de ce jeu), en utilisant le langage de programmation **« Processing »**.

**C’est quoi ce langage ?** On le connaîtra plus tard.

# Chapitre 2 : Création du jeu

## JAVA Processing :

Le **Processing** est un logiciel flexible et un langage pour apprendre à coder dans le contexte des arts visuels. Depuis 2001, Processing fait la promotion de la connaissance des logiciels dans les arts visuels et de la connaissance visuelle dans la technologie. Des dizaines de milliers d’étudiants, d’artistes, de designers, de chercheurs et d’amateurs utilisent **Processing** pour l’apprentissage et le prototypage.

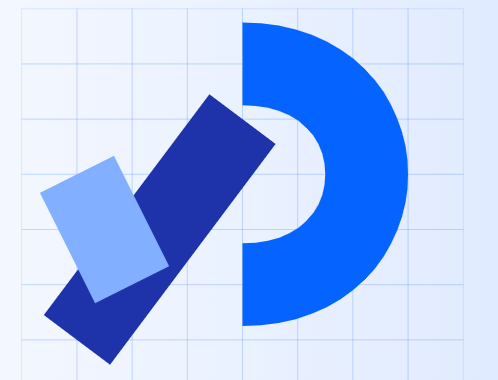


Figure 2 : Le « Processing »

Le principe majeur de **Processing** est de simplifier au maximum la mise en œuvre des programmes. Le langage **Processing**, bien que très proche de langages tels que **Java** ou **C++**, se veut lui aussi assez simple.

Adapté à la création graphique, **Processing** réclame moins d'efforts que **Java** pour effectuer des tâches simples telles que la modification d'une animation à intervalle régulier (qui permet des créations animées).

Ses fonctionnalités sont limitées aux besoins des créateurs d'images 2D et 3D générées par programmation mais peuvent être étendues, par le biais de modules externes, à la capture d'un flux vidéo, à la génération et à la manipulation de son, à l'interfaçage des ports d'entrées-sorties, etc.

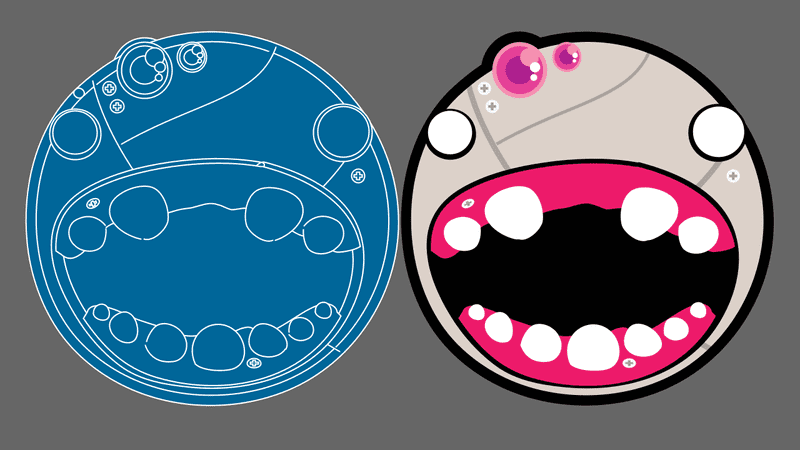


Figure 3 : Exemple d’animation d’objet

## P5.JS :

**p5.js** est une bibliothèque **JavaScript** pour le codage créatif, avec un accent sur rendre le codage accessible et inclusif pour les artistes, les concepteurs, les éducateurs, les débutants et toute autre personne ! **p5.js** est libre et open-source parce que nous croyons que les logiciels, et les outils pour l’apprendre, devraient être accessibles à tous.  
  
En utilisant la métaphore d’un **sketch**, **p5.js** a un ensemble complet de fonctionnalités de dessin. Cependant, vous n’êtes pas limité à votre toile de dessin. Vous pouvez penser à toute la page de votre navigateur comme votre **sketch**, y compris les objets **HTML5** pour le texte, l’entrée, la vidéo, la webcam et le son.



Figure 4 : p5. Javascript

## Création du code :

Le jeu se compose d’un espace de jeux, comme le **Background** (en faire simuler la table de Ping Pong), une balle et deux joueurs. Aussi il comporte en menu d’interface avant de jouer et qu’on peut le nommer **Main**.

## . Création du Background :

Avant tous, on va créer notre fenêtre avec des dimensions spécifiques, tel qu’on écrit :

* Void setup () {

}

Avec la commande **« size (width, height) »**, on dimensionne notre fenêtre tel que son largeur et son hauteur

Pour créer le **Background**, on va utiliser la fonction « draw », talque on peut l’écrire avec cette syntaxe :

* Void draw () {

}

On a spécifié une photo spéciale pour le **Background**.



Figure 5 : Background spéciale designer par Chahid Khalil

On peut ajouter une image (nommé par exemple **‘ Khalil ‘**) avec ces étapes :

* Créer un **« PImage X »** avant le setup.
* Puis, donné une valeur à **X** tel que : X = loadImage(‘’Khalil.jpg’’).
* On fait l’appel à cette image dans le draw et avant de faire conditionner le transfert entre le Background et le Main : image(X,0,0)

L’mage **khalil** doit être trouver dans un dossier nommé **« data »** qui se trouve dans l’emplacement de notre projet.

Le code généralement est gérer par une condition. On a défini un entier nommé **‘’ stage ‘’** qui a une valeur générale de 1.

Si stage égale à 1, on a dans le Main et si stage égale à 2, on va déplacer vers le **Background**.

On commande ce transfert entre les pages par la fonction **« keyPressed »**, tel que :

* Void keyPressed () {

if (key == ‘x’) {

stage = 2 ;

} else if (key == ‘p’) {

stage = 1 ;

}

Ça veut dire que lorsqu’on appuie sur le bouton **‘ x ‘**, on déplace vers **le stage 2** (**Background**), et lors d’un appui sur le bouton **‘ p ‘** en revient **au stage 1** (**Main**).

## 3.2. Création du Main :

Pour créer le Main, il suffit d’apporter une image designer spécifiquement pour ce jeu, comme on a fait dans la **Background**.



Figure 6 : Main spéciale designer par Chahid Khalil

(La photo original avant l’édit revient au jeux olympique Tokyo 2021)

Pour ajouter un indice pour savoir qu’il faut appuyer sur x pour naviguer vers le **Background**, on ajoute simplement un texte **‘’PRESS X‘’**.

Mais pour avoir un sens plus artistique, on peut construire un **Fade**, c’est le fait de faire masquer et afficher le texte dans un temps de quelque seconde (2 par exemple).

Cet effet va se répéter. Donc on va créer une fonction nommée **« textEffect »**, et le faire appeler dans le **Main**.

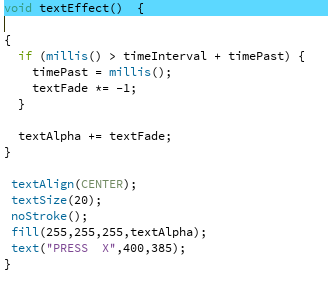


Figure 7 : La fonction textEffect

Et voici notre résultat final du **Main** :



Figure 8 : Le Main

## 3.3. Les joueurs :

D’abord, on va créer une class nommée **« Player »**. **Qu’est-ce que c’est une classe ?**

Class est un composite de champs (données) et de méthodes (fonctions faisant partie de la classe) qui peuvent être instanciés en tant qu’objets. La première lettre d’un nom de classe est généralement en majuscules pour la séparer des autres types de variables.

On représente notre Player sous la forme d’un rectangle, et on le donne un type **Boolean**.

**Pourquoi ?**

Pour qu’on avoir deux joueurs, gauche et droite.

Puis, on va faire bouger ces joueurs avec une fonction **« move »** et en mettre des limites à ces mouvements, donc on utilise la commande **« constrain ».**

**Constrain** permet le constrain d’une valeur à ne pas dépasser une valeur maximum et minimum.



Figure 9 : Les joueurs

Pour commander le mouvement des 2 joueurs, la fonction **« keyPressed »** fait intervenir aussi dans ces cas :

* On a **‘z’** et **‘d’**pour le mouvement haut et bas pour le joueur 1 (**gauche**).
* On a **‘4’** et **‘3’** pour le mouvement haut et bas pour le joueur 2 (**droite**).

On utilise aussi une commande **« keyReleased »**, c’est juste pour stopper le joueur dès qu’on stoppe de presser sur les boutons de commande des joueurs.

Pour avoir mieux de flexibilité dans le mouvement des joueurs, on a créé une fonction nommé **« update »**, en utilisant un entier nommé ‘’ **smooth** ‘’ avec une condition, et en utilisant aussi la commande « constrain » pour limiter le mouvement des joueurs dans le cadre (**Background**).

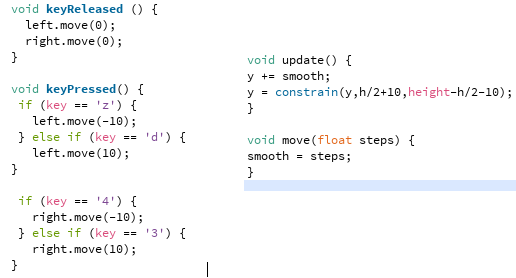


Figure 10 : Commander les joueurs

## 3.4. La balle :

Comme on a créé une classe pour les joueurs, on va aussi créer une classe pour la balle.

On se basant sur l’exemple de **« Bouncing Ball »** sur le **Processing**, on va appliquer la notion des coordonnées cartésiennes au coordonne polaire, en utilisant le théorème de **Pythagoras**.

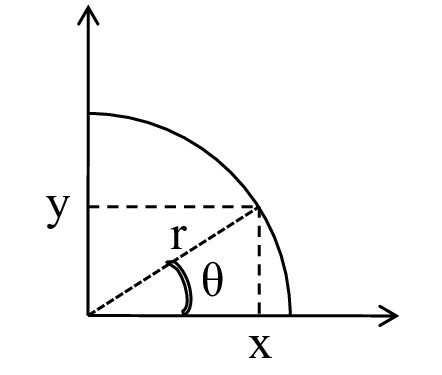


Figure 11 : Coordonnées cartésiennes et polaires

**C’est quoi la notion des vecteurs ?**

C’est une classe pour décrire un vecteur bidimensionnel ou tridimensionnel, en particulier un vecteur euclidien (également appelé géométrique). Un vecteur est une entité qui a à la fois l’ampleur et la direction. Le type de données, cependant, stocke les composants du vecteur (**x,y pour 2D**, et **x,y,z pour 3D)**.

Dans notre cas, on va utiliser deux vecteurs, un pour la position de la balle et le deuxième pour la vitesse de la balle.

Donc, faire additionner ces deux vecteurs va bouger notre balle, **mais dans quel sens ?**

Puisque la balle est une ellipse avec un rayon **r**, elle va se déplacer par exemple entre **PI/4** et **PI/4** dans le concept d’un cercle trigonométrique, en tenant compte du sinus et du cosinus.

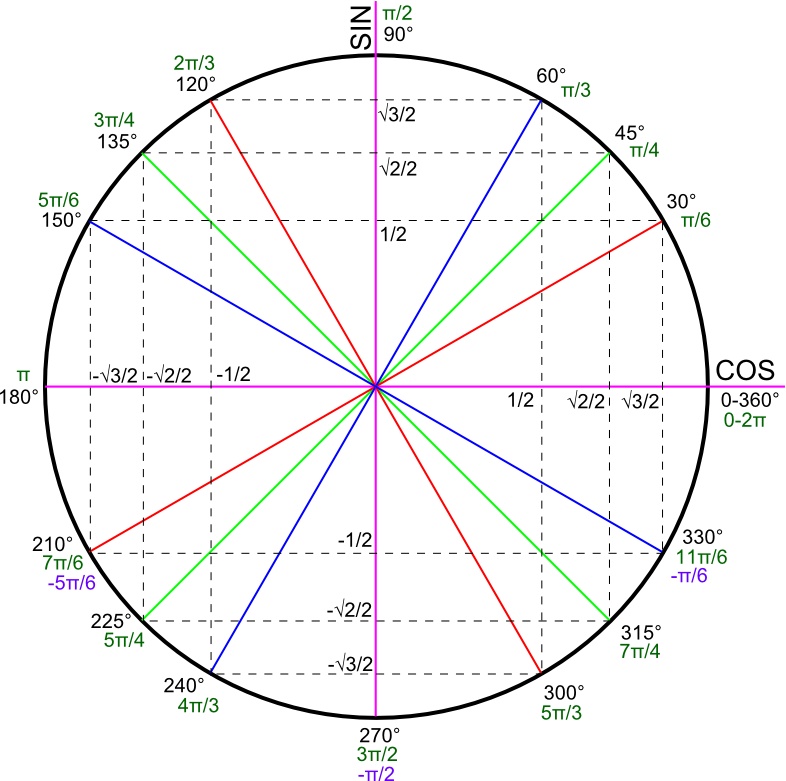


Figure 12 : Cercle trigonométrique

On doit aussi maitres des limites pour le mouvement de la balle dont on a créé une fonction nommée **« edges »**, c’est à dire pour qu’elle reste dans le cadre du **Background**, mais d’une manière spécifique tel que la balle doit se rebondir lorsqu’elle atteint le haut du **Background** (de 0 à **height**), et va quitter dès qu’elle dépasse la largeur du **Background** (de 0 à **width**).

Dès que cette balle dépasse la largeur, elle va pointer un score pour le joueur gauche et le joueur droit. Donc on va incrémenter deux entier (**left score** et **right score**) dans la fonction **« edges »**.

Aussi, dès que cette balle dépasse cette largeur, elle va être restaurer dans le milieu de Background, donc on a créé une fonction nommée « reset ».

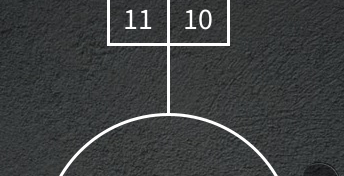


Figure 13 : Score

**Qu’est-ce qu’il nous reste ?**

Il nous reste de faire interpréter la relation, ou plutôt, les conditions du rebond de la balle par rapport au 2 joueurs.

Le concept est de faire détecter la rencontre entre la balle et le joueur et conditionner tel que lorsque la balle faire le premier contact avec le Player, elle va rebondir.

Après plusieurs **bugs** et plusieurs tutoriels sur le **YouTube** et le **Github**, on a finalement trouvé le contact exact entre la balle et les 2 joueurs, bien sûr en tenant compte le rayon et les coordonnées de la balle et les coordonnées des joueurs.

Apres la fixation du point du rebond de la balle par rapport au joueur (ce sont deux points de rebond puisqu’on a deux joueurs), on va mapper l’angle de rebondissement de la balle.

**Qu’est-ce que c’est le « Map » ?**

Utilisez la fonction **« map() »** pour prendre n’importe quel nombre et l’adapter à un nouveau nombre plus utile, ce plus proche du **« constrain »**. Par exemple, dans notre cas, on va mapper une distance sur le long de nos joueurs et de donner un intervalle d’angle, en respectant le sens direct du cercle trigonométrique.

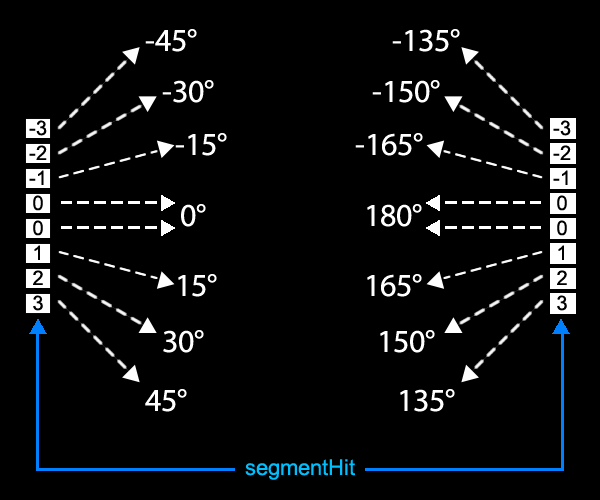


Figure 14 : Pong angle segments (from Github)

# Finalement, voici notre background de jeux :

# 

Figure 15 : Bienvenue chez King Pong

# Conclusion

Bien que ce soit la première fois que j’utilise le Processing, il a été très amusant et passionnant et cela ouvre l’appétit pour découvrir plus de capacité de ce logiciel dans la création des jeux vidéo et des animations.

# Références bibliographiques

[1] http://geeksforgeeks.org

[2] lyceelecorbusier.eu/p5js/

[3] processing.org

[4] https://fr.wikipedia.org/wiki/Processing

[5] https://p5js.org/

[6] https://github.com/NITDgpOS/AirHockey/issues/39

[7] The Coding Train - YouTube