

# PONG9.735496274

---

Un nouveau concept de Pong

## Documentation technique

**Auteur :** Tom Marti

**Classe :** INF3b

**Date :** 17.01.2020

## Table des matières

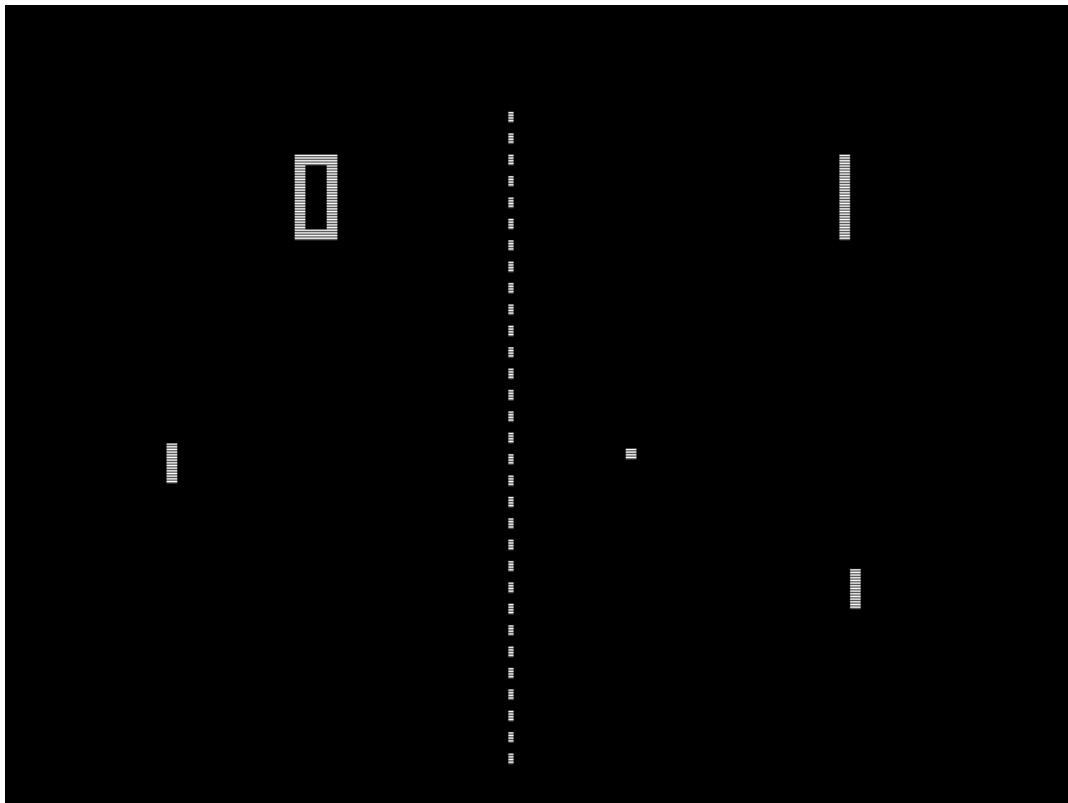
<b>Table des matières .....</b>	<b>2</b>
<b>1 Introduction .....</b>	<b>3</b>
<b>2 Schéma UML .....</b>	<b>4</b>
<b>3 Convention de nommage .....</b>	<b>5</b>
<b>3.1 Variable.....</b>	<b>5</b>
3.1.1 Simple .....	5
3.1.2 Membre .....	5
3.1.3 Pointer .....	5
3.1.4 Pointer Membre.....	5
3.1.5 Constante.....	5
<b>3.2 Fonction .....</b>	<b>5</b>
<b>3.3 Classe .....</b>	<b>5</b>
<b>3.4 Type énuméré .....</b>	<b>5</b>
<b>4 Explication .....</b>	<b>6</b>
<b>4.1 SFML .....</b>	<b>6</b>

## 1 Introduction

Ce projet est un mix entre « **HEADS SOCCER** » un jeu sorti sur smartphone il y a quelques années et « **PONG** » un jeu sorti en novembre 1972 sur Atari. Cela inclut les superpouvoirs de « **HEADS SOCCER** » et le Game Play de « **PONG** ».



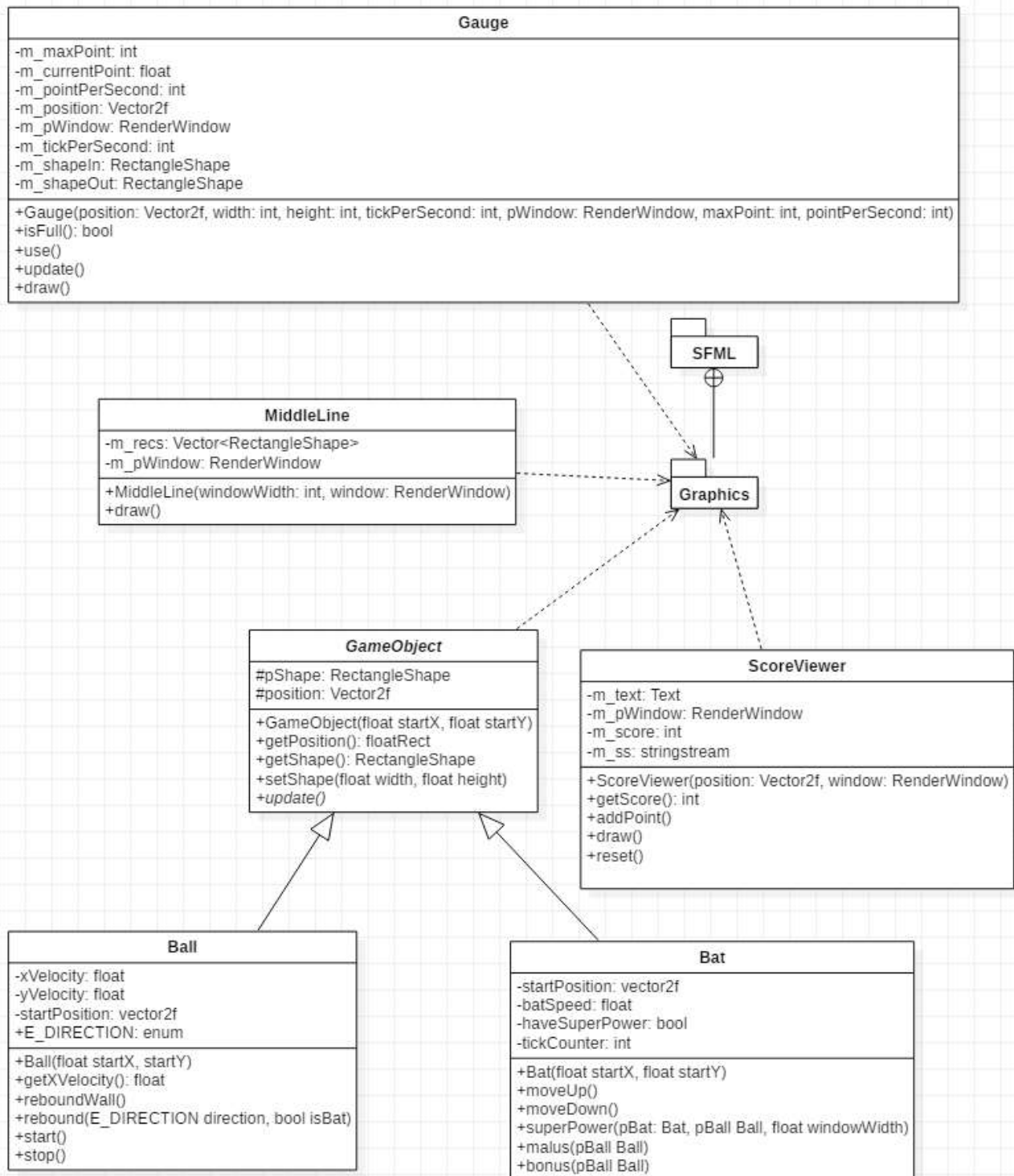
*Heads Soccer*



*Pong*

Le but du jeu consiste à marquer des buts en touchant le mur derrière le joueur.

## 2 Shéma UML



### 3 Convention de nommage

Il est important dans la conception d'un programme de respecter une convention de nommage simple, mais lisible cela permet à une personne extérieure au code de le lire facilement. Dans mon cas j'ai choisi d'utiliser la langue anglaise dans le nommage de mes différents éléments et commentaires.

#### 3.1 Variable

##### 3.1.1 Simple

Les variables ont une majuscule au début de chaque mot excepter le premier.

```
// Variable simple  
int myVar = 0;
```

##### 3.1.2 Membre

Les variables membres ont « m\_ » ajouter devant.

```
// Variable membre  
int m_myVar = 0;
```

##### 3.1.3 Pointer

Les pointeurs ont un p minuscule devant et une majuscule au début de chaque mot.

```
// Variable pointeur  
Ball* pMyVar = new Ball();
```

##### 3.1.4 Pointer membre

Les pointeurs membres sont un mix entre une variable pointeur et une variable membre.

```
// Variable pointeur membre  
Ball* m_pMyVar = new Ball();
```

##### 3.1.5 Constante

Les constantes sont tout en majuscule avec des underscores entre chaque mot.

```
// Variable constante  
int FRAME_RATE = 60;
```

#### 3.2 Fonction

Comme une variable simple.

```
// Fonction  
int myFunc ();
```

#### 3.3 Classe

Tout les mots ont une majuscule.

```
// Classe  
class myClass {};
```

#### 3.4 Type énuméré

Tout en majuscule avec des underscores entre et « E\_ » devant.

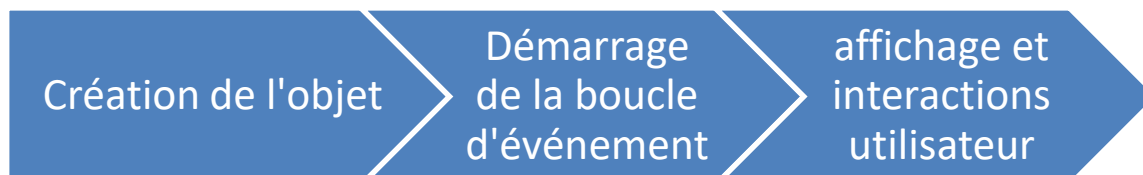
```
// Classe  
enum E_ENUM {};
```

## 4 Explication

### 4.1 SFML

SFML est une librairie multi langage et multiplateforme qui offre une interface simple vers les différents composants d'un PC. Elle se compose de cinq modules : système, fenêtrage, graphisme, audio et réseau. Pour ma part je n'ai utilisé que le module de fenêtrage.

Son fonctionnement est assez simple, au début il faut créer l'objet fenêtre. Ensuite, il faut démarrer la boucle d'évènement. Après, à l'intérieur de cette boucle, nous pouvons interagir avec les éléments de la fenêtre et dessiner les différentes formes.



*Schéma du fonctionnement de SFML*