

Soutenance de projet LIFAPCD

## Pirate Defense

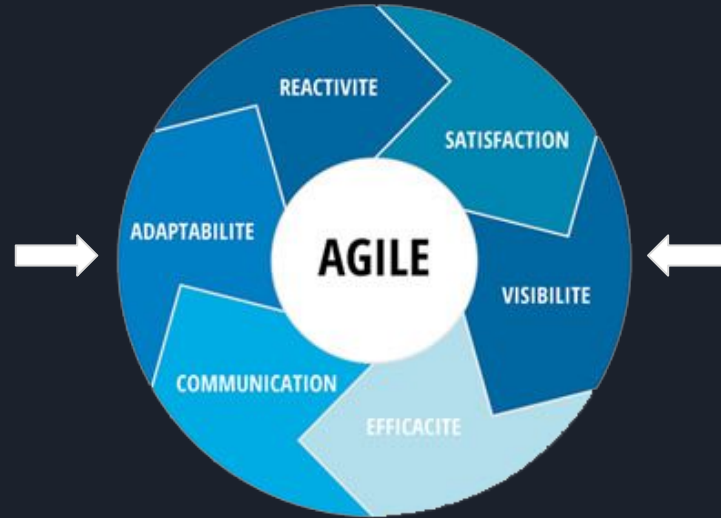
Cheikh Ali Abdoul-Anziz | Riadh Chinoun | Diego Galfré



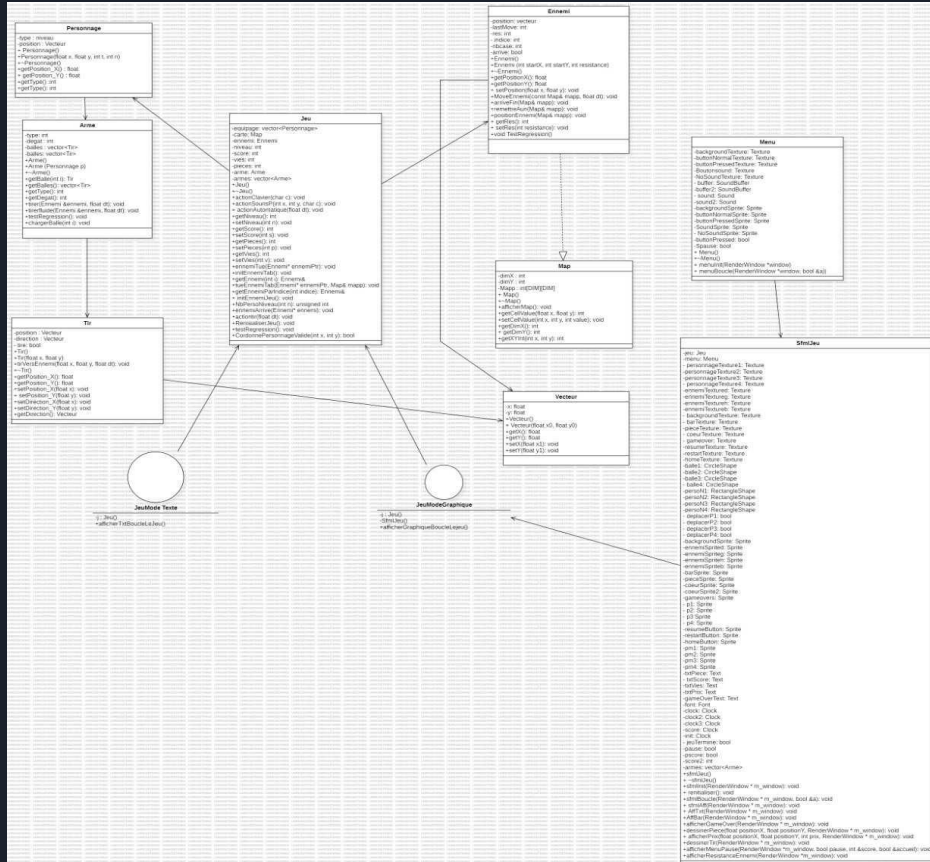
# Diagramme de Gantt :



# Organisation et répartition des tâches :



# Diagramme UML:

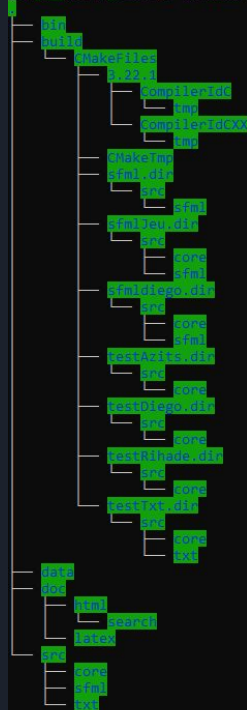


➔ Voir pdf

# Organisation des fichiers & Doxygene :

MVC

```
diego_galfre@DESKTOP-8J40LSH: /mnt/c/Users/diego/OneDrive/Bureau/Pirate Defense/projetjeu$ tree -d
```



## Pirate Defense

Main Page

Classes

Files

### Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<b>Arme</b>	Une arme est représentée par un type qui dépend du niveau d'un personnage, la puissance des dégâts (entier), et un tableau de balles La classe <b>Ennemi</b> représente un ennemi dans le jeu
<b>Ennemi</b>	
<b>Jeu</b>	Un jeu inclut un tableau dynamique de personnages, un objet de type <b>Map</b> , un tableau de cinq ennemis, un tableau dynamique d'armes, et des variables entières pour le niveau, le score, la vie et les pièces
<b>Map</b>	Une map= une matrice de dimension (dimx,dimy)
<b>Menu</b>	Permet de passer de la page d'accueil au jeu
<b>Personnage</b>	<b>Personnage</b> = position (un vecteur à deux dimension) et un type
<b>sfmlJeu</b>	Gère le jeu en utilisant la bibliothèque SFML pour l'affichage graphique. Cette classe gère le jeu en utilisant la bibliothèque SFML pour l'affichage graphique. Elle contient des fonctionnalités pour initialiser SFML, gérer la boucle principale du jeu et afficher les éléments du jeu
<b>Tir</b>	Représente un tir dans le jeu, caractérisé par sa position (vecteur 2D) et sa direction (vecteur2D)
<b>Vecteur</b>	Un vecteur est composé de coordonnées x,y: float
<b>WinTXT</b>	Une fenêtre de texte est un tableau 2D de caracteres

# Classes et fonctions importantes :

## Map :

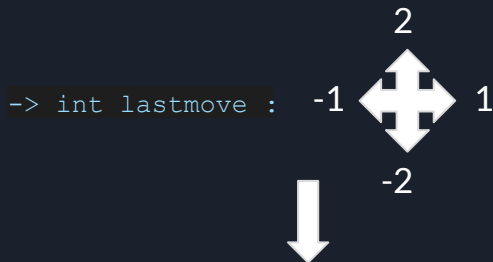
```
{{0,0,0,0,0,0,0,0},  
{1,1,1,0,0,0,0,0},  
{0,0,1,0,0,1,1,1},  
{0,0,1,0,0,1,0,0},  
{0,0,1,1,1,1,0,0},  
{0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0}}
```



```
{{0,0,0,0,0,0,0,0},  
{1,1,4,0,0,0,0,0},  
{0,0,4,0,0,1,1,1},  
{0,0,4,0,0,3,0,0},  
{0,0,1,1,1,3,0,0},  
{0,0,0,0,0,0,0,0},  
{0,0,0,0,0,0,0,0}}
```

## Ennemi :

```
-> void Ennemi::MoveEnnemi(Map& mapp);
```



- Déplacements en fonction de la case sur laquelle il se trouve.



## Jeu :

```
-> void tueEnnemiTab(Ennemi*e_ptr,  
Map& mapp)
```

0	1	2	3
3	0	1	2
2	3	0	1

```
-> (ennemis[i].getIndice -1) mod NBEnnemis
```

```
-> getEnnemiParIndice(0);
```

```
-> void actionAutomatique();
```

```
-> void Jeu::testRegression();
```

# Affichage textuel:

```
Ennemi 0 Position (x;y) = (2;7) Indice = 0 Resistance = 1 arrive? false
Ennemi 1 Position (x;y) = (2;6) Indice = 1 Resistance = 8 arrive? false
Ennemi 2 Position (x;y) = (2;5) Indice = 2 Resistance = 8 arrive? false
Ennemi 3 Position (x;y) = (2;4) Indice = 3 Resistance = 8 arrive? false
Ennemi 4 Position (x;y) = (2;3) Indice = 4 Resistance = 8 arrive? false
```

Pieces: 0

Vies restantes: 5

```
6 0 0 0 0 0 0 0
1 1 0 0 1 1 1 0
0 8 0 0 1 0 1 1
0 8 0 0 1 0 0 0
0 8 0 0 1 0 0 0
0 8 0 0 1 0 0 0
0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0
```

Ennemi mort

```
Ennemi 0 Position (x;y) = (0;0) Indice = 4 Resistance = 8 arrive? false
Ennemi 1 Position (x;y) = (2;7) Indice = 0 Resistance = 8 arrive? false
Ennemi 2 Position (x;y) = (2;6) Indice = 1 Resistance = 8 arrive? false
Ennemi 3 Position (x;y) = (2;5) Indice = 2 Resistance = 8 arrive? false
Ennemi 4 Position (x;y) = (2;4) Indice = 3 Resistance = 8 arrive? false
```

Pieces: 5

Vies restantes: 5

```
6 0 0 0 0 0 0 0
1 1 0 0 1 1 1 0
0 1 0 0 1 0 1 1
0 8 0 0 1 0 0 0
0 8 0 0 1 0 0 0
0 8 0 0 1 0 0 0
0 8 1 1 1 0 0 0
0 0 0 0 0 0 0 0
```

```
Ennemi 0 Position (x;y) = (1;2) Indice = 4 Resistance = 8 arrive? false
Ennemi 1 Position (x;y) = (3;7) Indice = 0 Resistance = 7 arrive? false
Ennemi 2 Position (x;y) = (2;7) Indice = 1 Resistance = 8 arrive? false
Ennemi 3 Position (x;y) = (2;6) Indice = 2 Resistance = 8 arrive? false
Ennemi 4 Position (x;y) = (2;5) Indice = 3 Resistance = 8 arrive? false
```

Version Terminal -> Test Regression

diego\_galre@DESKTOP-8J4OLSHT /mnt/c/Users/diego/OneDrive/Bureau/Pirate Defense/projetjeu

```
Ennemi 0 Position (x;y) = (2;4) Indice = 4 Resistance = 8
Ennemi 1 Position (x;y) = (5;6) Indice = 0 Resistance = 4
Ennemi 2 Position (x;y) = (5;7) Indice = 1 Resistance = 8
Ennemi 3 Position (x;y) = (4;7) Indice = 2 Resistance = 8
Ennemi 4 Position (x;y) = (3;7) Indice = 3 Resistance = 8
```

Score: 13

Pieces: 0

Vies restantes: 5

Nb personnage niveau 1 :1

Nb personnage niveau 2 :0

```
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 1 1 1 0 0
0 0 1 0 0 1 0 1 1 1
3 0 8 0 0 1 0 0 0 0
0 0 1 0 0 1 0 0 0 0
0 0 1 0 0 4 0 0 0 0
0 0 1 8 8 8 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

Version WinTxt



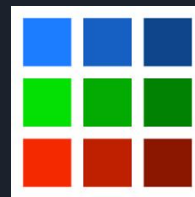
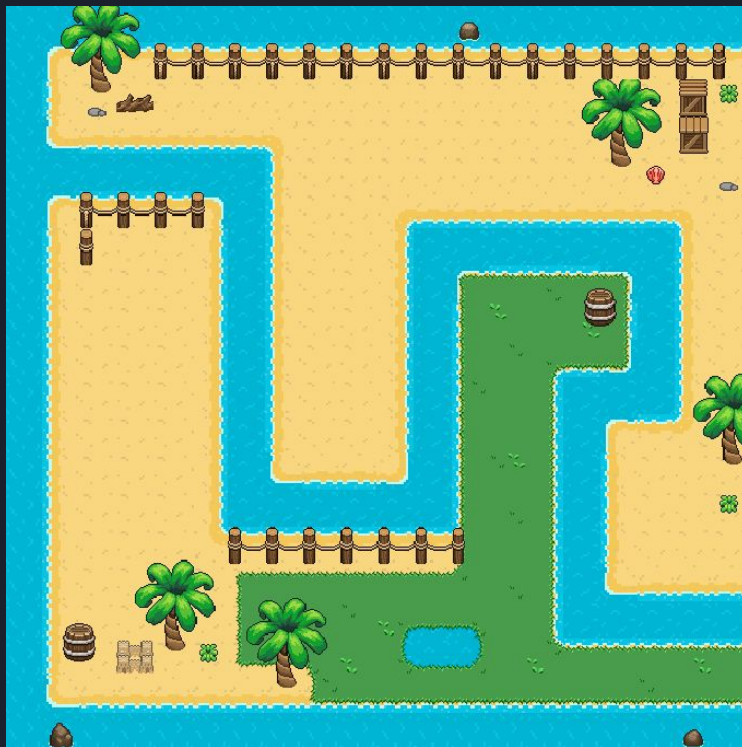
- Gestion d'entrée de l'utilisateur
- Gestion du son/musique
- Portabilité (Window,linux .....)
- `sfmlJeu.h/sfmlJeu.cpp` (843 ligne de code )



- Complet
- Lourd
- Optimiser pour dev Application



# Graphismes & Assets :



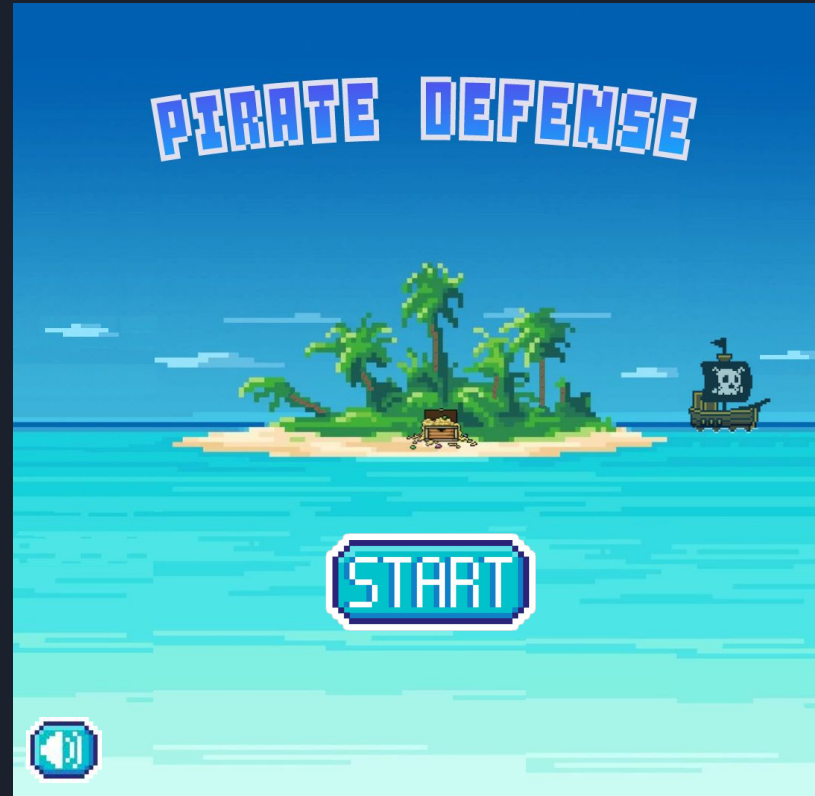
MapleStory



RESTART



# Démonstration et conclusion :





Conclusion: