

Sly Contest

Par Goloviatinski Elisa, Jaton David, Mendes Sébastien et Naddeo Eddy

Sommaire

- État de l'art
- Spécifications
- Technologies utilisées
- Gestion de projet
- Conception
- Démonstration
- Conclusions et résultats

État de l'art

Quelques inspirations

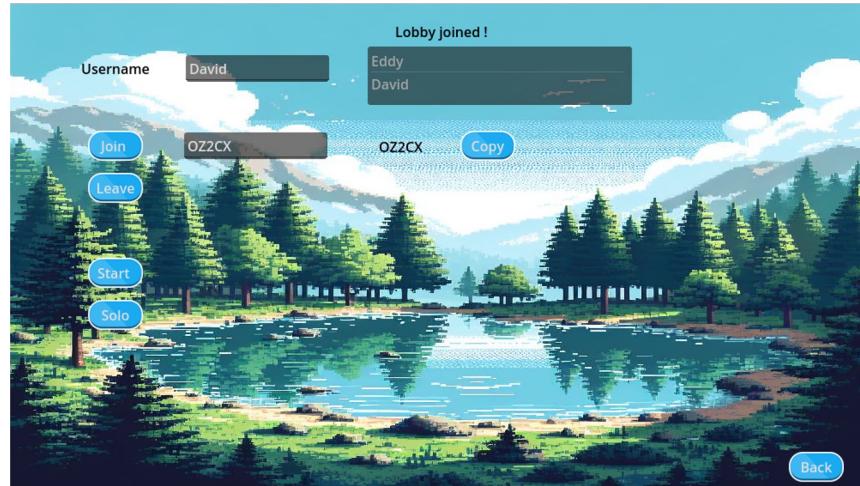


Spécifications

- Multijoueur
- Éditeur de niveau
- Sauvegarde de niveau
- Départ et arrivée aléatoire
- Système de point équilibré

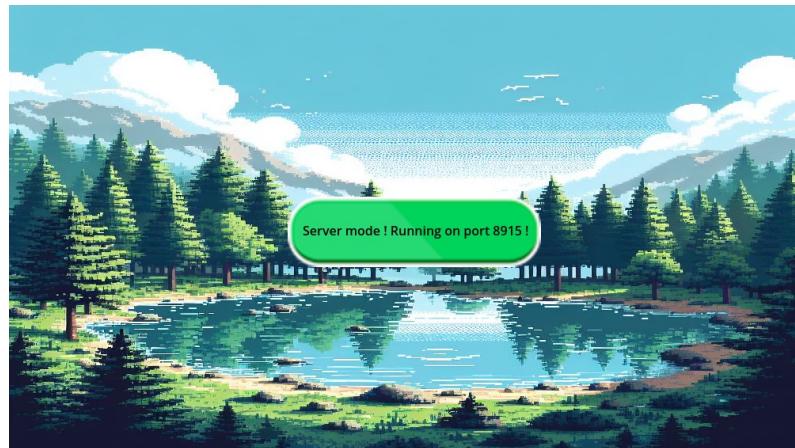
Spécifications

Multijoueur



Spécifications

Multijoueur



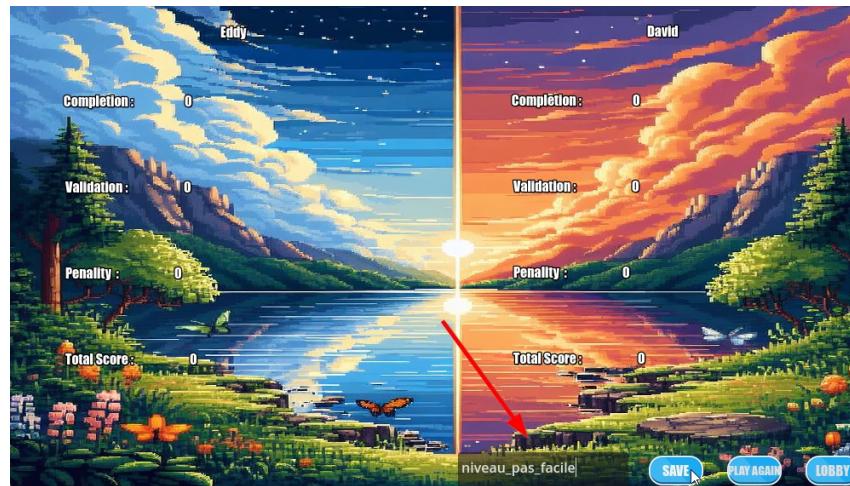
Spécifications

Éditeur de niveau



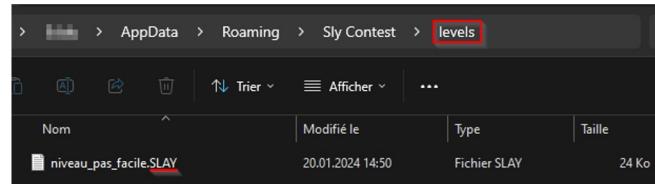
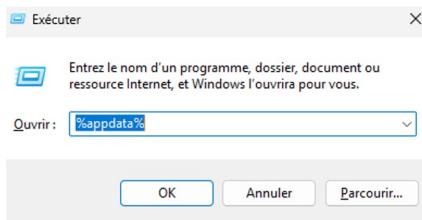
Spécifications

Sauvegarde de niveau



Spécifications

Sauvegarde de niveau



Spécifications

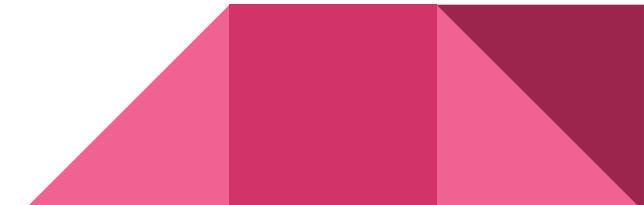
Départ et arrivée aléatoire



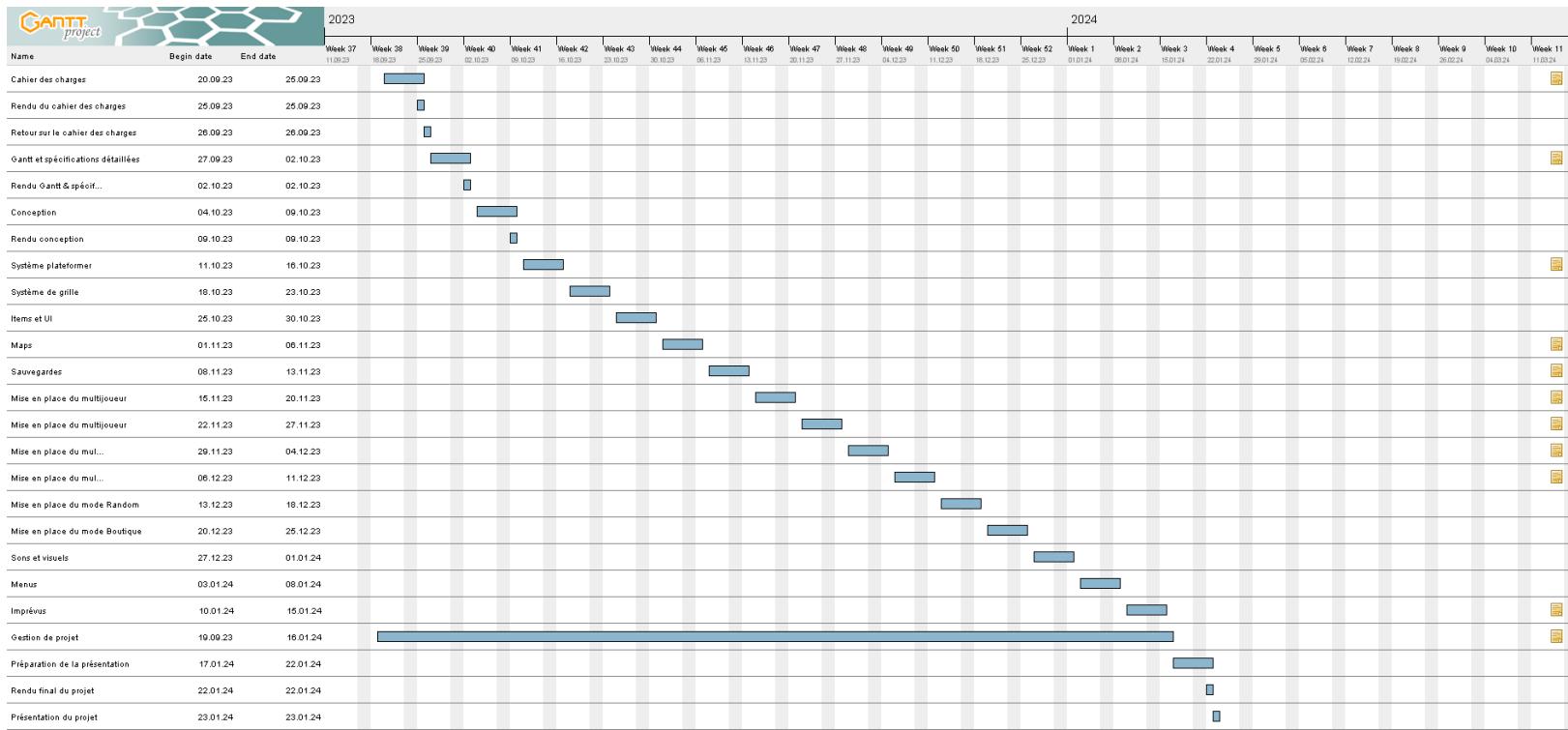
Spécifications

Système de point équilibré

Événements	Nombre de points
Compléter le niveau adverse en premier	300
Compléter le niveau adverse en deuxième	200
Valider son propre niveau avant l'adversaire	100
Personne arrive notre propre niveau alors que l'on a réussi le niveau de l'adversaire	-400



Gestion de projet

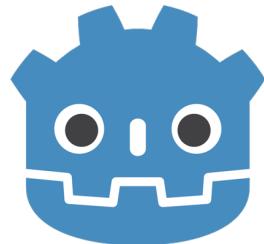


Gestion de projet

- Sébastien :
 - Game physics
 - Game mechanics
 - Points system
- Eddy :
 - Network
 - Scene management
 - Solo mode
- Elisa :
 - Grid system
 - Level editor
 - Traps system
- David :
 - User Interface
 - Sound design
 - Game loop

Technologies utilisées

Godot



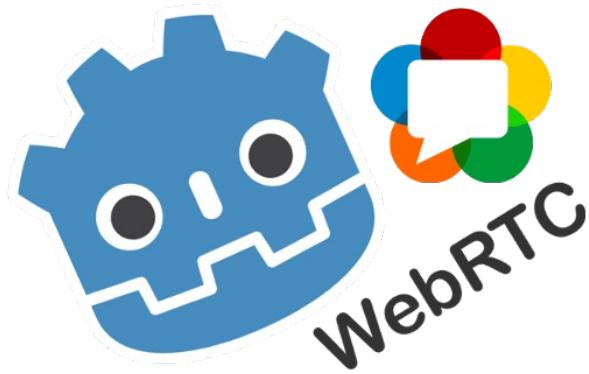
GODOT
Game engine



GDScript

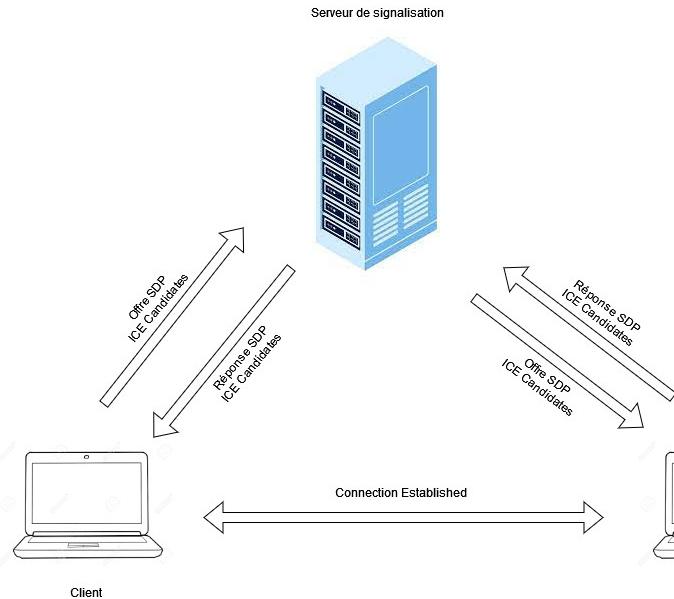
Technologies utilisées

WebRTC



Technologies utilisées

WebRTC



Technologies utilisées

Bing image creator

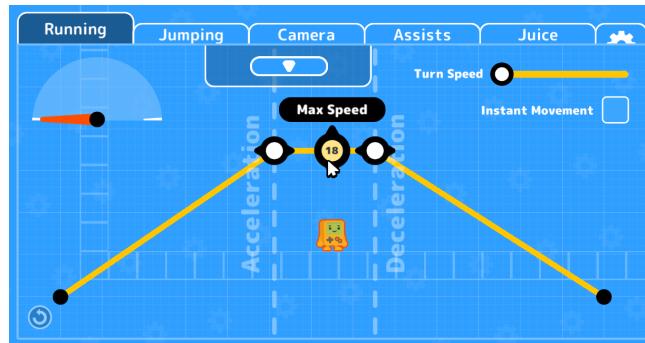


Site internet :



Technologies utilisées

Platformer Toolkit



Site internet :



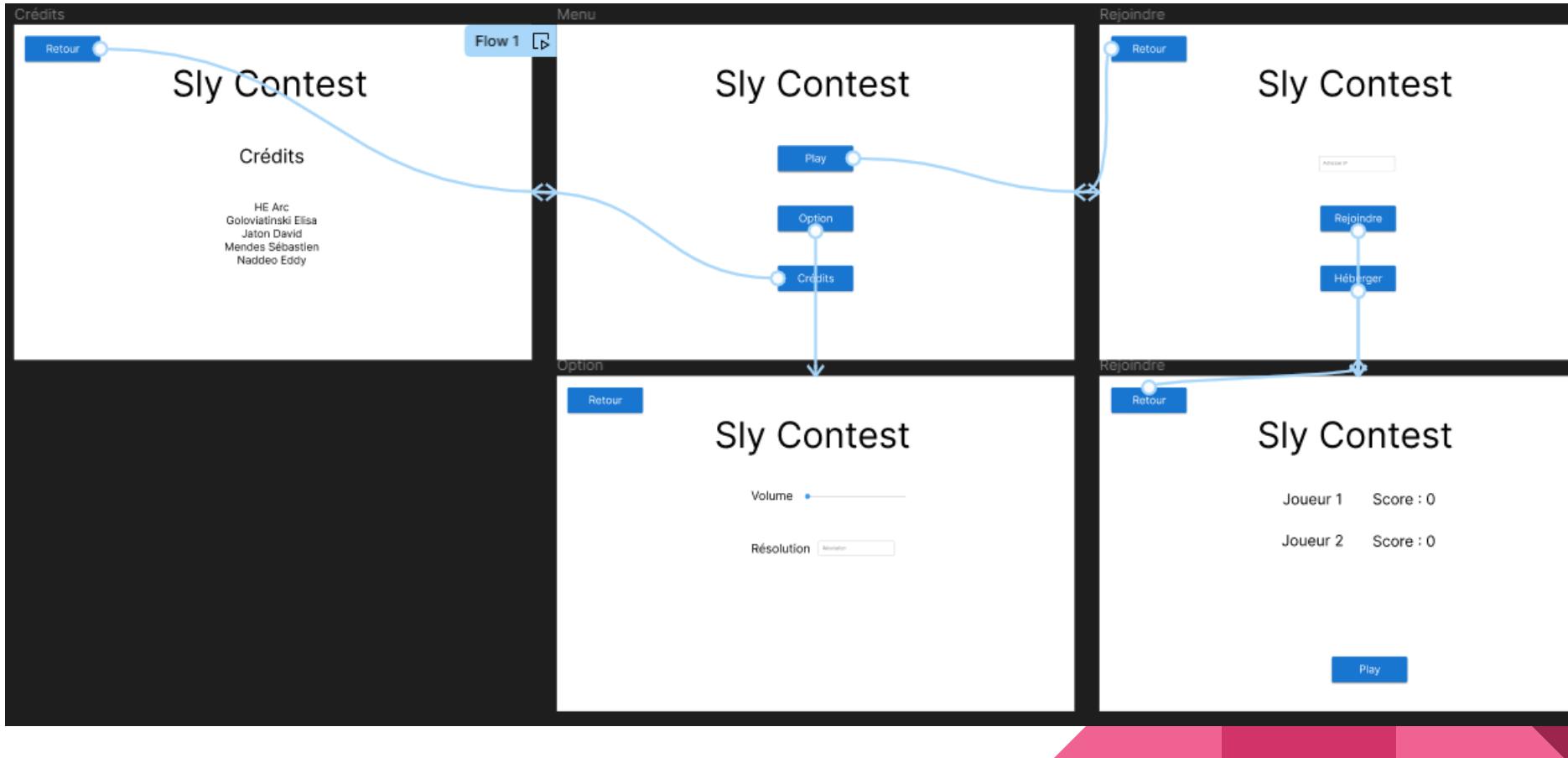
Conception

Maquette



Version interactive :





Retour

Sly Contest

Adresser IP

Rejoindre

Héberger

Retour

Sly Contest

Joueur 1 Score : 0

Joueur 2 Score : 0

Play

Plateau de jeu (Création)

Time left : 4:32

Plateau de jeu (Test)

Time left : 4:32

En test !

Adversaire

Se servir, Annuler, Sauvegarder

Plateau de jeu (Jeu)

Time left : 4:32

En jeu !

Fin de jeu

Joueur 1

Niveau adverse :
Mon niveau :
Pénalité :
Score du round :
Score total :

Joueur 2

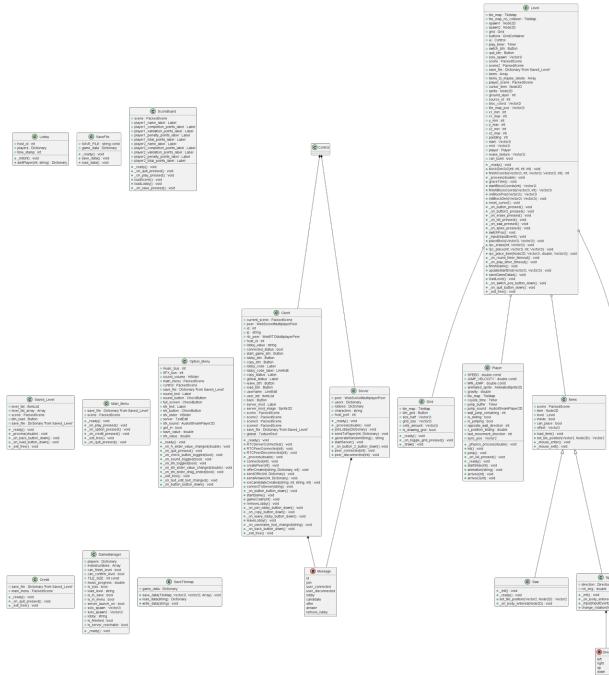
Niveau adverse :
Mon niveau :
Pénalité :
Score du round :
Score total :

Round suivant

Menu Principal

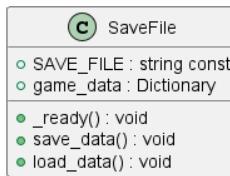
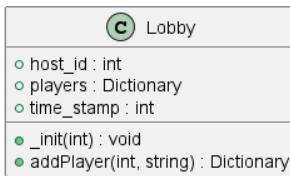
Conception

UML



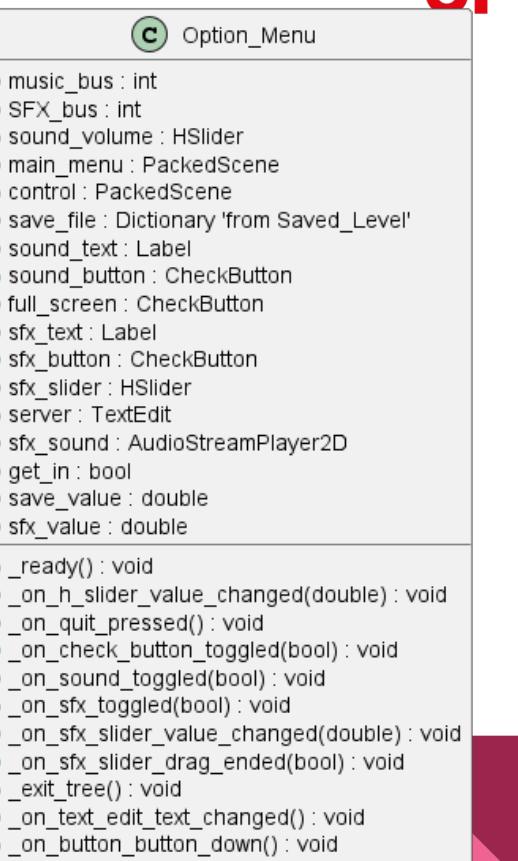
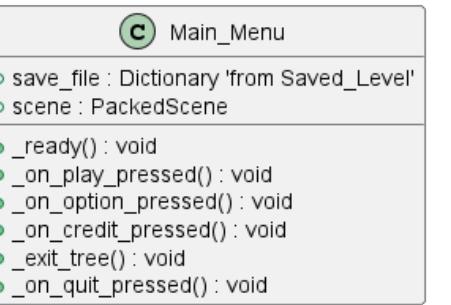
Conception

UML



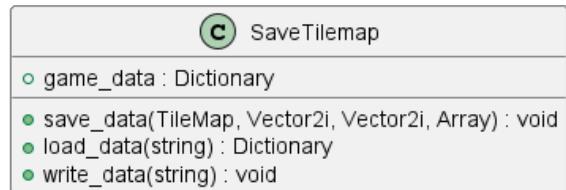
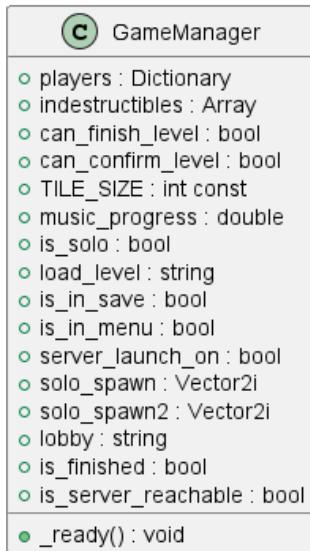
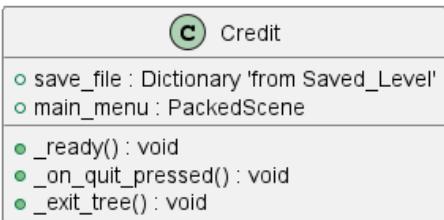
Conception

UML



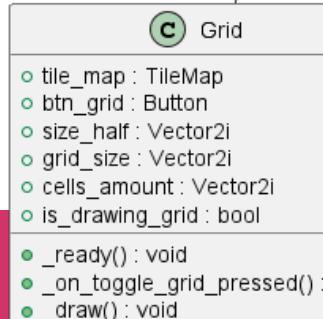
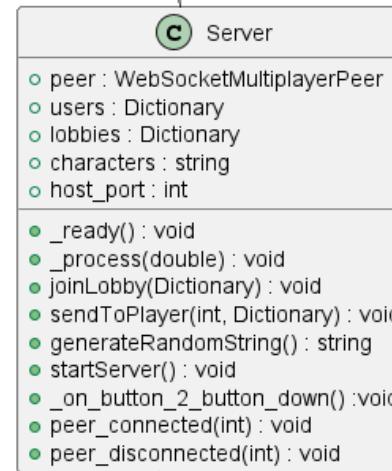
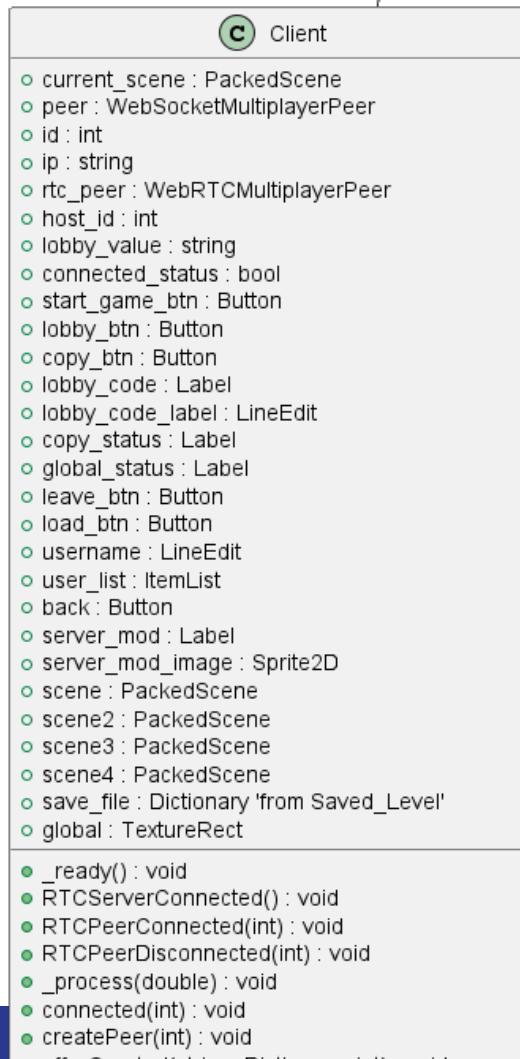
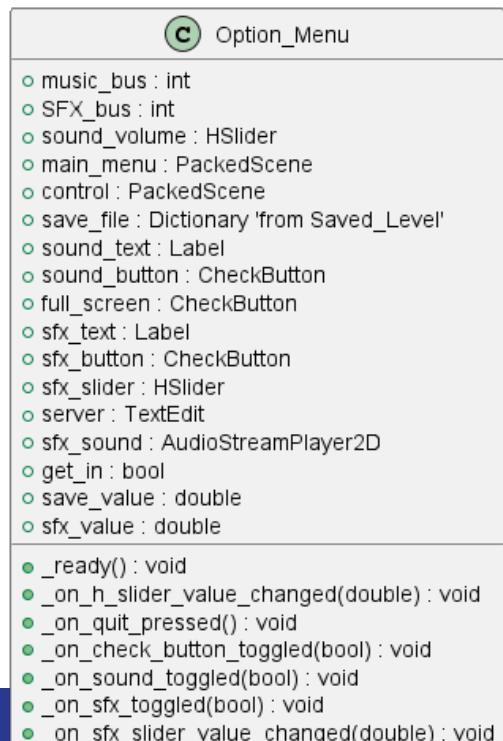
Conception

UML



Conception

UML

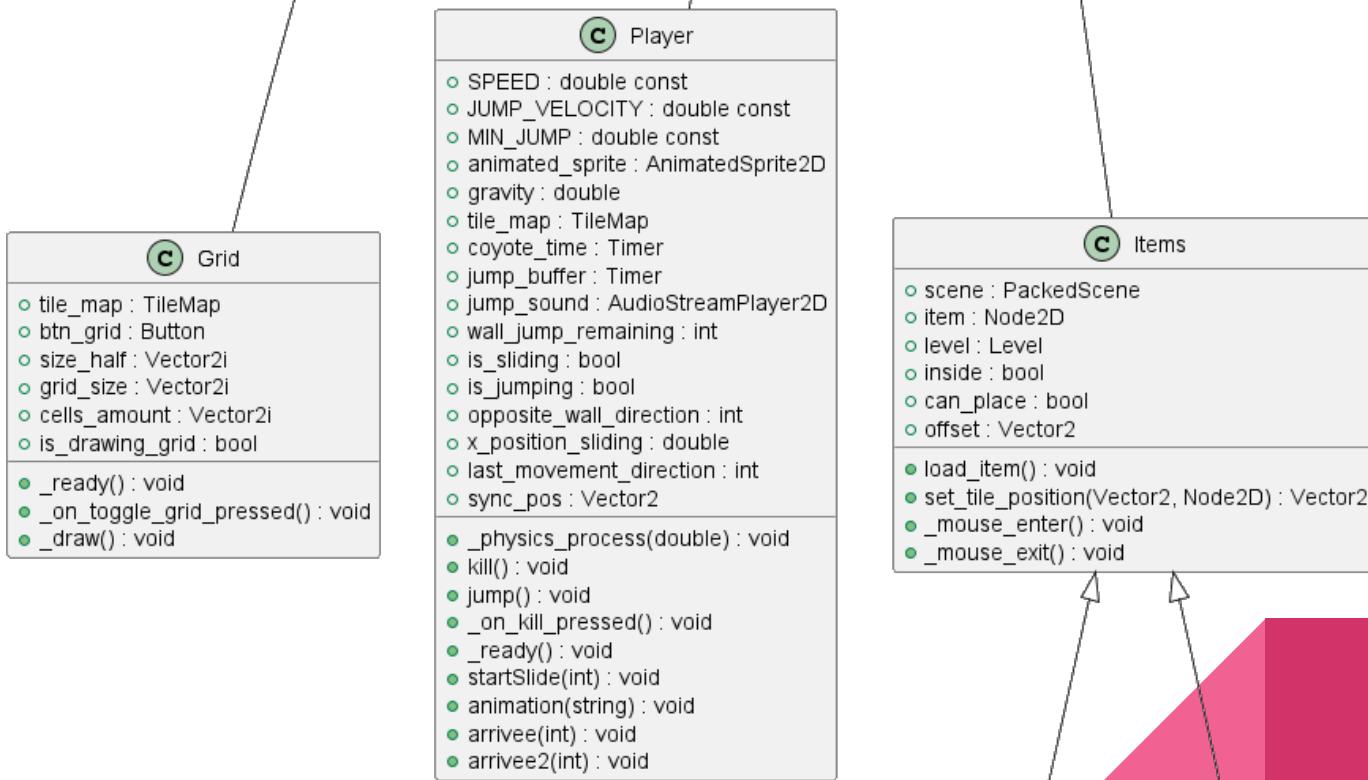


Conception

UML

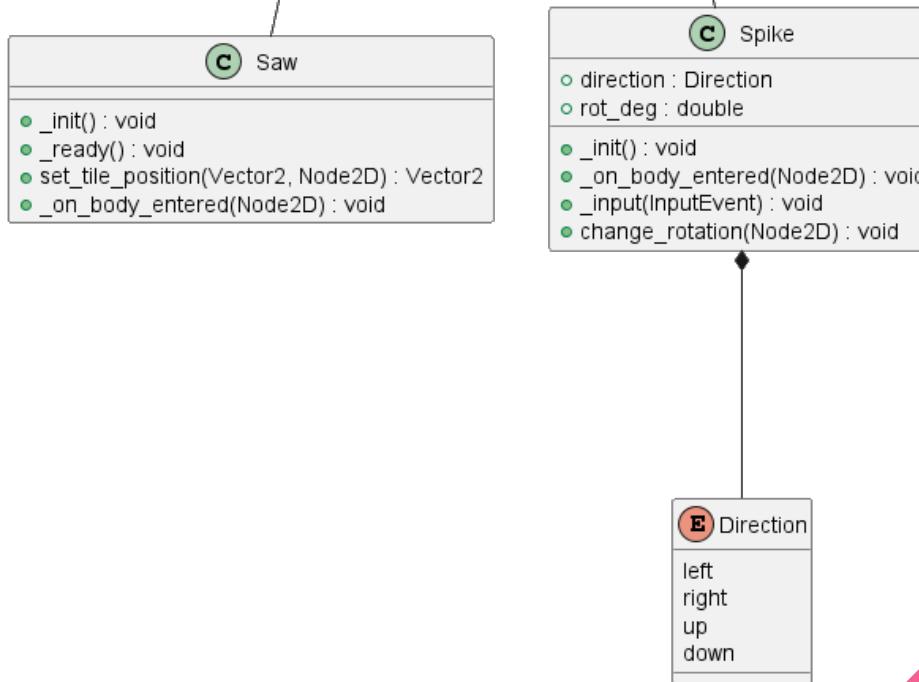
```
server
getMultiplayerPeer
/
ary
g

) : void
ary) : void
c, Dictionary) : void
nString() : string
id
utton_down() :void
nt) : void
ed(int) : void
```



Conception

UML



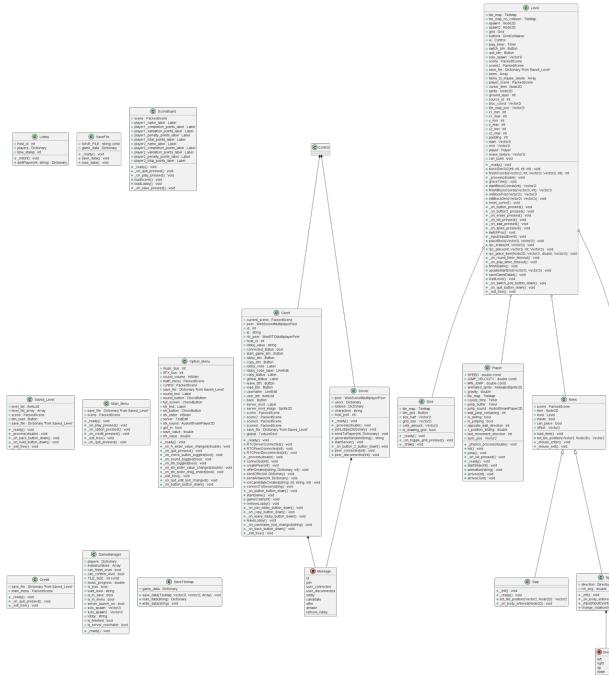
Conception

UML

C Level	
<ul style="list-style-type: none">o tile_map : TileMapo tile_map_no_collision : TileMapo spawn1 : Node2Do spawn2 : Node2Do grid : Grido buttons : GridContainero ui : Controlo play_timer : Timero switch_btn : Buttono quit_btn : Buttono solo_spawn : Vector2io scene : PackedSceneo scene2 : PackedSceneo save_file : Dictionary 'from Saved_Level'o items : Arrayo items_to_maybe_delete : Arrayo player_scene : PackedSceneo cursor_item : Node2Do sprite : Node2Do ground_layer : into source_id : into bloc_coord : Vector2io tile_map_pos : Vector2io x1_min : into x1_max : into y_min : into y_max : int	

Conception

UML



Démonstration



Conclusion et résultats

- Multijoueur ✓
- Éditeur de niveau ✓
- Sauvegarde de niveau ✓
- Départ et arrivée aléatoire ✓
- Système de point équilibré ✓

Questions ?