



Tower Defense

<https://gitlab-cw1.centralesupelec.fr/lescanapdefrancis/tower-defense.git>

Coding Weeks 2024

Alexis Noé, Juliette Oria, Léo Paul Maruitte,
Sacha Le Dily, Adrien Bouvet, Amaury Bonnet

Sommaire

Présentation générale

- Principe du jeu
- Limites et points forts
- Chronologie

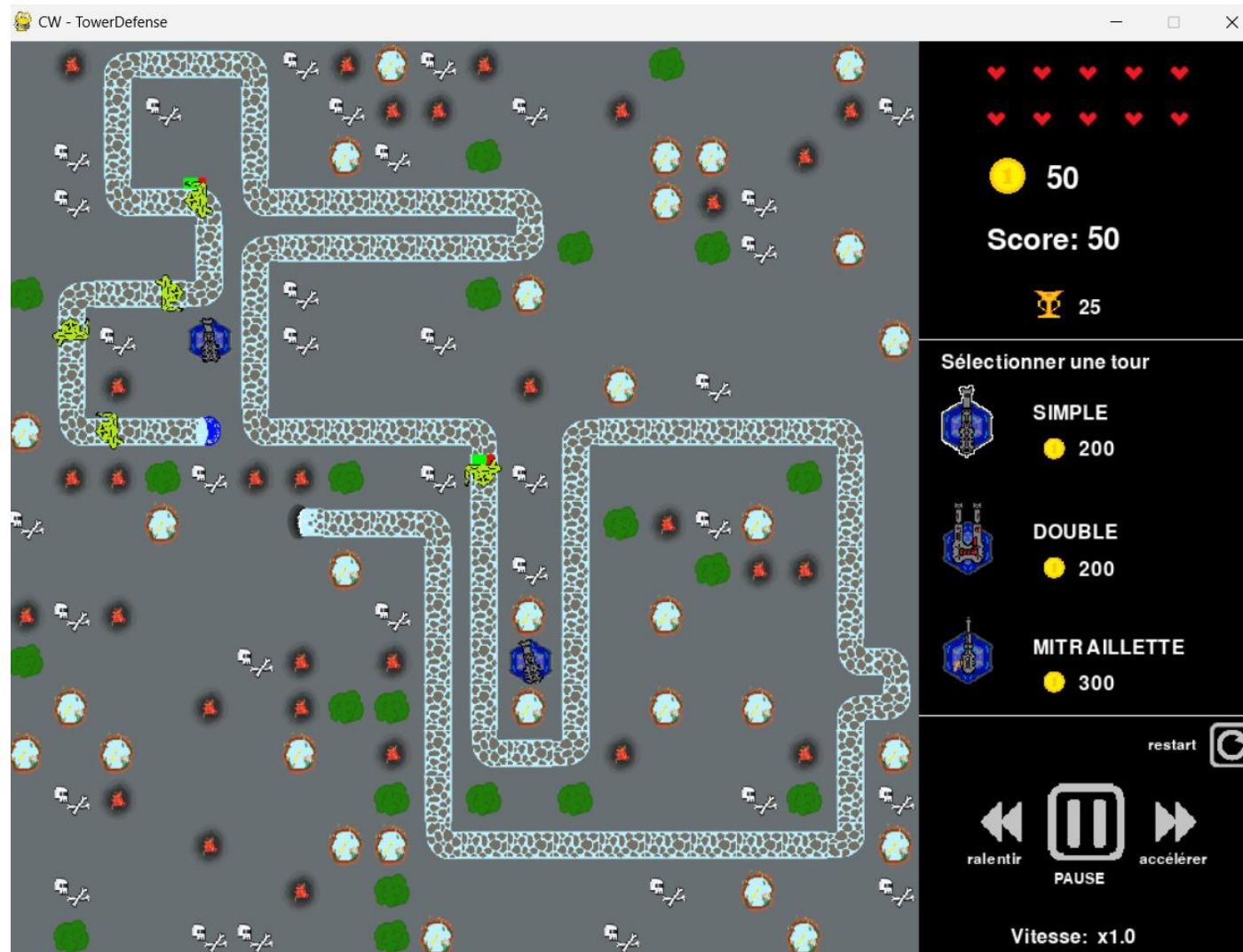
II. Le projet minimal

- Grille
- Tour et projectile
- Ennemi
- Variables de jeu

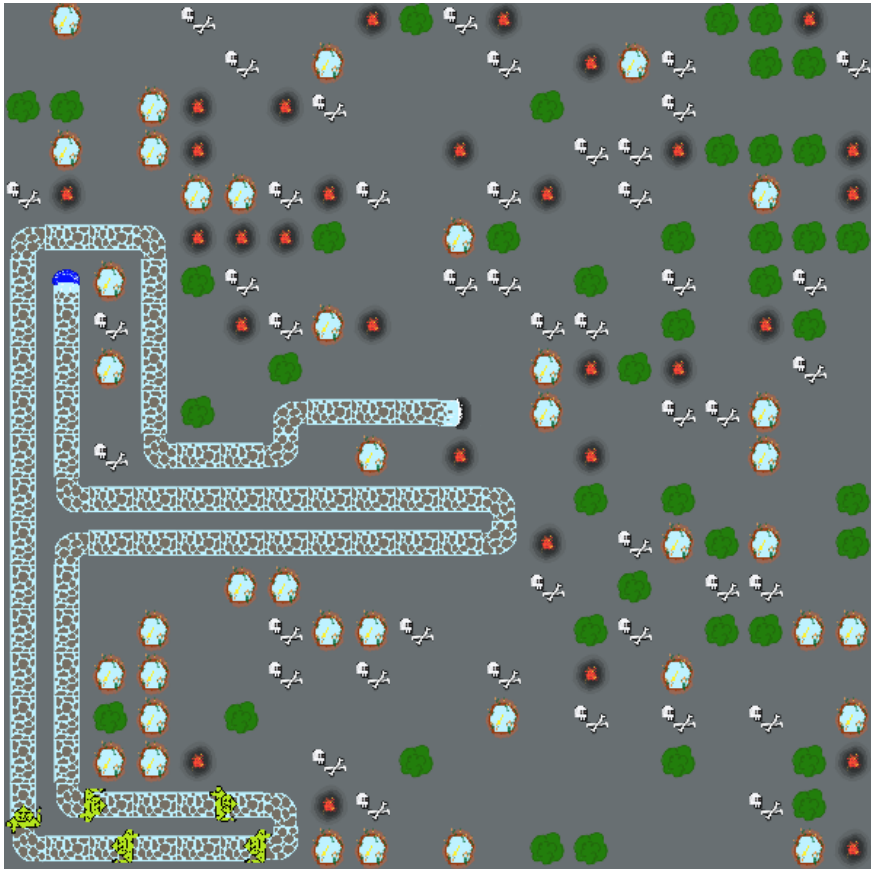
III. Améliorations

- Interface graphique
- Nouvelles fonctionnalités (boutons, vitesse...)

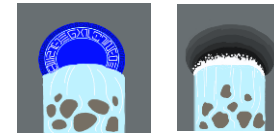
I. Principe du jeu de Tower Defense



Principe du jeu



- Génération aléatoire d'un terrain sur lequel s'étend un **chemin**, avec entrée et sortie



- Apparition d'**ennemis** à l'entrée qui tentent d'atteindre la sortie
- Pour les supprimer, on installe des **tours** qui envoient des **projectiles**

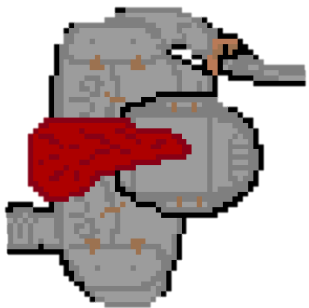
Les ennemis



Gobelins : rapides, nombreux et faibles



Barbares : un peu plus lents, plus difficiles à tuer



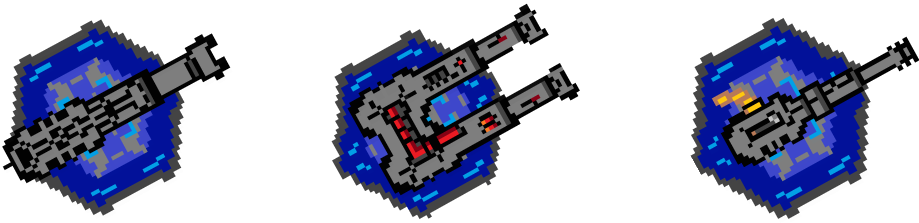
Chevaliers : lents, peu nombreux et très difficiles à tuer

Apparition par **vagues**, par difficulté croissante

Suivent le chemin

Objectif du joueur : tuer les ennemis avant qu'ils atteignent la fin du chemin (sinon on perd des **points de vie**)

Les tours



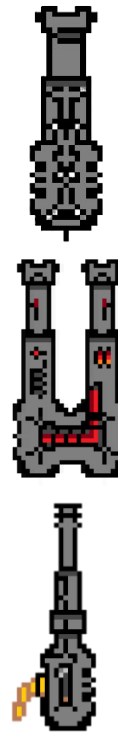
Placées par **un clic** sur une case de la grille

Choix de leur type dans le **menu de sélection**

Nombre limité par leur **prix** (qui augmente)

Destruction possible par un clic

Tirent des projectiles 🎯 vers l'ennemi le plus proche, dès qu'il est à sa portée

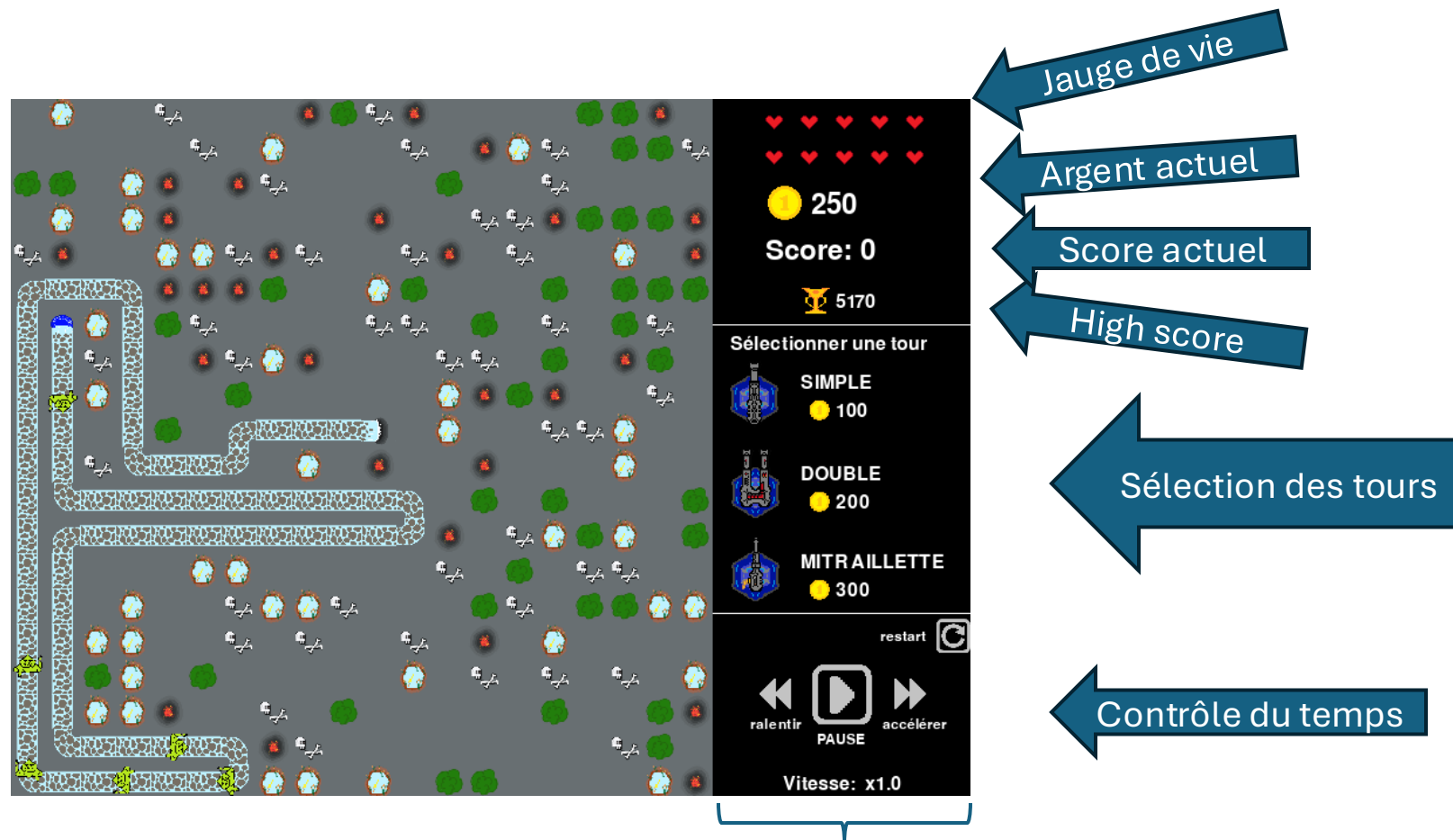


Canon simple : (valeur sûre)
Tire à intervalles réguliers

Canon double :
2 projectiles tirés en simultané
(vers la même cible)
Portée plus faible
Cadence de tir plus lente

Mitrailleuse :
Cadence de tir plus élevée

Menu interactif



Zone d'interaction
Entre le joueur et le jeu

Limites

Chemin aléatoire : +/- avantageux

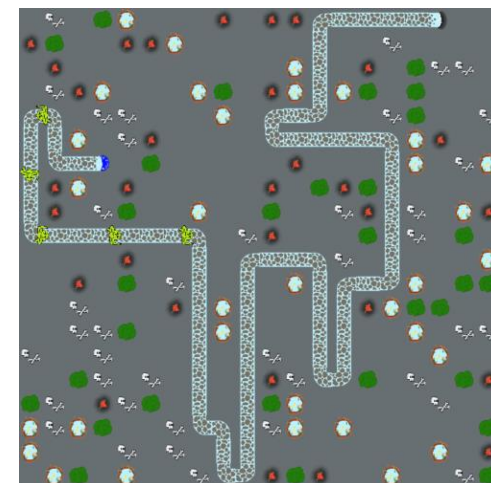
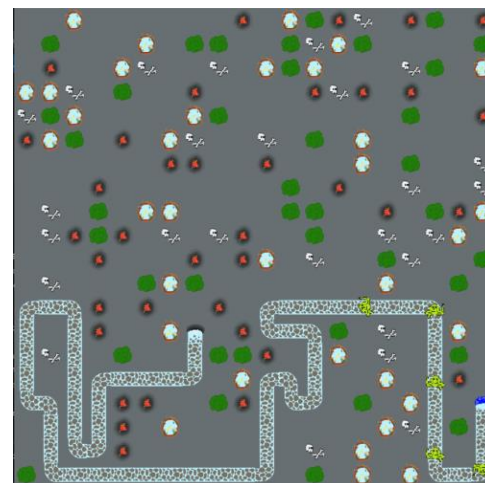
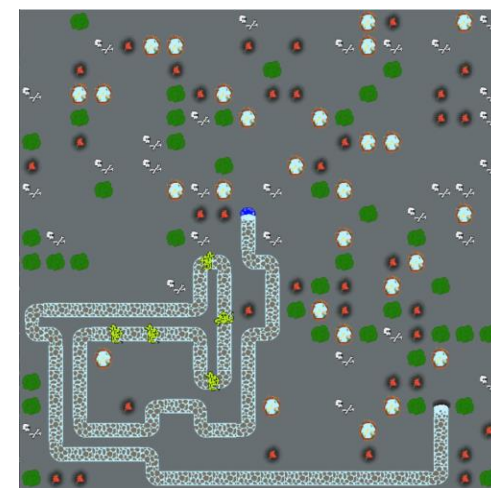
Peu de **types** de tour et d'ennemis

Pas d'**amélioration des tours** une fois qu'elles sont construites

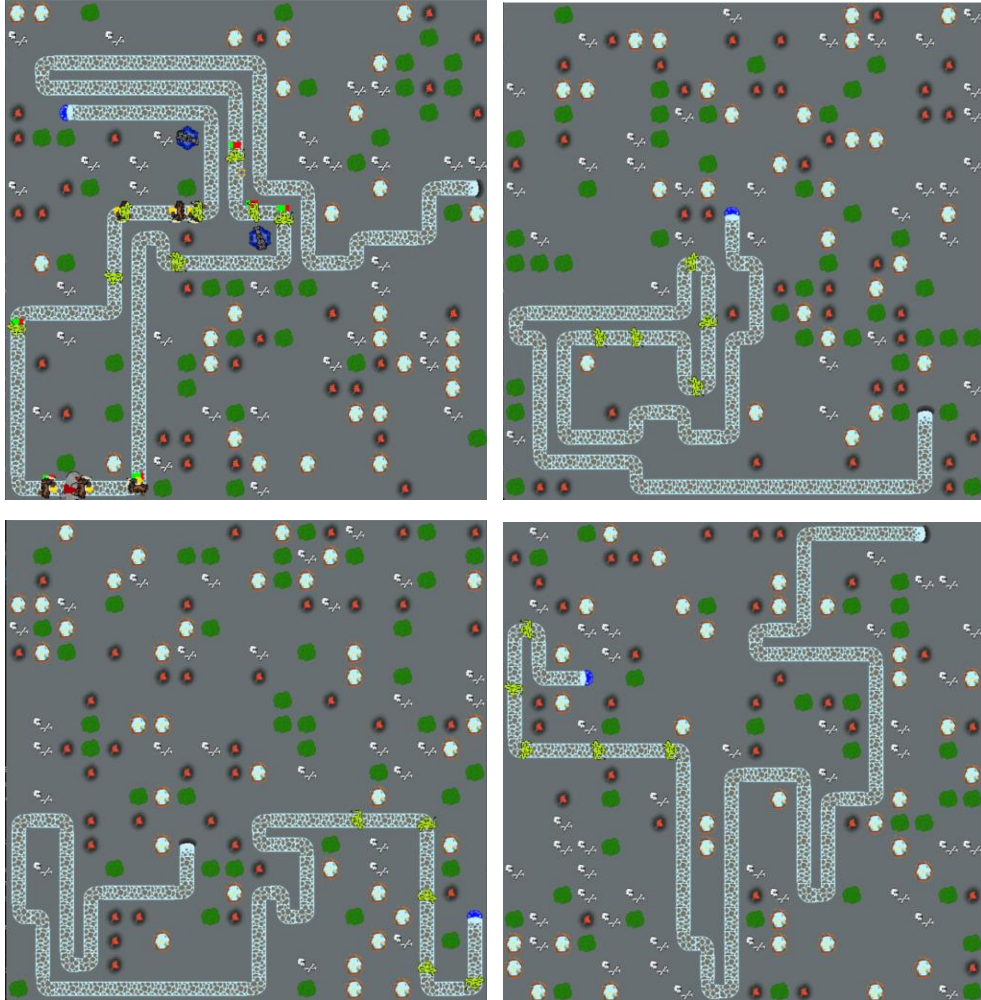
Destruction des tours : problème du **double clic**

Musique entêtante

Taille fixe de la fenêtre (pas de plein écran)



Points forts



Chemin aléatoire
Vagues d'ennemis
aléatoires



Expériences de jeu variées
Jamais 2 parties identiques

Programmation orientée objet :

- Facilité d'amélioration du code (ajout d'ennemis, de tours)
- On crée plusieurs fois le même type d'objet facilement (les projectiles typiquement)

Chronologie du projet

Projet Minimal



Projet Amélioré

Projet minimal, étapes:

Etape 1: grille

- * creation de la grille de jeu, chaque case étant initialisée vide

- * creation d'une fonction pour ajouter une tour -----

- * creation d'une fonction pour ajouter une tuile de chemin, avec une valeur -----

PYGAME: creation d'une fonction renvoyant les coordonnées dans la grille en fonction de coordonnées en x,y

PYGAME: creation d'une fonction d'affichage

Etape 2: chemins /OK/

création des chemins, une classe avec deux méthodes:

- * une méthode qui indique les coordonnées de la prochaine tuile chemin dans le parcours du depart à l'arrivée

PYGAME: une methode pour l'afficher

NB: le départ et l'arrivée des ennemis seront de simples chemins. L'arrivée n'aura pas de tuile suivante.

- * création d'une fonction d'initialisation dans la GRILLE pour créer le chemin

Etape 3: sprites /OK/

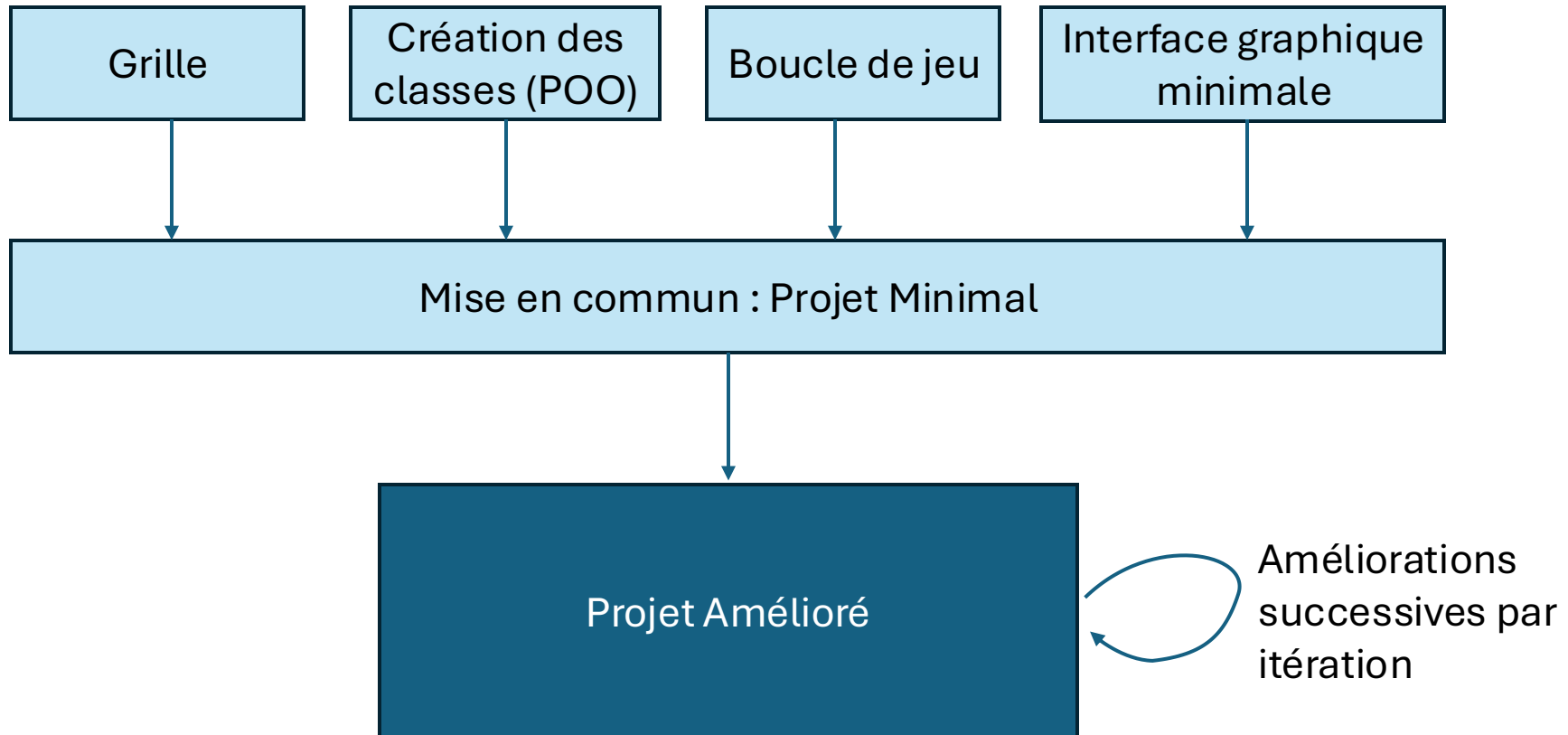
creation d'une classe abstraite sprite, avec

- * des coordonnéesXY

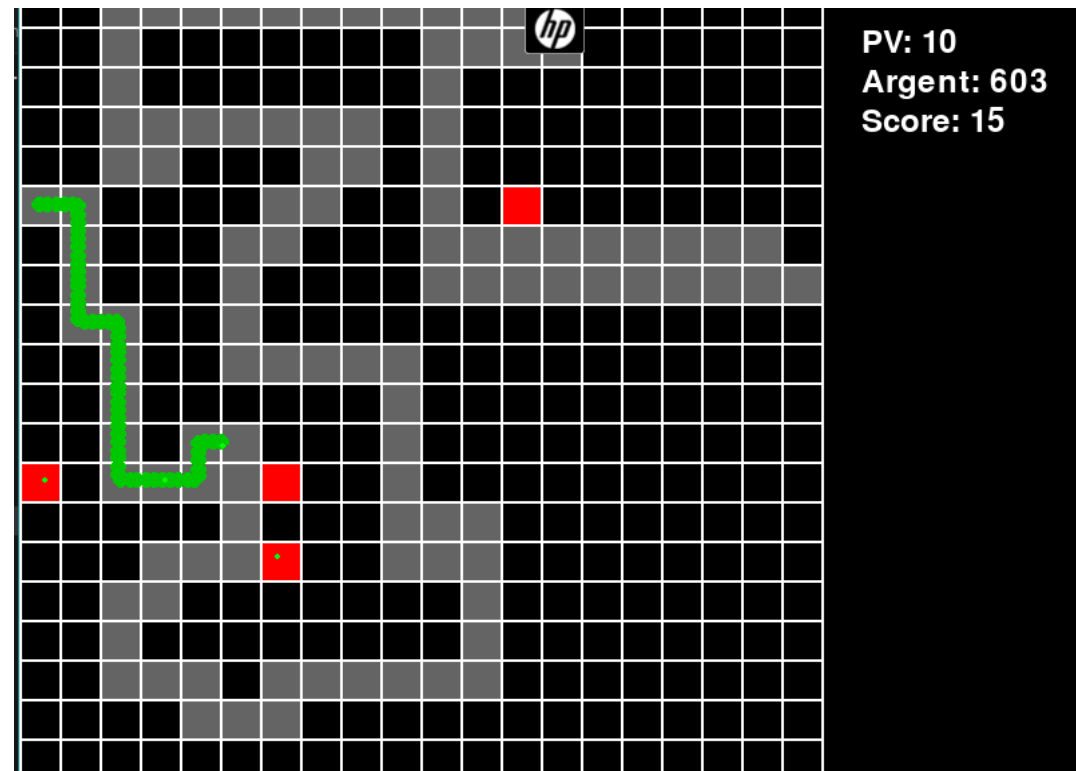
- * une methode abstraite d'affichage (vide)

- * une methode d'initialisation qui le stocke dans une liste de tous les sprites

Principe de développement



II. Le jeu minimal



Implémentation d'une version simple du jeu
Graphismes minimalistes

Répartition des tâches - Jeu minimal

Réflexion initiale

Division du travail en étapes: Alexis

Interface graphique minimale Interactions avec le joueur

Alexis

Fusion des scripts

Alexis

Création de la grille

Grille : Alexis

Création du chemin initial : Léo-Paul

Création des classes

Sprites : Amaury

Ennemis : Amaury

Projectiles : Juliette et Sacha

Tours : Alexis

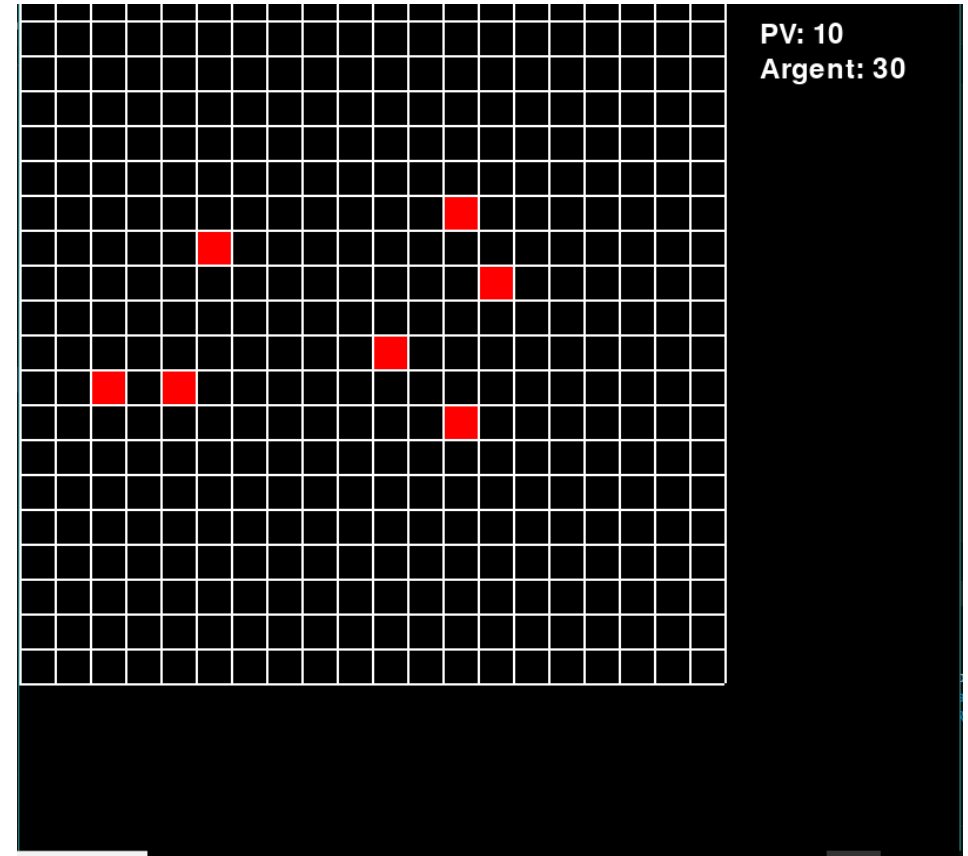
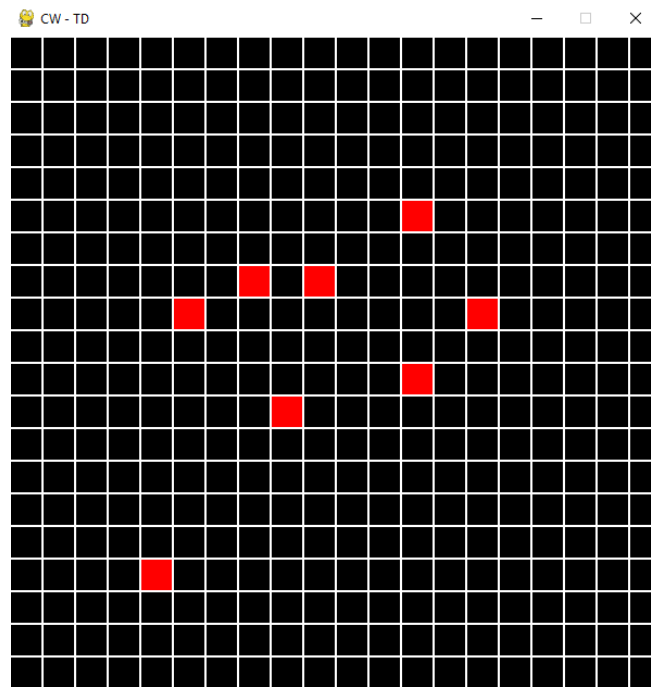
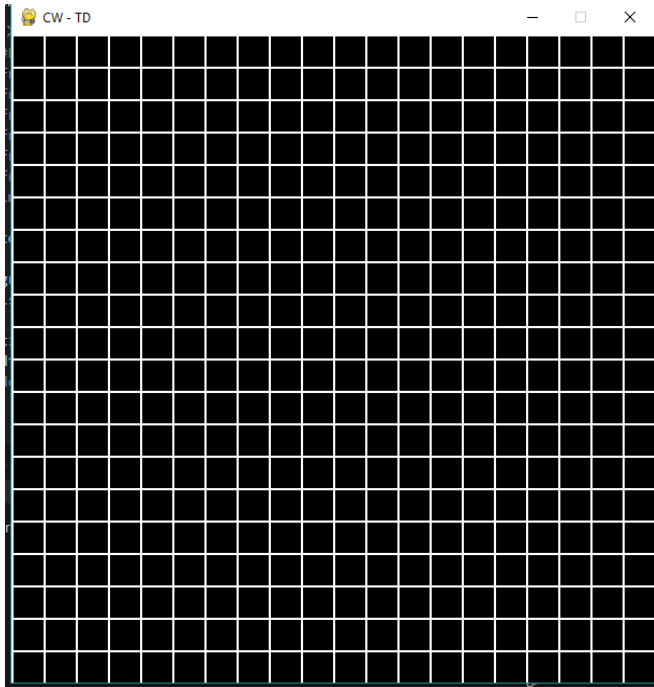
Chemin : Léo-Paul

Case : Léo-Paul

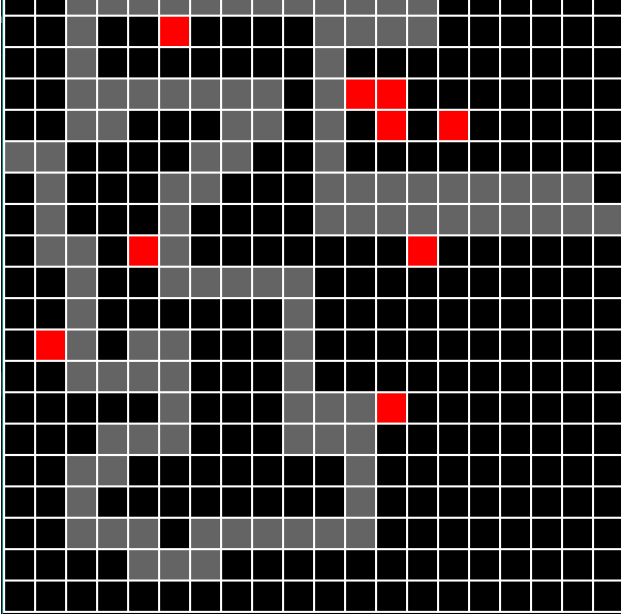
Gestion des variables du joueur (score, argent, points de vie)

Adrien

Jeu minimal: implémentation



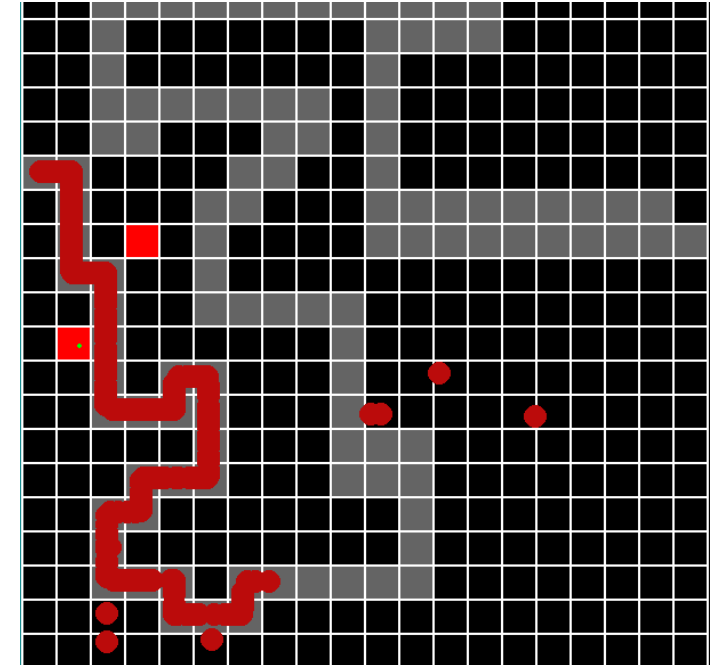
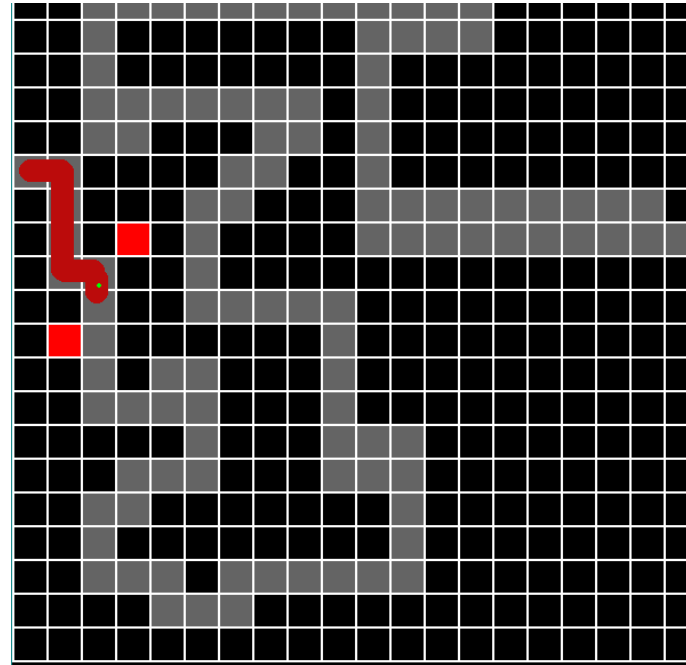
Jeu minimal: implémentation



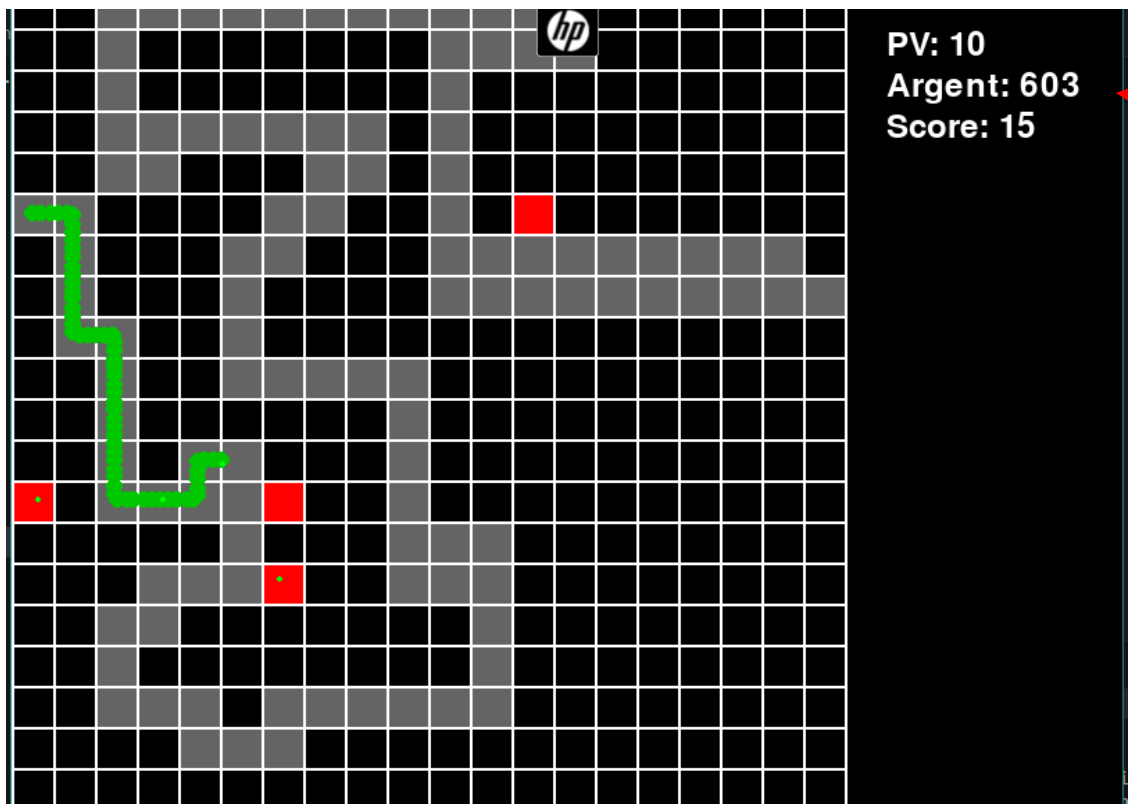
Implémentation d'un premier chemin, non aléatoire :
définition d'une classe '**Chemin**' qui relie les objets '**Cases**'

Création d'**ennemis**, qui suivent le chemin

Création de **projectiles**, tirés par les **tours**
sur l'ennemi le plus proche.



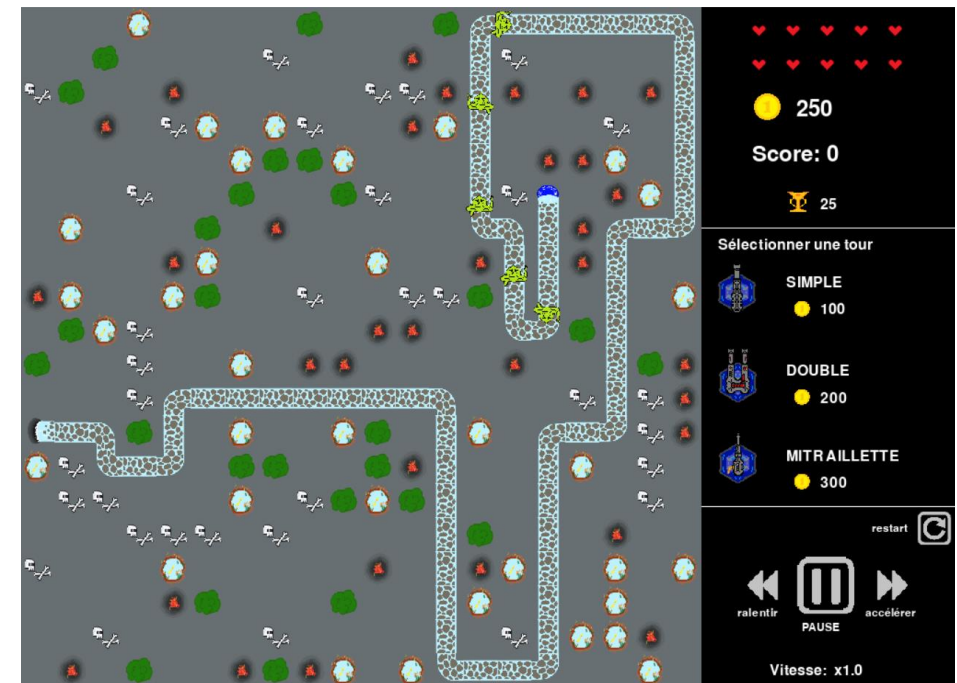
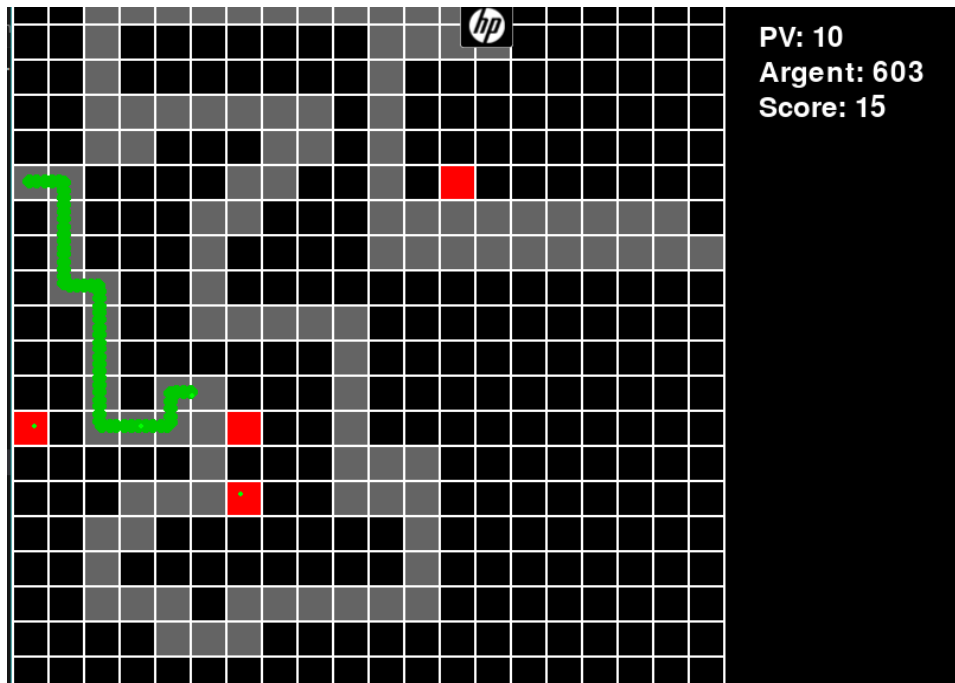
Jeu minimal: implémentation



Création du score

Ennemis apparaissent à intervalles réguliers

III. Les améliorations



Comment passer du jeu minimal ...

... au jeu final ?

Répartition des tâches - Améliorations

Équilibrage du jeu et tests

Alexis

Décomposition du travail en étapes

Alexis

Graphismes

Chemin : Juliette (dessin et affichage)
Décor : Juliette (dessin et affichage)
Tours : Amaury (dessin), Sacha (affichage)
Ennemis : Amaury (dessin), Sacha (affichage)
Boutons : Sacha (dessin et affichage)
Menu de variables : Sacha (dessin et affichage)
Écran de Game-over : Léo-Paul (dessin et affichage)
Jauge de vie des ennemis: Alexis (affichage)

Musique et effets sonores

Sacha

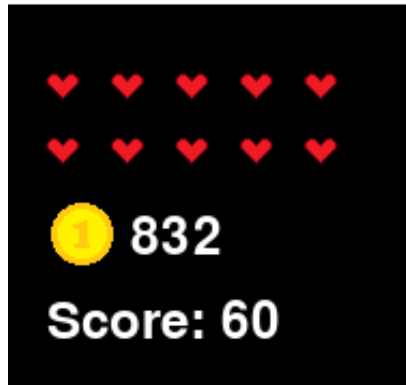
Fonctionnalités

Chemin aléatoire : Alexis
Vagues d'ennemis : Alexis
Choix des tours : Sacha
Choix de vitesse de jeu : Sacha
Pause : Alexis
Boutons: Alexis
Restart : Léo-Paul
Etats de jeu et Game-Over: Léo-Paul
Sauvegarde du meilleur score : Alexis
Suppression des tours: Sacha

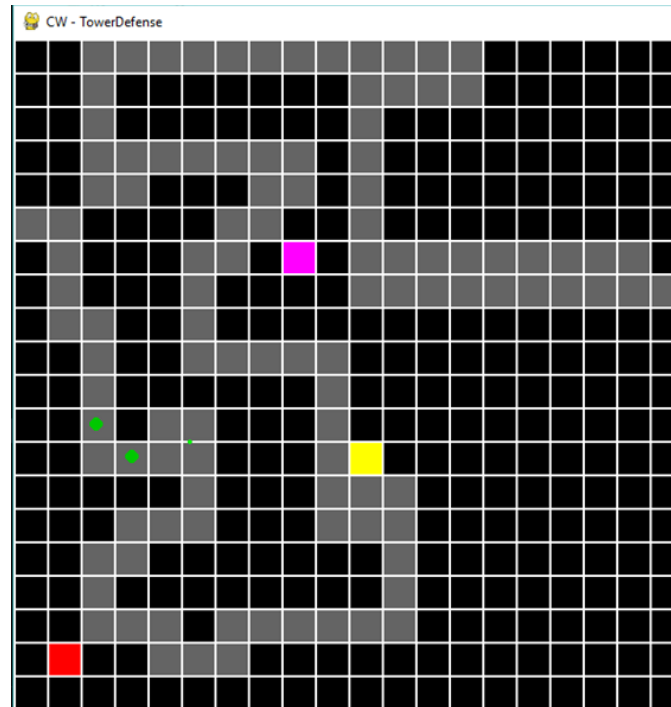
Documentation, mise en forme du code

Readme : Juliette
Tags : Juliette
Documentation des fonctions : Juliette et Sacha
Amélioration de la mise en forme du code : Alexis
PowerPoint : Amaury, Juliette, Adrien, Sacha

Améliorations : graphismes, ennemis et tours



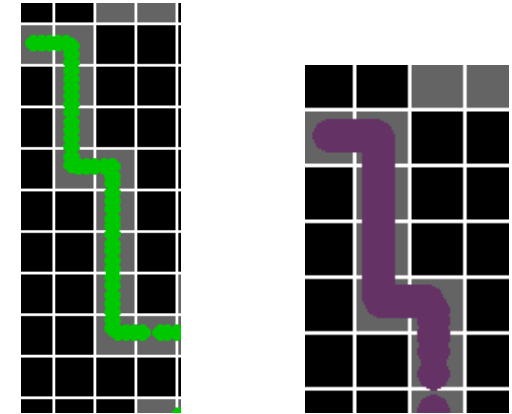
Aspect graphique des variables



Suppression d'une tour
en cliquant dessus
(rend la moitié de son prix)

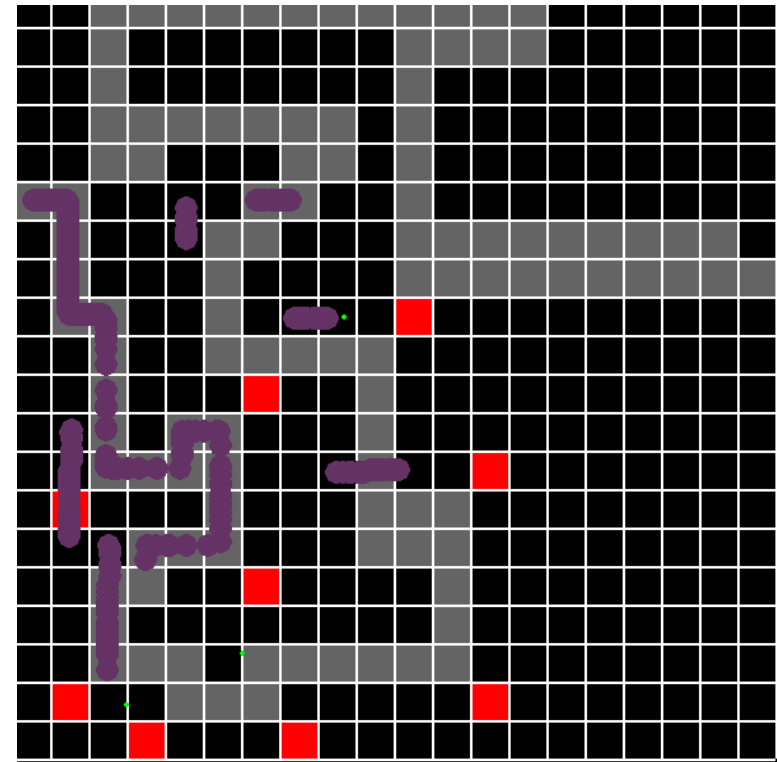
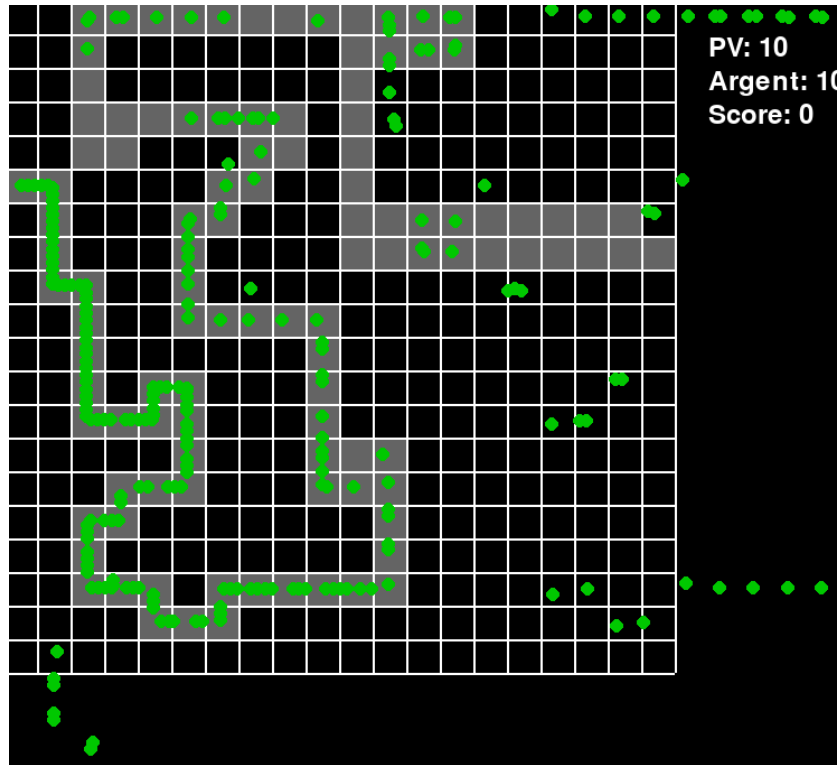
Implémentation de **plusieurs types de tour** :

- Simple
- Double (deux projectiles simultanés)
- Mitraillette (moins de délai entre 2 tirs)



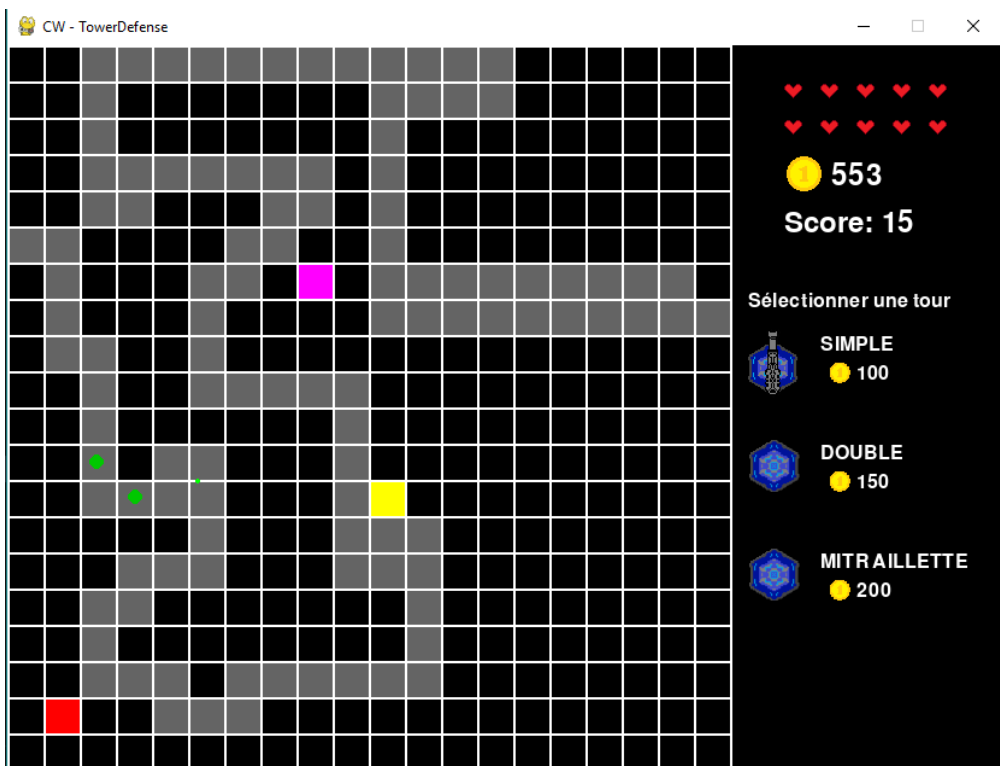
Plusieurs **types d'ennemis**
(gobelins, barbares, chevalier)

Bogues rencontrés

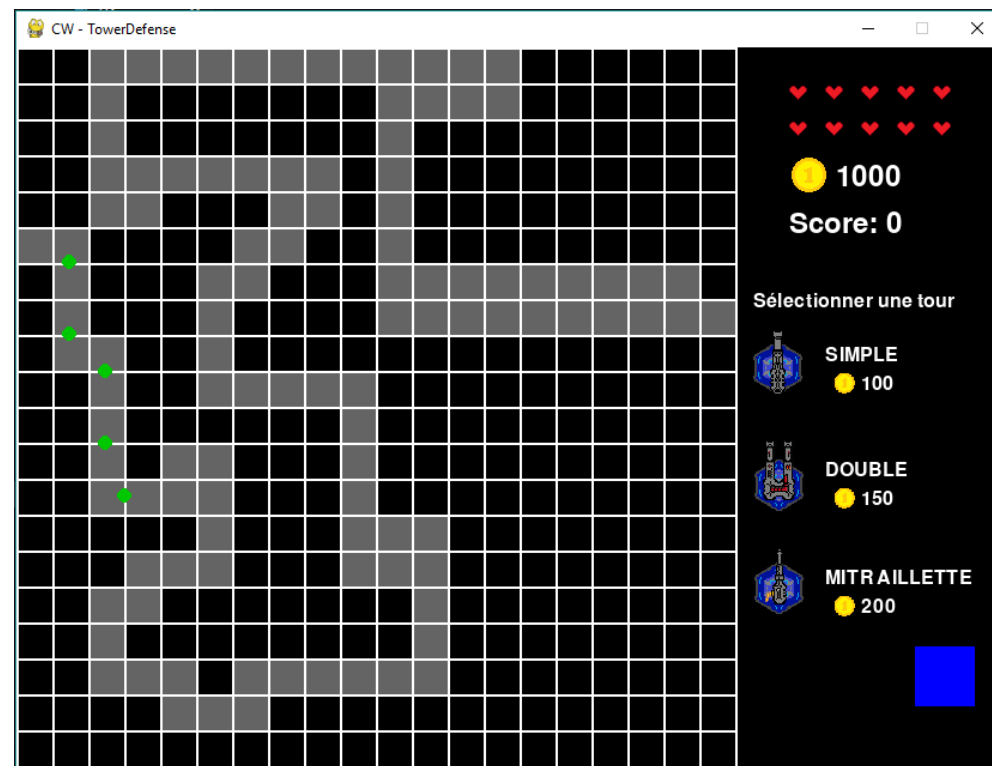


Les ennemis tendent à sortir du chemin lorsqu'ils vont trop vite (problème corrigé)

Améliorations : interface graphique

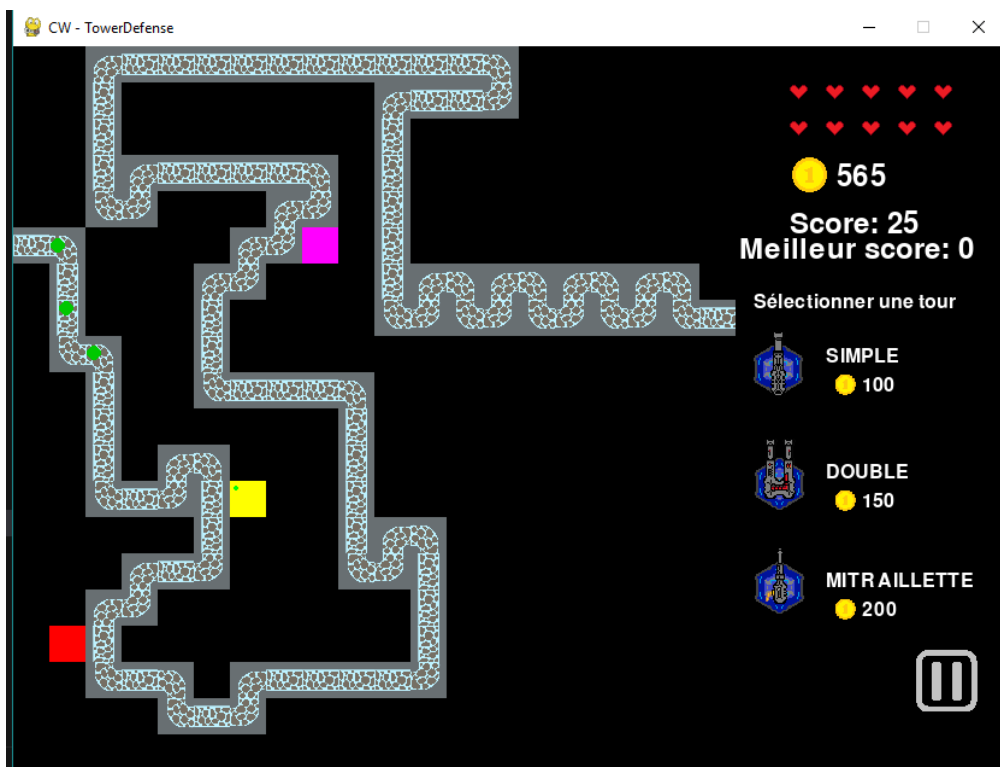


Design des tours dans le menu
Sélection du type de tours possible

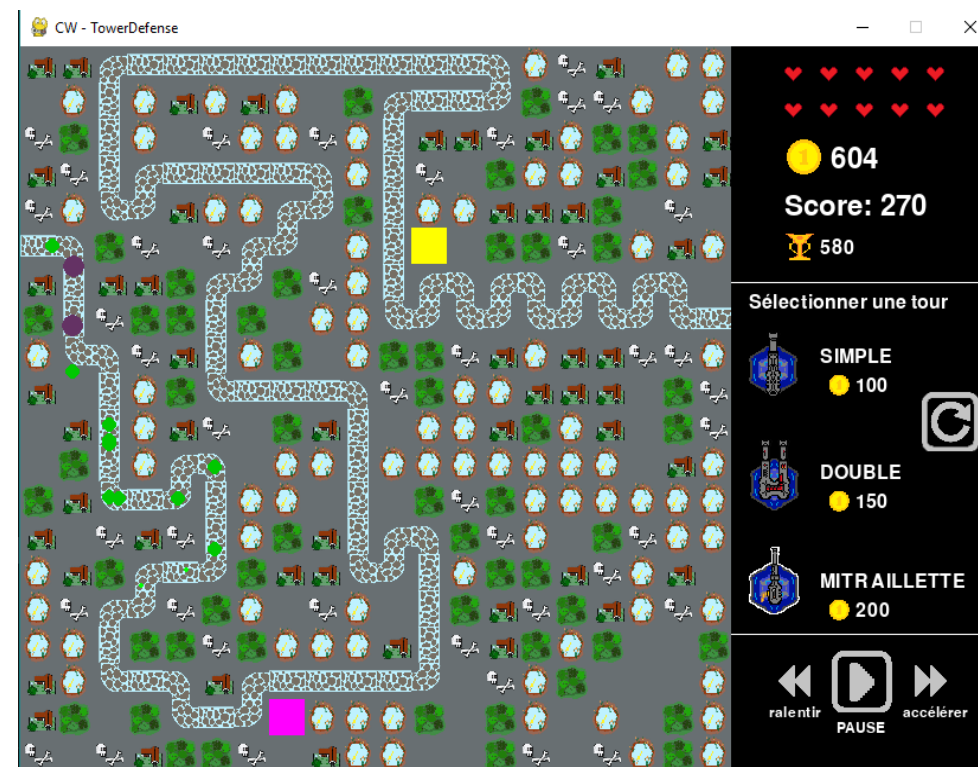


Implémentation du bouton **pause** (en bleu à droite ici)

Améliorations : interface graphique

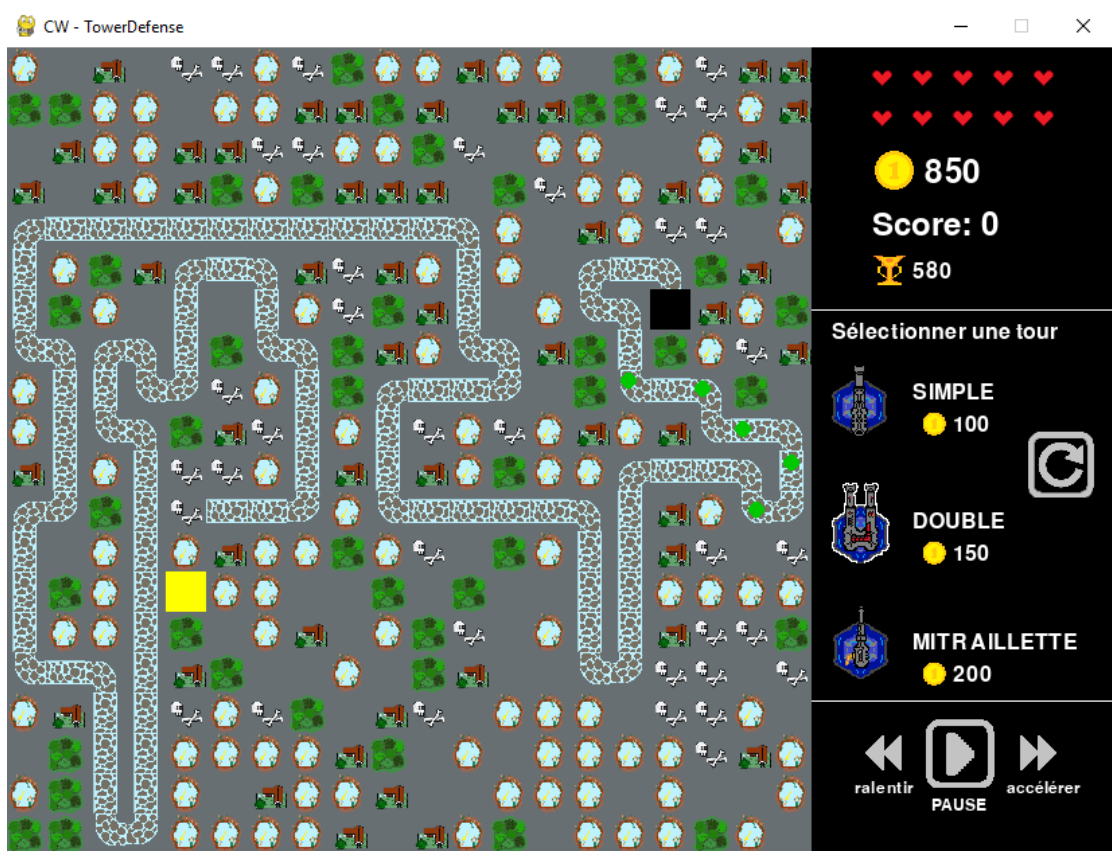


Création et orientation de la **rivière**
Ajout du **meilleur score** (qui est sauvegardé)

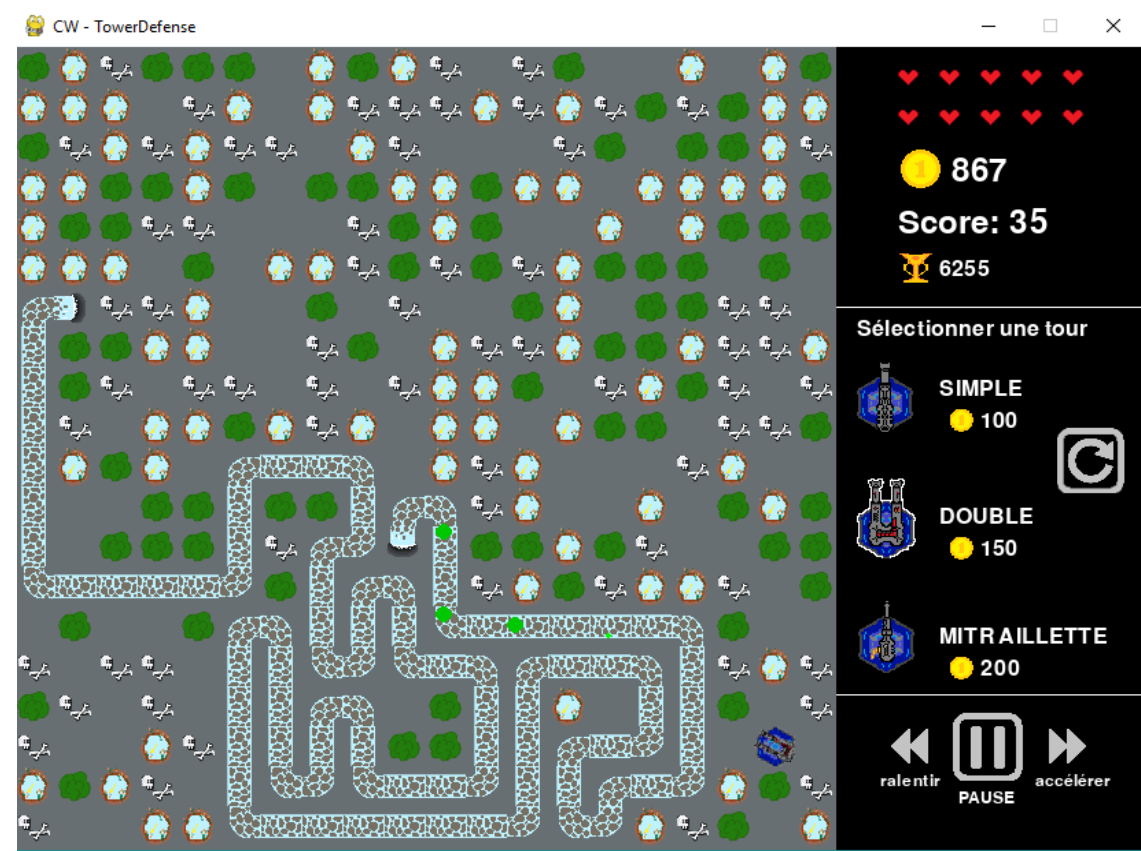


Accélérer et ralentir
Modification du design du meilleur score
Ajout des boutons à 2 états (tours et bouton pause)
Bouton **restart**

Améliorations : chemin aléatoire



Implémentation d'un **chemin aléatoire**

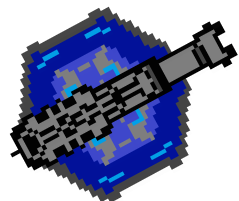


Modification du fond et ajout **tuile de fin**

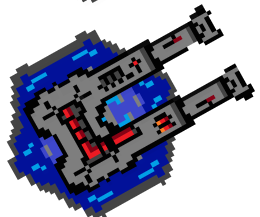
Améliorations : graphisme

Apparence des tours et des projectiles

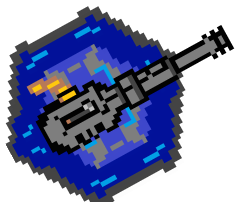
Canon orienté vers la cible



Tour simple



Tour double



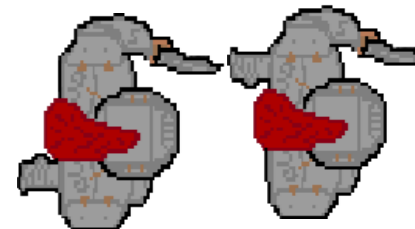
Tour mitrailleuse



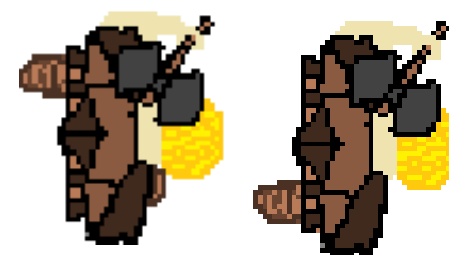
Apparence des ennemis



gobelins



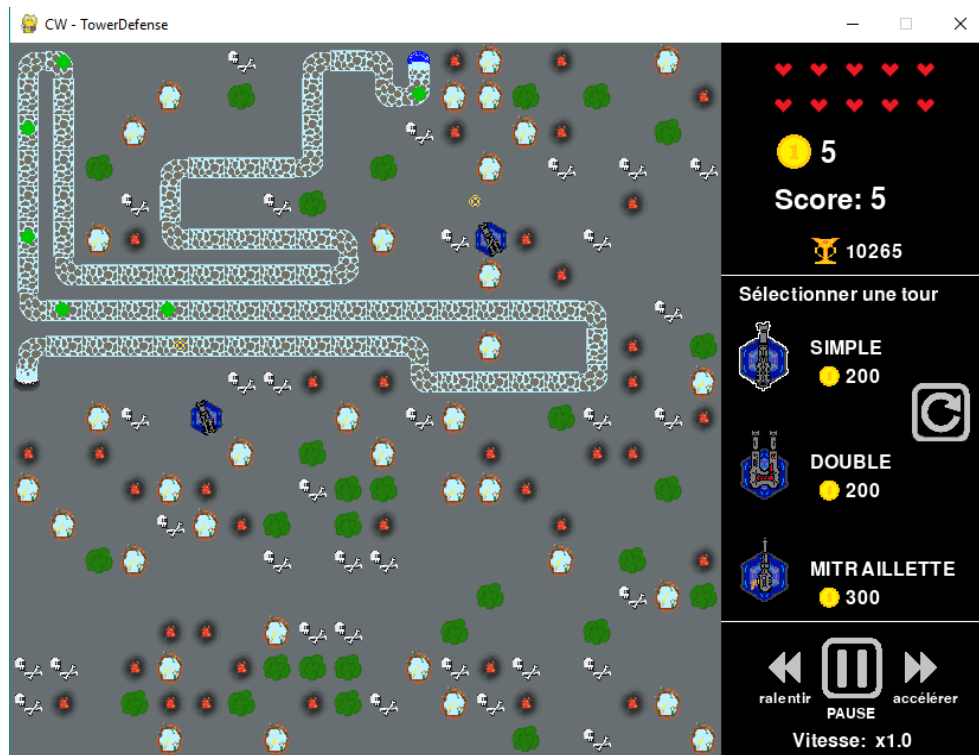
chevaliers



barbares

- Images gauches et droites => marche
- Orientation dans le sens du chemin

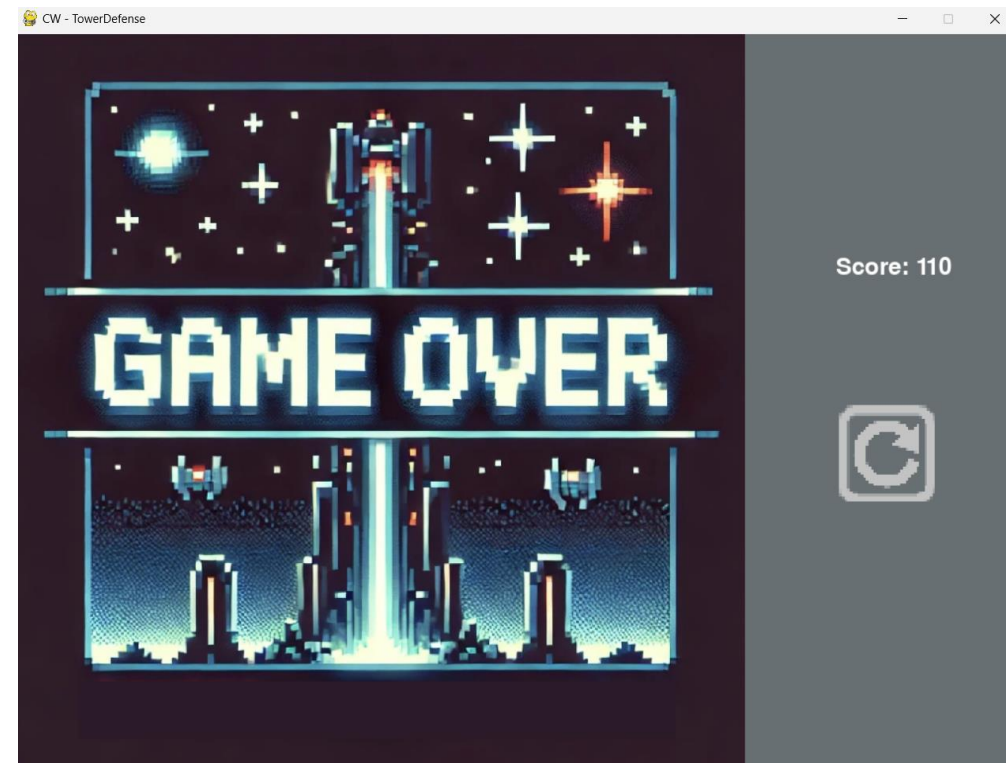
Dernières améliorations



Graphisme du **décor de fond** (avec une plus grande fréquence de gris)

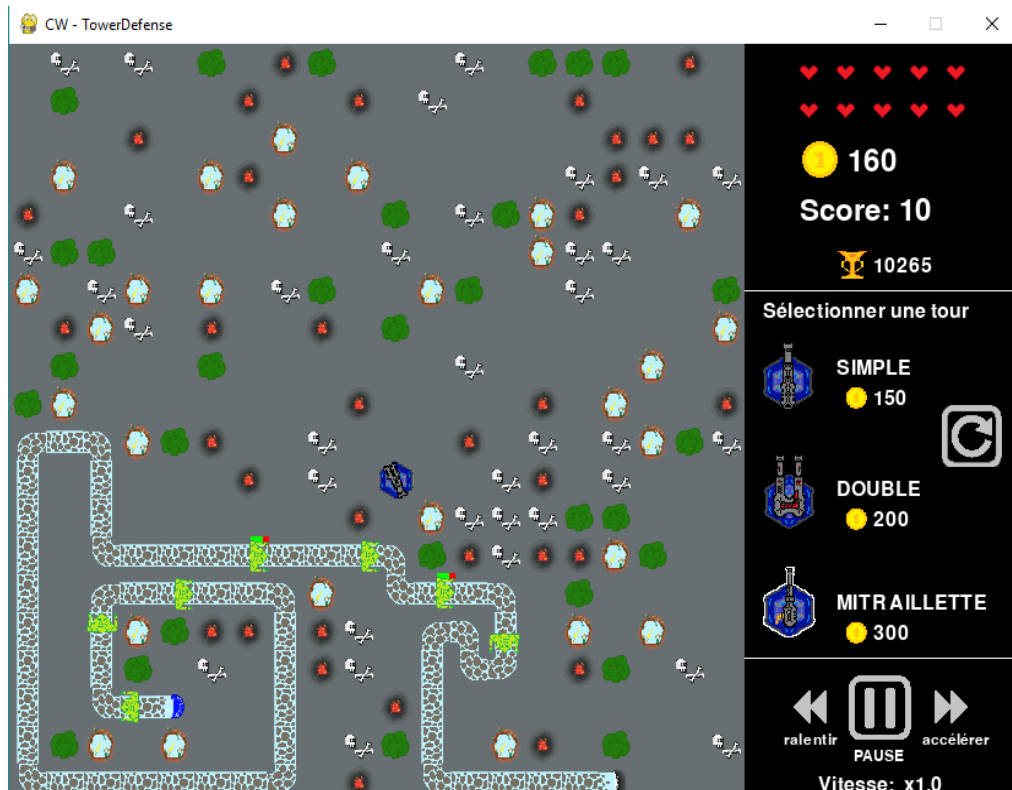
Affichage de la **vitesse**

Équilibrage

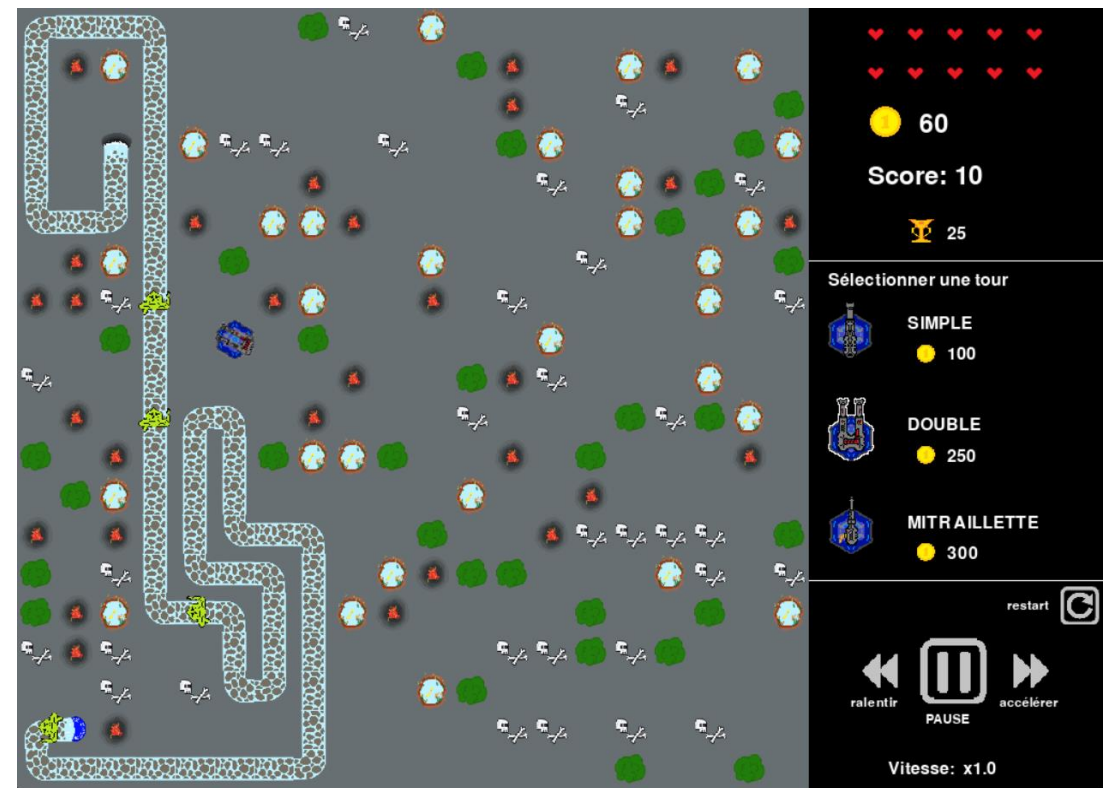


Écran de **fin de jeu** (score, bouton restart)

Dernières améliorations



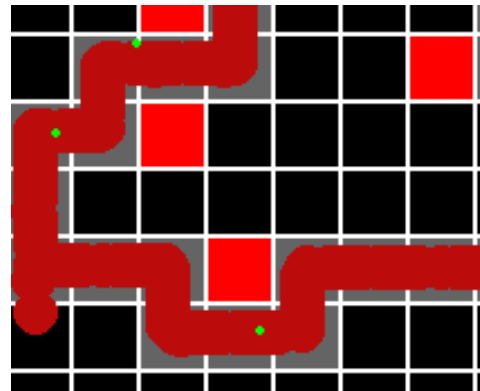
Correction du bogue du bouton pause
Ajout de la **jauge de vie** des ennemis



Agrandissement de la fenêtre de jeu
Positionnement du bouton restart

Comparatif avant/après amélioration

Tir des tours

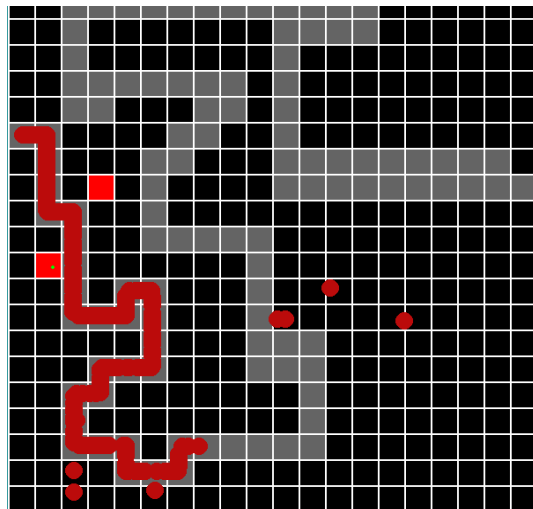


Version minimale



Version finale

Arrivée des ennemis
par vagues



Couverture de tests

Coverage report: 93%

Files

Functions

Classes

coverage.py v7.6.4, created at 2024-11-21 14:48 +0100

File ▲	statements	missing	excluded	coverage
game__init__.py	2	0	0	100%
game\bouton.py	24	0	0	100%
game\globalVariableManager.py	16	0	0	100%
game\IG.py	110	0	0	100%
game\Joueur.py	35	0	0	100%
game\object_classes.py	224	9	0	96%
game\opponentsManager.py	142	42	0	70%
game\save_manager.py	15	0	0	100%
game\sprites.py	166	1	0	99%
game\TD_game.py	160	10	0	94%
game\TD_grid.py	116	13	0	89%
Total	1010	75	0	93%

coverage.py v7.6.4, created at 2024-11-21 14:48 +0100

Améliorations possibles

Raisonnable

- Ajout de types de tours, de types d'ennemis
- Amélioration des tours après leur construction
- La fenêtre de jeu s'adapte à la taille de l'écran

Ambitieux

- Animation des projectiles et des tirs
- Structures annexes type ferme/bunker pour gagner davantage d'argent et regagner des PV
- Zone de jeu en relief (les ennemis en altitude sont inatteignables par certains tours, les tours sur les sommets ont davantage de portée)

À vous de jouer !!

Version finale du Tower defense

