

# Compte rendu

## Organisation

Le projet est composé de trois packages :

- Display (classes dédiées à l’affichage) :
  - Card (entité héritant de SelectionCard représentant une carte de plante sélectionnable dans le jeu)
  - CardList (une ArrayList de Card permettant de gérer les Card)
  - SelectionCard (carte / bouton sélectionnable du menu)
  - SelectionCardList (une ArrayList de Card permettant de gérer les SelectionCard)
  - GameView (interface dédiée à l’affichage)
  - PVZView (contient toutes les méthodes utilitaires d’affichage et de calcul par rapport aux coordonnées du plateau)
- Entities (ensemble des zombies et des plantes)
  - LivingBeing (attributs et méthodes utilitaires de tout être animé du jeu)
  - Plant & Zombies (attributs et méthodes utilitaires de toutes plantes / tous zombies du jeu)
  - Toutes les plantes du jeu (organisées par types de maps via les différents packages)
  - Tous les zombies du jeu (organisés par types de maps via les différents packages)
  - Grave (tombe spawnant des zombies disponible en mode nuit)
  - Shroom (interface représentant les plantes qui dorment le jour)
  - SunSpawner (interface représentant les plantes qui spawnent des soleils)
  - SunSpawners (classe abstraite contenant les méthodes et attributs communs des SunSpawners)
- Projectiles (gestion des projectiles)
  - Projectiles (interface représentant les projectiles)
  - Projectiles (classe abstraite représentant les projectiles)
  - ClassicalProjectile (projectile rectiligne ou incurvé)
  - BoomerangProjectile (projectile rectiligne qui fait un aller retour)
- Control (classes dédiées à la gestion des entités sur le plateau)
  - BoardGame (plateau de jeu)
  - Line (ligne du plateau)
  - Cell (case du plateau)
  - Coordinates (méthodes utilitaires sur les positions 2D)
  - ZombieGenerator (générateur de tableaux aléatoires de zombies)
  - WaveGenerator (générateur de waves de zombie grâce à ZombieGenerator)
  - Time (entité qui joue le rôle d’un chronomètre)
  - PVZGameController (boucle principale du jeu)
  - MowerList (liste de tondeuses)
  - Mower (tondeuse)
  - ProjectileList (liste de projectiles)
  - Projectile (projectile tirée par une plante)
  - SunList (liste de soleils)
  - Sun (soleil à récolter)
- Maps (classes dédiées à la gestion des types de BoardGame)
  - Map (superclasse représentant une map)

◦DayMap, FogMap, NightMap, PoolMap, RoofMap (méthodes utilitaires des maps pour l'initialisation du BoardGame)

Deux fichiers sont également présents :

- config.txt (configuration des données du jeu)
- save.txt (sauvegarde de la partie précédente)

## Boucle principale

Le jeu est conçu de manière à :

- Initialiser un BoardGame
- Initialiser la PVZView correspondant au tableau (pour gérer l'affichage)
- Initialiser la CardList correspondant au choix de cartes
- Initialiser le WaveGenerator
- Tout dessiner et gérer dans l'ordre :
  - Les waves
  - Les attaques
  - Les tombes
  - Les tondeuses
  - Les conditions de victoire / défaite
  - Les touches (lancement du mode debug / fin de partie)

## Implémentations

### Entités

Chaque entité vivante implémentée devra directement ou indirectement hériter de LivingBeing.

Chaque zombie doit hériter de la classe zombie et doit renseigner les méthodes getVictims() s'il a un mode d'attaque particulier. Sinon, il héritera de la méthode classique d'attaque. Chaque plante doit renseigner la méthode doActionIfNecessary() si elle a une attaque particulière. Sinon, la plante enverra des projectiles aux caractéristiques correspondant à son constructeur.

NB : Chaque plante lance des projectiles.

### Plateau

Chaque plateau à ajouter doit avoir une classe héritant de map. Ceci permettra l'initialisation du BoardGame. La gestion des états s'effectue par le placement de booléens dans les Cell.