

Sujet: Réaliser le jeu Plants vs. Zombies en Java

Le but de ce projet est de réaliser une version simplifiée du jeu (mobile/en ligne) Plants vs. Zombies (<https://www.ea.com/studios/popcap/plants-vs-zombies>). Il s'agit d'un jeu de *tower defense* où l'on utilise des plantes pour se défendre d'une attaque de zombies.

Le jeu¹

Voici un extrait (légèrement modifié et complété) de la description du jeu que l'on trouve sur Wikipédia : https://fr.wikipedia.org/wiki/Plantes_contre_zombies.

Le but du jeu est d'empêcher les zombies d'entrer dans votre maison pour manger votre cerveau. Pour ce faire, il faut placer sur le terrain de jeu un ensemble très varié de plantes ou de champignons. Certaines plantes crachent des pois (un par un, deux par deux ou quatre par quatre), d'autres catapultent des choux ou des grains de maïs et des mottes de beurre qui paralysent les zombies [...].

Chaque objet à placer a ses propres capacités offensives, défensives ou de production de ressources. Le joueur doit les répartir pour assurer une défense efficace tout en accumulant suffisamment de ressources, par exemple des rayons de soleil, qui tombent directement dans le jardin (ou sont produits par des plantes telles que le tournesol) pour pouvoir acheter d'autres plantes et améliorer ses défenses face à des hordes toujours plus nombreuses [...]. Les zombies rencontrant une plante sur leur chemin commencent à la manger avant de continuer. S'ils parviennent au bout de la ligne, un outil à usage unique est lancé, en général il s'agit d'une tondeuse à gazon [...]. Si les zombies parviennent malgré tout à arriver au bout d'une ligne non défendue, la partie est perdue.

Chaque terrain de jeu est composé de cinq cases verticales, ou six pour les niveaux comportant une piscine et d'une dizaine de cases horizontales chacune. On peut poser en général une plante par case. La gauche de la grille représente votre maison, et les zombies arrivent par la droite. Vous pouvez planter sur n'importe quelle case du plateau, sauf s'il y a déjà une plante. Dans le jeu officiel, les zombies se déplacent en continu, mais pour simplifier l'implémentation, dans votre projet, ils pourront avancer case par case. Les niveaux sont divisés en types de terrain:

- le jardin,
- le même jardin la nuit,
- le jardin avec une piscine au milieu,
- le toit,
- le jardin avec piscine la nuit.

Chaque type de terrain a ses particularités : les soleils étant la ressource utilisée, la nuit, cette ressource est beaucoup plus rare (il y a aussi des champignons qui donnent des soleils (champi soleil)) et il vaut mieux les utiliser car plus économiques [...]. La plupart des plantes ne peuvent être mises directement dans la piscine, il faut donc acheter un nénuphar avant de les mettre. Lorsqu'il fait nuit, de la brume apparaît sur les 4 premières colonnes de droite, empêchant de voir les zombies et les plantes se trouvant dessous. Des deux côtés du jardin, des tombes apparaissent la nuit, empêchant toute construction dessus et faisant sortir des zombies lors de l'assaut final. Sur le toit, il faut avant tout acheter des pots pour y mettre les plantes. Chaque niveau est rythmé par des vagues de zombies arrivant progressivement après un court temps de mise en place. À intervalles réguliers, une importante vague de zombies apparaît [...].

¹Il est fortement conseillé d'y jouer un peu (mais pas pendant les cours), pour bien en comprendre les mécanismes.

Les zombies existent en plusieurs types. Certains sont classiques, d'autres un peu plus résistants car se protégeant avec un cône de chantier (zombie cône) ou un seau métallique (zombie seau). Il y a des modèles ultra-résistants équipés d'une panoplie de joueur de football américain (zombie quaterback), certains creusent sous terre pour s'infiltrer derrière les lignes et détruire les défense à revers (zombie mineur). Il y a des sauteurs à la perche qui évitent ainsi les obstacles (zombie perchiste) comme les noix qui sont habituellement plantées pour faire office de mur et bloquer la progression des ennemis. D'autres apparaissent du ciel et enlèvent une plante au hasard (zombie-élastique), à moins d'être stoppés avant [...].

Chaque plante possède un mode d'action et des caractéristiques différentes, notamment :

- son **coût**: en nombre de soleils,
- son **temps de recharge**: le temps qu'il faut attendre pour pouvoir à nouveau la planter,
- sa **résistance**: la quantité de dommage qu'elle peut absorber avant de mourir,

À cela s'ajoutent des caractéristiques propres au type d'action que la plante effectue:

- ses dommages: la quantité de dommage infligée par action (tir, explosion, ...),
- sa portée: la façon dont les projectiles atteignent les zombies.
- son taux de production, si elle produit une ressource,
- ...

Le détail des caractéristiques de chaque plante peut être trouvé ici : [https://plantsvszombies.fandom.com/wiki/Plants_\(PvZ\)](https://plantsvszombies.fandom.com/wiki/Plants_(PvZ)).

Le programme à réaliser

Vous allez réaliser votre jeu en 3 phases. Chaque phase devra être terminée et fonctionnelle avant de pouvoir passer à la suivante.

Phase 1 Dans un premier temps, vous devez réaliser une version très simplifiée du jeu, avec uniquement 3 plantes possibles: *Peashooter*, *Cherry Bomb* et *Wall-nut* et 3 types de zombies: *Zombie (normal)*, *Flag Zombie* et *Conehead Zombie*. Voici les fonctionnalités que l'on attend dans ce premier prototype. L'ordre est une indication de la progression dans le développement du jeu.

1. Créer le plateau de jeu.
2. Un peu après le début de partie, faire apparaître des zombies aléatoirement sur la droite du plateau et les faire avancer en ligne vers la gauche.
3. Pouvoir choisir une plante et la rajouter sur une case du plateau (dans un premier temps, sans se soucier du coût et du temps de recharge) et la planter.
4. Prendre en compte les dégâts causés par les zombies aux plantes et le fait que les plantes ralentissent la progression des zombies. Retirer les plantes détruites.
5. Prendre en compte les dégâts causés par les plantes aux zombies et retirer les zombies détruits (vous n'êtes pas obligés d'afficher le déplacement des projectile, vous pouvez simplement afficher les dégâts causés).
6. Gérer la durée et la fin de partie.
7. Avoir un journal des événements (arrivée de zombie, ajout de plante, destruction de plante ou de zombie, ...) qui s'affiche sur le terminal.
8. Avoir un mode test qui joue aléatoirement en accéléré.

Phase 2: soleils et nouveaux terrains Après avoir fini et validé la première phase:

1. Rajouter le mécanisme des soleils: pour simplifier, ils apparaissent sur une case aléatoire et il suffit de cliquer sur la case pour les récupérer. Rajouter la jauge de soleils et la gestion du coût des plantes.
2. Rajouter le temps de recharge des plantes.
3. Rajouter les tondeuses de défense à la gauche du jardin.
4. Rajouter les autres plantes du jardin de jour (*Sunflower*, *Potato Mine*, *Snow Pea*, *Chomper* et *Repeater*) et le reste des zombies de jour.
5. Rajouter deux nouveaux types de terrain: la nuit et la piscine (mais pas les deux en même temps) ainsi que les plantes et zombies correspondants.
6. Rajouter un mécanisme de sélection du terrain par le joueur avant de commencer une partie.
7. Faire en sorte de sélectionner un sous-ensemble de zombies restreint et correspondant au type de terrain en début de partie, le montrer au joueur et offrir un menu qui propose de choisir une sélection d'un nombre restreint de plantes avant de commencer à jouer.

Phase 3: le jeu complet Lorsque la phase 2 est terminée, vous allez rajouter les fonctionnalités avancées:

1. On veut pouvoir configurer facilement le jeu: vitesse de déplacement de base, nombre de plantes sélectionnables, taux d'apparition des zombies, durée d'une partie, etc... Faire en sorte de pouvoir lire ces informations dans un fichier de configuration (à vous d'en choisir le format).
2. Faire en sorte de pouvoir mettre en pause/relancer. Faire en sorte de pouvoir sauvegarder une partie et relancer le programme à partir d'une sauvegarde.
3. Rajouter les deux derniers terrains et les plantes et zombies restants.
4. (Bonus) Rajouter le mécanisme d'engrais de Plants vs. Zombies 2.

L'interface graphique

Vous allez devoir réaliser une interface graphique simple:

- Vous devrez utiliser la bibliothèque d'interface graphique **zen** fournie avec ce sujet (fichier **zen5.jar**).
Pour ajouter un jar à un projet sous Eclipse:
 - Rajouter un dossier **lib** dans le répertoire du projet et y placer le fichier **.jar**.
 - Dans Eclipse, faire un clic droit sur le fichier **.jar** et choisir **Build Path > Add to Build Path**.
- On vous fournit également un mini-exemple (très incomplet) de code utilisant cette bibliothèque et suivant un modèle de développement classique appelé MVC (Modèle-Vue-Contrôleur) que vous verrez dans le cours d'IHM et que **vous devrez appliquer** dans votre code (même si vous faites l'affichage au terminal):
 - Une classe **SimpleGameData** est utilisée pour gérer les données du jeu (le modèle) ainsi que toutes les actions possibles, suivant les règles du jeu.
 - Une interface **GameView** implémentée par la classe **SimpleGameView** permet de gérer l'affichage graphique (la vue).
 - Le contrôleur **SimpleGameController** implante la boucle de jeu et la gestion des événements utilisateur (clics, touches, ...).

Conseils

- Pensez à programmer “objet”. Vous serez pénalisés si vous n’exploitez pas suffisamment les notions vues en cours. La hiérarchie des différentes classes de votre projet doit être soigneusement réfléchie.
- Lisez les descriptions de toutes les plantes et de tous les zombies avant de vous lancer, afin d’avoir une idée des actions possibles.
- Le sujet est évolutif, respectez bien les phases de réalisation, mais gardez à l’esprit ce que vous devrez faire dans les phases suivantes lorsque vous faites des choix d’implantation.

Consignes de rendu

- Ce projet est **à faire en binôme**. Si vous avez une bonne raison de demander une exception à cette règle, contactez votre enseignant (pivoteau@univ-mlv.fr).
- Sur e-learning, vous déposerez une archive contenant
 - **Les sources (.java) de votre projet** (dossier `src` sous Eclipse), correctement commentés. Il n’est pas nécessaire de fournir la javadoc de l’ensemble des classes et méthodes, mais on doit pouvoir comprendre aisément votre code grâce à son organisation, les noms de méthodes choisies et vos commentaires.
 - Un **fichier texte README.txt** qui explique
 - * ce qui a été implémenté,
 - * l’organisation du programme (la hiérarchie des classes implantées),
 - * les choix techniques/algorithmiques et
 - * les éventuels problèmes rencontrés.

Vous pouvez y ajouter tous les commentaires ou remarques que vous jugez nécessaires à la compréhension de votre code.

- La date limite de rendu **final** est le **10 juin 2019 à 23h55**. Passé ce délai, la note de votre projet sera 0. De plus, vous ferez deux rendus intermédiaires **obligatoires**: la phase 1 doit être terminée et déposée sur e-learning avant le **7 avril 2019 à 23h55** et la phase 2 avant le **5 mai 2019 à 23h55**.
- Des mini-soutenances seront organisées quelques jours après le rendu, pendant lesquelles vous ferez une démonstration sur une machine des salles de TP et serez interrogés sur le projet. Les deux membres du binôme doivent pouvoir répondre aux questions et leur notes peuvent être différentes en fonction de l’évaluation du travail fourni.
- Vous perdrez des points si vous ne respectez pas ces consignes.
- Si le code ne compile pas, la note de votre projet sera 0.
- Il va sans dire que si deux projets trop similaires sont rendus par 2 binômes différents, la note de ces projets sera 0.