

Plants vs Zombies vs E3FI

1. Information sur le code source donné pour le projet

Global.cs

Elle contient des informations de type général partagées par l'ensemble des objets. On y trouve par exemple : le gestionnaire de Sprites, la hauteur de la zone d'affichage, le numéro du tour courant, l'argent du joueur, une fonction de tirage aléatoire...

Coord.cs

Classe utilitaire. Elle permet de convertir les coordonnées écran en numéro de case du jeu et inversement.

Sprite.cs

Cette classe permet de gérer l'affichage d'un Sprite à l'écran par l'intermédiaire de la fonction DrawToScreen. Pour vous faciliter la vie, un repère cartésien a été choisi. La coordonnée (0,0) correspond ainsi au coin en bas à gauche de la fenêtre.

SpriteManager.cs

Ce gestionnaire de Sprites vous permet d'accéder à n'importe quel Sprite par l'intermédiaire de la fonction Get(String). Il suffit pour cela de donner son nom par le biais d'une chaîne de caractères. Le chargement des Sprites est automatique. Pour visualiser ces derniers, faites un clic droit dans l'explorateur de solutions sur la ligne « Plant vs Zombies » et choisissez Propriétés. Sélectionnez l'onglet Ressource pour voir défiler la liste de Sprites sous forme de vignettes. La transparence de fond des Sprites est gérée automatiquement par l'intermédiaire de la couleur noire (0,0,0).

MainLoop.cs

Il faut dissocier le déroulement du jeu : les actions, l'IA... de l'affichage. Dans le moteur du projet, un chronomètre lance la fonction Dolt() toutes les 50ms. Cette fonction a pour objectif de demander à toutes les entités du jeu de faire « ce qu'elles ont à faire » : avancer, tirer, exploser... Il s'agit d'un **tour** de jeu. La fonction Affichage() est par contre elle appelée par le système pour déclencher l'affichage des divers éléments du jeu. A chaque fin de tour, une demande d'affichage est envoyée. Mais cela ne présume en rien de ce qui va se passer. On peut par exemple avoir 1 seul affichage par seconde s'il y a beaucoup de calculs !! Gestion des tours et Affichage sont deux parties déconnectées.

MouseClic.cs

Fonction qui traite le clic souris sur la zone de jeu. Les coordonnées (x,y) reçus sont en coordonnées cartésiennes.

Zombie.cs

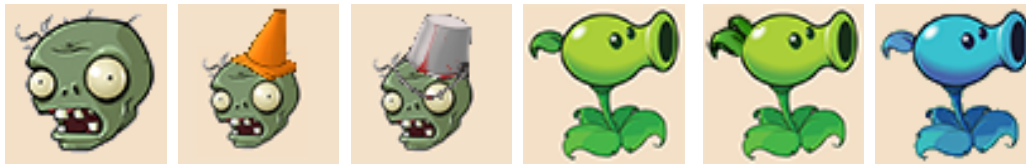
Classe fournie à titre d'exemple. Elle sera bien sur destinée à évoluer en fonction de l'ajout des fonctionnalités.

2. Différents éléments du jeu

Le plus simple est de regarder une démo du jeu pour reproduire la plupart des effets.

<http://www.popcap.com/fr/games/plants-vs-zombies/online>

Voici un résumé des caractéristiques :



Zombie	Prix : 0\$	PV : 100	Dégâts : 2 PV / tour	Vitesse : 1 pixel / tour
Zombie cône	Prix : 0\$	PV : 250	Dégâts : 2 PV / tour	Vitesse : 1 pixel / tour
Zombie seau	Prix : 0\$	PV : 500	Dégâts : 2 PV / tour	Vitesse : 1 pixel / tour

Pisto-pois	Prix : 100\$	PV : 100		
Tir	Dégât : 15 PV	Recharge : 40 tours		Vitesse : 12 pixels / s
Pois Geleur	Prix : 175\$	PV : 100		
Gel	Dégât : 5 PV	Recharge : 40 tours		Effet : 80 tours
		Divise la vitesse du zombie par 4		
Double Pisto	Prix : 200\$	Recharge : 20 tours		



Tournesol	Prix 50\$	Recharge : 300 TOURS	
Noix	Prix : 50\$	PV : 2000	
Mine	Prix : 25\$	Temps d'enclenchement : 100 tours	
	Dégât : 1000	Effet : sur 1 zombie	
Bombe Cerise	Prix : 150\$	Dégât : 500	Zone : distance 90 pixels
Canon	Prix : 500\$	PV : 100	
Tir	Dégât : 250	Recharge : 200 tours	Vitesse : 80 pix / s
	Zone d'effet : 90 pixels		
Soleil	Rapporte : 50\$		
	Génération : 80 tours		

Vous devez effectuer les étapes dans l'ordre. Vous ne pouvez passer à l'étape suivante que si vous avez entièrement terminé l'étape précédente.

Etape 1 : Soleils

Soleil

Dans le jeu, des soleils apparaissent en haut de l'écran et descendent verticalement pour finir poser sur une case. Créez une nouvelle classe Spawner. Elle n'a pas d'affichage à l'écran. Son rôle est de créer tous les 80 tours de jeu un « objet » soleil, de tirer aléatoirement sa position en X et sa hauteur finale en Y. Créez une nouvelle classe Soleil. Sa vitesse est de 1 pixel par seconde. Une fois le soleil créé, il vit au maximum 500 tours avant de disparaître.

Récolte des soleils

Le clic souris sur un soleil provoque sa disparition et l'augmentation du nombre de crédits de 50\$. Nous vous conseillons de créer une fonction qui pour un rectangle donné indique si un point est à l'intérieur ou non.

Etape 2 : Premiers éléments du jeu

Points de vie

Le niveau de vie est affiché au-dessus des éléments du jeu. Il correspond à une jauge rectangulaire constituée d'un fond noir de 100 pixels de large et d'un niveau en bleu dont la largeur

en pixel correspond au niveau de vie. On veillera à ce que l’affichage se produise au-dessus du personnage.

Pisto-Pois

Créez la classe Pisto-Pois. Un Pisto-Pois sera créé dans le jeu suite à un clic souris sur une case du décor. La case doit bien sûr être disponible. Suite à la création, le prix de la plante doit être retiré de la bourse du joueur.

Spawner Zombie

Créez une nouvelle classe dans le jeu servant à générer aléatoirement 1 zombie tous les 40 tours. Une fois le zombie créé, il se met à avancer vers la gauche.

Etape 3 : Premiers échanges

Tir

Chaque Pisto-Pois produit un tir tous les 40 tours si un zombie est présent sur sa rangée et se place devant lui. Un tir est un nouvel élément du jeu. Le tir naît en avant de la plante à hauteur de sa bouche.

Collision et dégâts

Le tir teste la collision avec un zombie en comparant les coordonnées de ces deux éléments. S’il y a collision, les points de vie sont retirés du zombie et le tir « disparaît ». Le zombie qui subit des dégâts s’affiche momentanément en mode « touché », Sprite identique mais légèrement plus clair pendant 1 tour.

Dead Zombie

Lorsqu’un zombie a un nombre de points de vie négatif, il passe en mode cadavre. Il ne bouge plus. Son cadavre est affiché sur le sol pendant 100 tours avant de disparaître. Pour simplifier la programmation, il peut être utile de créer une classe cadavre. Attention, les tirs des Pisto-Pois ne collisionnent pas avec un cadavre !!

Ordre d’affichage

Peut-être vous en êtes-vous aperçu, mais les tirs sont affichés dans le jeu original en surimpression (un tir n’est jamais masqué par une plante amie). Modifiez la routine d’affichage pour qu’elle s’effectue en 2 passes. D’abord les plantes et zombies ensuite les tirs.

Collision entre amis et ennemis

Il existe trois types d’entité, neutre, ami, ennemi. Un objet neutre n’interagit pas avec les autres : soleil, cadavres.... Les tirs n’interagissent qu’avec les ennemis et les zombies n’aiment croquer que les amis. Choisissez une manière de gérer ce problème.

Etape 4 : Sport de contact

Attaque

Lorsqu'un zombie rencontre une plante, il s'arrête et passe en mode combat. Il lui porte alors un certain nombre de points de dégâts. Il faut caler vos éléments correctement. En effet, les tirs du Pisto-pois doivent collisionner avec le zombie qui l'attaque afin qu'il se défende. Si vos tirs sont mal réglés, ils peuvent apparaître derrière lui.

Mort des plantes

Lorsqu'une plante a un nombre de points de vie négatif, elle disparaît du jeu.

Etape 5 : Enfin riche !

Tournesol

Il est temps de créer cette plante célèbre du jeu ! Petit soucis.... Elle va générer des soleils. Pour la beauté du jeu, lorsque le tournesol crée un soleil, celui-ci saute en l'air de 100 pixels et retombe au pied de la plante. Comme les autres soleils, il vit 500 tours.

Déplacement

Il faut penser à remodeler la conception du jeu à ce niveau. Créez une classe `Deplacement`. Elle a pour objectif de gérer le déplacement de chaque entité. On peut ainsi avoir une hiérarchie : déplacement rectiligne, rectiligne avec arrêt, rebond et statique, statique. A terme, chaque élément du jeu appelle son objet `Deplacement` à chaque tour pour lui demander de gérer son avancée.

Etape 6 : Le plat de résistance

Noix

Développez la classe noix, attention, son apparence change en fonction du nombre de PV restants.

Zombiesss

Développez les zombies-seau et les zombies-cône. Modifiez le `Spawner zombie` pour qu'il en crée aléatoirement.

Etape 7 : Objets trouvés

Remodélisation

En quoi les classes canon, zombies, noix et autres sont-elles différentes ? Par leur méthode d'affichage et leur IA sans aucun doute....

Hiérarchie d'affichage

Créez une hiérarchie pour l'affichage et agrégez un objet Affichage aux entités du jeu. Certains objets n'ont pas d'affichage, mais cela est aussi un choix d'affichage ☺

Hiérarchie d'IA

Le Pisto-pois, le doubles Pisto-pois et le Pois geleur partagent la même routine. Les zombies semblent tous avoir la même cervelle eux ! Créez une hiérarchie pour l'IA et agrégez un objet IA aux entités du jeu.

Etape 8 : Au choix

Double Pisto-Pois

Pas grand-chose de neuf, mais il tire deux fois plus vite !

Bombe cerise

La bombe cerise met 80 tours avant d'exploser. Elle n'est pas attaquable. L'explosion touche tous les zombies à une certaine distance.

Canon

Lent à la recharge mais efficace au tir. Une variation du Pisto-pois. Cependant, son tir affecte tous les zombies du jeu à une certaines distances en pixels. Son tir produit donc un effet différent. Il est de plus déclenché par le joueur, par un clic souris sur le canon.

Mine Patate

Un objet qui change d'IA en cours de route. Est-ce si compliqué. Peut-être que non. Que se passe-t-il si l'on considère que la mine enterrée et la mine active sont deux objets différents. La mine enterrée ne subit pas de dégât.

Tondeuse

Lorsqu'un zombie arrive devant une tondeuse, cette dernière démarre et roule en écrasant tous les zombos !!! Une tondeuse est-elle vraiment différente d'un tir !!!

Pois gel

Une entité qui modifie le comportement d'une autre... De plus lorsque le zombie est gelé, donc ralenti, une zone de glace apparaît sous ses pieds (icône gel).