Maxime Brassard, Gabriel genest et Nathan gagnon

Programmation

420-204-RE, gr.00001

Dossier de conception

**Les Indécis**

Travail présenté à

M. Jocelyn Goulet

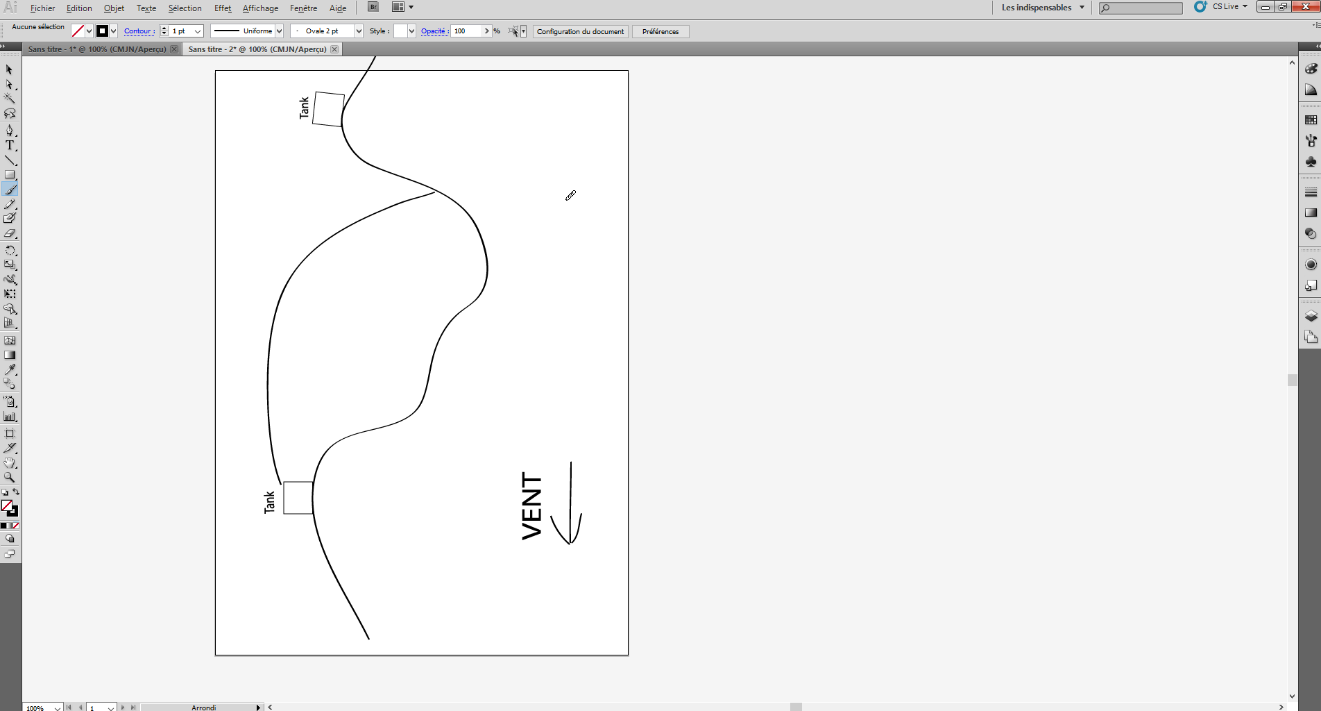
Département de l’informatique

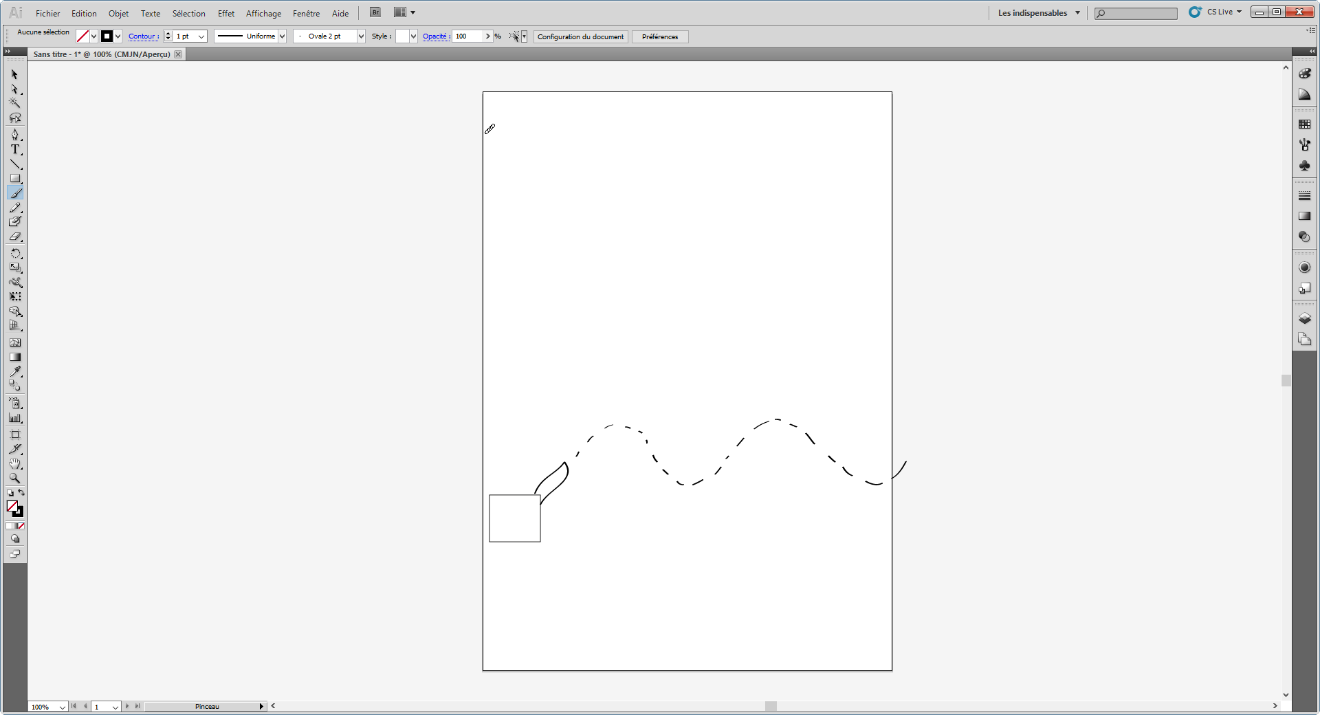
Cégep Limoilou

Le 9 février 2017

## Description du projet :

L’objectif de notre projet est de s’amuser tout en comprenant ce que sont les impacts des variations d’environnements sur des projectiles et d’expérimenter le développement progressif de la difficulté de l’IA.  
  
Le but du jeu est de simplement détruire le tank du joueur adverse. Le jeu se passera en temps continu. Tout d’abord, l’utilisateur pourra naviguer par l’écran d’accueil à l’écran d’option, de jeu ou quitter le programme. L’écran d’option permettra à l’utilisateur de contrôler les variables du jeu tels que la difficulté de l’IA, la création du terrain et de son type, ainsi que le vent. La partie intéressante du programme passe nécessairement par les tirs.

Il y aura deux types de projectiles dans notre jeu soit les projectiles lancé par des armes physiques ou d’énergie. Le premier type de projectiles explosent à l’impact et qui par conséquent, inflige des dommages à l’ennemie et une altération du terrain. Cette catégorie est influencée par la gravité :



et la résistance du vent :

Ce type d’arme contient plusieurs variantes tels que :  
Un tir de projectile normal,  
Un tir de projectile avec ‘x’ rebond avant l’explosion. Ces rebonds seront influencé par le type du sol avec un coefficient de restitution arbitraire qui respecterait la formule de collision inélastique :   
Un tir de projectile avec fragmentation (division de plusieurs projectiles normal) dans les airs.  
Un tir de projectile avec fragmentation lors de la première collision.

Pour le deuxième type d’arme présenté, ceux-ci ne seront pas influencé par la gravité et le vent mais seront plutôt déterminé par une fonction mathématique que l’utilisateur devras manipuler pour avoir les bons paramètres afin de toucher l’ennemie et de lui infliger des dommages. Ce type de projectile ne cause pas la détérioration du terrain même que l’avantage est qu’il ne sont pas influencé par la collision du terrain. En effet l’animation envisagé serait plutôt un type de laser étroit suivant la fonction demandé et infligeant des dommages sur tous les points de la fonction. Les fonctions qui seras accessible sont :

Notre programme sera en mesure de bien gérer les différents projectiles en mouvement ainsi que les tanks. Pour facilité la jouabilité, l’utilisateur pourras observer la trajectoire du tir qu’il prépare par une ligne pointillé mais cette ligne serait calculé a priori de l’influences des forces da la résistance du vent et de la gravité.

L’une des difficultés du programme sera la conception d’un terrain formé aléatoirement mais de façons contrôlé pour le plaisir des utilisateurs. Le procédé que nous utiliserons seras de former une courbe de fonction trigonométrique (pour de belles montagnes) de manière à ce que l’amplification des paramètres modifierais l’amplitude globale des montagnes, rendant le terrain beaucoup plus accidenté. Il reste un ambigüité face à la méthode utilisé pour la reconnaissance de collision sur un tel terrain qui doit être finalement dynamique puisqu’il se fera peu à peu dégrader par les explosions des projectiles. L’une des façons serait de procéder par le changement de couleur du décor, ou bien par certains objets graphiques. Bien évidemment l’orientation des tanks serait déterminée par la tangente de la partie discontinue de points en dessous de lui. De plus, le mouvement des tanks serait influencé par le type de terrain (comme le rebond des projectiles) qui serait caractérisé comme suit : Gazon, Boue, Asphalte.

Une grande partie de la difficulté qui sera rencontrée vient du fait de trouver un IA performant et évolutif. Nos algorithmes se baseront sur des théories déjà établies tels que le pathfinding pour cibler l’optimisation d’un tir et aussi par des théories d’algorithme évolutionnistes pour observer une amélioration des choix d’anticipation du vent et de l’esquive des projectiles ennemis de manière sensible au cours de la partie. De la recherche plus approfondie reste à faire pour bien cibler les algorithmes cherchés d’intelligence artificielle évolutif avec le temps.

Sur la partie plus artistique, notre jeu sera accompagné par des musiques d’ambiances et d’animations dynamiques.

Distribution des rôles : - Chef : Nathan

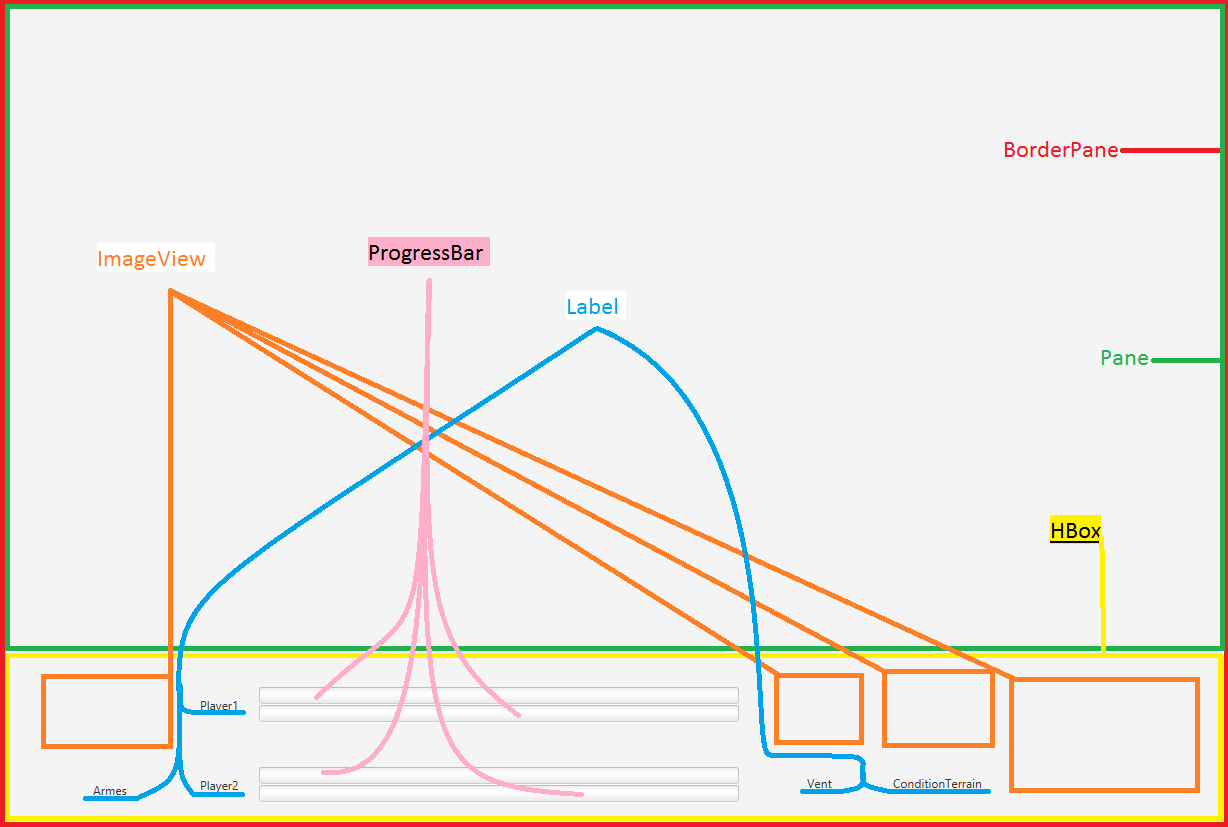
-Contrôle qualité : Gabriel

-Secrétaire : Maxime

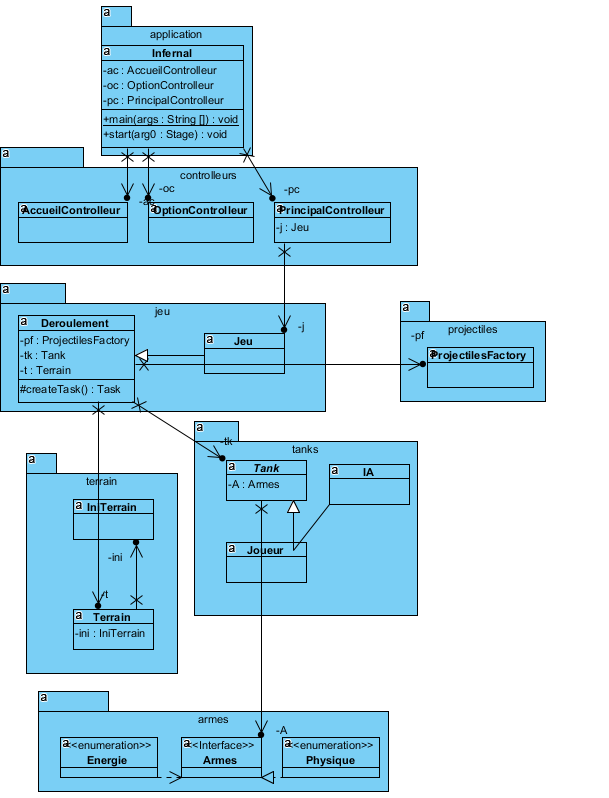
Type d’application : Windows

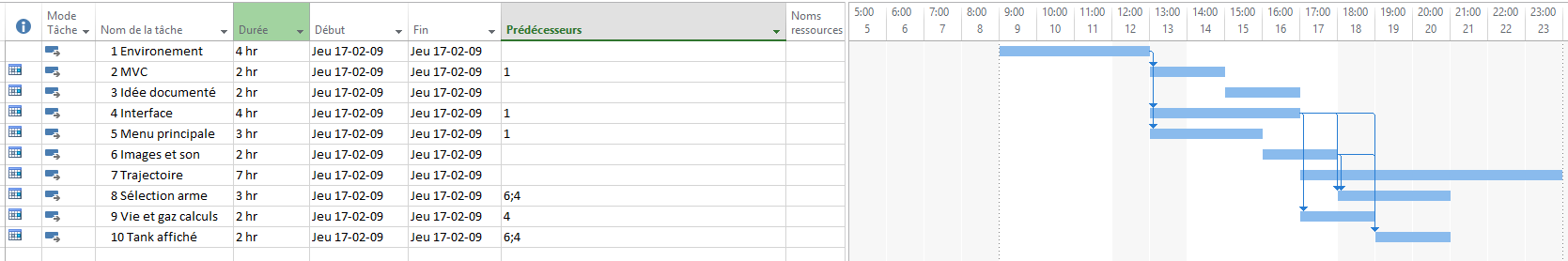
Langage : Java, CSS

Technologies impliquées : Il y avait possibilité d’utiliser les manettes d’xbox 360 mais le temps impliqué pour cette réalisation qui rembourre le projet ne serait que réellement conçus si le projet avance bien et que la librairie n’est pas trop difficile à utiliser.



## UML du projet :





## User Stories :

**1**

En tant que : Développeur

Je veux que : nous ayons un environnement de développement stable et efficace.

Afin que : -Nous ayons un environnement qui est propice au développement de l’application, que l’on soit au cégep ou non.

-Nous ayons un environnement sur lequel on peut travailler en même temps sans que les fichiers ne soient écrasés à chaque modification (GitHub)

-Nous ayons un moyen de communication rapide et efficace lorsqu’on n’est pas au même endroit.

-Nous ayons un environnement de développement Java avec *Eclipse*.

-Nous ayons

Test acceptation : Essayer, chacun de notre côté, de modifié un fichier à l’extérieur du cégep et vérifier si le fichier reste propre et ne se fasse pas écraser à chaque changement.

Complexité : 1

Effort : 1 (Fibonacci)

**2**

En tant que : développeur

Je veux que : nous ayons un packaging qui suit les règles du MVC

Afin que : -Nous ayons de l’ordre dans les classes et packages

-Nous ayons une organisation MVC pour qu’on n’ait pas à y retoucher en fin de projet

Test acceptation : Avoir une structure de package qui est divisé en packages généraux

Complexité : 3

Effort : 2 (Fibonacci)

**3**

En tant que : développeur

Je veux que : nous ayons une idée générale et documentée de ce que la hiérarchie des classes et packages doit être

Afin que : -Nous ayons une vision à moyen-long terme de comment structurer nos classes et les différents packages

Test acceptation : Avoir une hiérarchie qui convient à tous dans sa simplicité et dans son organisation cohérente.

Complexité : 3

Effort : 2 (Fibonacci)

**4**

En tant que : développeur

Je veux que : nous ayons une interface de jeu qui s’affiche sans problème

Afin que : -Nous ayons une interface de jeu qui s’affiche sur laquelle on peut faire des tests si besoin

-Nous ayons une interface du jeu qui permet d’avoir une vue d’ensemble et pouvoir déterminer des proportions convenable

Test acceptation : Faire afficher l’interface avec tous les éléments de base

Complexité : 3

Effort : 3 (Fibonacci)

**5**

En tant que : développeur

Je veux que : nous ayons une interface qui s’affiche pour le menu principal

Afin que : -Nous ayons une interface pour permettre d’entrer dans le jeu quand on est prêt

-Nous ayons une interface pour entrer dans un menu d’option

Test d’Acceptation : Faire afficher l’interface avec les boutons qui s’affichent

Complexité : 3

Effort : 3 (Fibonacci)

**6**

En tant que : développeur

Je veux que : nous ayons les prototypes des images et sons utilisés dans le jeu

Afin que : -Nous ayons des images de tanks

-Nous ayons de la musique

-Nous ayons des fonds d’écrans

-Nous ayons ces outils a porté de main sans avoir à les chercher à la dernière minute

Test acceptation : Avoir au moins une image de tanks et deux musiques (une pour le menu, l’autre pour le jeu)

Complexité : 2

Effort : 1 (Fibonacci)

**7**

En tant que : développeur

Je veux que : nous ayons une classe qui s’occupe de la gestion de la trajectoire des particules

Afin que : -Nous ayons des algorithmes de calcul des trajectoires selon des paramètres externes (gravité, vent)

-Nous ayons une classe qui peut être utilisée pour déterminer le point d’impact d’un projectile

Test acceptation : Être capable de calculer une trajectoire précise sans erreur

Complexité : 6

Effort : 5 (Fibonacci)

**8**

En tant que : Utilisateur

Je veux que : nous puissions sélectionner une arme

Afin que : -Nous puissions sélectionner l’arme avec la molette de la souris

* Le joueur puisse choisir son arme

Test d’acceptation : Être capable de choisir son arme en utilisant la molette. Chaque tic de la molette change d’une arme

Complexité : 3.5

Effort : 3 (Fibonacci)

**9**

En tant que : Développeur

Je veux que : Les opérations possibles sur les barres de vie et de gaz soient préparés

Afin que : -Cela soit facile, lorsque nous aurons à programmer les mécaniques du jeu de modifier ces barres

Test d’acceptation : Les tests faits sur les barres pour vérifier si les données restent cohérentes passent

Complexité : 2

Effort : 2 (Fibonacci)

**10**

En tant que : Développeur

Je veux que : Mon tank s’affiche sur l’écran

Afin que : Nous puissions faire des manipulations qui le nécessite (tirs, etc.)

Test d’acceptation : Test visuel (On voit le tank s’afficher)

Complexité : 2

Effort : 2 (Fibonacci)

## Références et documentation :

https://fr.wikipedia.org/wiki/Chute\_avec\_r%C3%A9sistance\_de\_l%27air

<https://fr.wikipedia.org/wiki/Collision_in%C3%A9lastique>

<https://fr.wikipedia.org/wiki/Algorithme_%C3%A9volutionniste#Principes_de_bases>

<https://fr.wikipedia.org/wiki/Recherche_de_chemin>