

## **PROJET : Bataille en forêt**

L'objectif du projet consiste à réaliser un jeu de bataille en forêt et de le rendre convivial via une interface graphique.

Deux joueurs s'affrontent dans une forêt hostile remplie d'obstacles en tous genres. Chaque joueur dispose d'une arme d'attaque avec des caractéristiques précises et cherche à éliminer son adversaire en le mettant hors jeu. Le joueur peut être aussi mis hors jeu par l'épuisement de ses forces.

Le projet va se dérouler en trois grandes étapes, certaines étant relativement indépendantes.

### **ETAPE 1 : Réalisation du monde virtuel**

Les joueurs évoluent dans une forêt hostile, remplie d'arbres, de rochers..., de formes et de tailles différentes. La première étape consiste tout d'abord à définir un catalogue d'objets, le maître de jeu pourra ainsi créer son monde virtuel à partir de celui-ci. Afin d'offrir plusieurs parties variées, on doit pouvoir sauvegarder et relire ce catalogue, ainsi que les différents mondes virtuels créés par le/les maîtres du jeu, ceci afin de gagner du temps et de rendre le jeu plus agréable.

La forêt (monde virtuel) est décrite par un terrain en 2 dimensions dont les côtés sont en nombre de pixels. Chaque obstacle sur le terrain possède une position  $(x,y)$  relative par rapport au centre de la forêt, un diamètre en nombre de pixels et une hauteur en nombre de pixels. Plus l'obstacle est imposant, plus il est difficile de le détruire.

Au début de la partie, chaque obstacle possède une force proportionnelle à sa taille, qu'on vous laisse le soin de définir. Quand l'obstacle est touché par un joueur sa force diminue de façon également proportionnelle à la force de l'arme qui l'a touché. Quand sa force arrive à zéro, l'obstacle disparaît de la partie.

Dans un premier temps, on vous demande de démontrer et valider toutes ces fonctionnalités sans interface graphique, puis dans un second temps, on vous demande d'offrir une interface graphique permettant aux joueurs de visualiser la forêt hostile qu'ils auront choisie dans une liste possible afin de démarrer une partie. (NB : l'affichage se faisant dans un plan, il est impossible pour un joueur d'estimer la hauteur des obstacles).

### **ETAPE 2 : Les joueurs**

Quand une partie démarre plusieurs modes de jeu sont offerts :

1. ordinateur contre ordinateur
2. humain contre ordinateur
3. humain contre humain

La différence entre un humain et l'ordinateur est le type des données. Celles de l'ordinateur sont générées de façon aléatoire, alors que les données du joueur sont données au fur et à mesure. Dans la suite du texte, pour simplifier le discours, nous parlerons des joueurs qui peuvent représenter soit un ordinateur, soit un humain.

Dans une partie, il y a toujours une alternance de jeu entre 2 joueurs. Au début de la partie, une position aléatoire dans la forêt est affectée au joueur. Une position regroupe :

- des coordonnées  $(x,y)$  relatives au centre de la forêt et valides par rapport à la taille de la forêt. Attention, aucun joueur ne peut s'accrocher aux arbres, ni reposer sur un rocher. Bien veiller à le positionner au sol.
- une orientation (de  $0^\circ$  à  $360^\circ$ ) indique dans quelle direction le joueur regarde.  $0^\circ$ , il regarde le centre de la forêt, ensuite les degrés augmentent lorsqu'il tourne dans le sens des aiguilles d'une montre.

Le déplacement d'un joueur dépend de son orientation, d'un nombre de pixels égal à 3 au maximum. Avant chaque déplacement, le joueur peut changer d'orientation. Attention, un joueur ne peut pas traverser les obstacles. Si un obstacle est sur sa route, il reste coincé derrière.

Chaque joueur possède une arme d'attaque qui lui permet de tirer sur l'adversaire. Lorsque le joueur tire, si la trajectoire (qu'il faudra calculer) rencontre un obstacle ou plusieurs obstacles, chaque obstacle perd de sa force. Plus la force de frappe est élevée, plus les dégâts au sol sont importants. Une frappe de 1 correspond à 1 pixel de diamètre, de 2 à 2, etc...

Un tir est caractérisé par :

- un angle de tir ( $0^\circ$  à  $90^\circ$ ), que le joueur peut ajuster avant de tirer
- une portée de frappe (un nombre de pixels, au maximum 5)
- une force de frappe entre 1 à 5.

Une fois un déplacement effectué, le joueur peut choisir ou pas de tirer. Ne pas tirer revient à économiser des points de vie. En effet, chaque joueur possède des points de vie attribués au début de la partie. A chaque fois qu'il tire, son nombre de points de vie diminue et plus la force de frappe est élevée, plus il perd des points de vie. Lorsqu'il est touché par le tir d'un autre joueur, il est éliminé de la partie et le jeu s'arrête.

Attention, si le joueur n'a plus assez de points de vie, il ne peut plus tirer.

Dans un premier temps, vous devez mettre en place toutes les informations relatives à un joueur sans interface graphique. Attention à ne pas oublier les 2 modes possibles d'un joueur (ordinateur ou humain), et bien saisir les coordonnées au clavier pour l'humain. Dans un second temps, vous devrez simuler sur une carte (monde virtuel-forêt) les dégâts faits par un seul joueur, en tenant compte du : (1) changement d'orientation (optionnel), (2) déplacement (fixer les paramètres), (3) tir (optionnel). Si le tir a lieu, vérifier que le nombre de points de vie est suffisant et calculer les dégâts. On devrait voir à l'écran la disparition des obstacles.

### **ETAPE 3 : Une vraie partie**

Maintenant il est temps d'assembler le tout via une interface graphique. Tout d'abord, il faut choisir un monde virtuel disponible, choisir le mode de jeu de la partie comme décrit plus haut qui va déterminer le mode de fonctionnement du joueur (aléatoire ou manuel). Par défaut, aucun joueur ne voit la position de l'autre car la plupart du temps les joueurs sont cachés par des obstacles. Mais plus la partie avance, plus les obstacles disparaissent et il devient peut-être possible de voir l'adversaire. Par exemple, si je trace une ligne entre un joueur et son adversaire et qu'il n'y a plus aucun obstacle alors les deux joueurs apparaissent en même temps sur la carte. Même si je vois le joueur je peux ne pas être en capacité de l'atteindre (plus assez de points de vie, ou il est trop éloigné pour être touché par un tir). Mais dès qu'un joueur bouge, il peut à nouveau se rendre invisible. Cela correspond un peu au jeu du chat et de la souris.

**Remarques :**

Pour l'interface graphique vous pouvez utiliser la bibliothèque graphique de votre choix. Il existe par exemple la bibliothèque QT (Lien contenant le descriptif de toutes les classes QT <http://doc.trolltech.com/3.3/classes.html>).

**Une attention particulière sera portée à votre analyse (orientée objet/procédurale) du problème, à la gestion des erreurs, à la qualité du code écrit et documenté, et sur la démonstration finale démontrant l'achèvement du projet.**

*Si vous n'arrivez pas au bout des trois étapes, prévoyez de démontrer des parties de votre code qui fonctionnent. Mieux vaut une démarche graduelle, que rien du tout...*

**Remise du projet :**

Ce travail est à réaliser en trinôme. Vous devez fournir un fichier .tar.gz contenant les sources **commentés**, un mécanisme d'installation (Makefile recommandé), et un compte rendu de 5 pages expliquant votre travail.

**Date de remise :** mardi 8 janvier 2019.

**Date de soutenance :** elle vous sera communiquée ultérieurement.