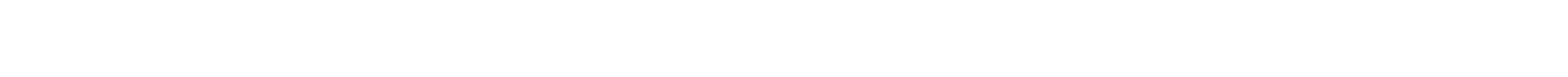
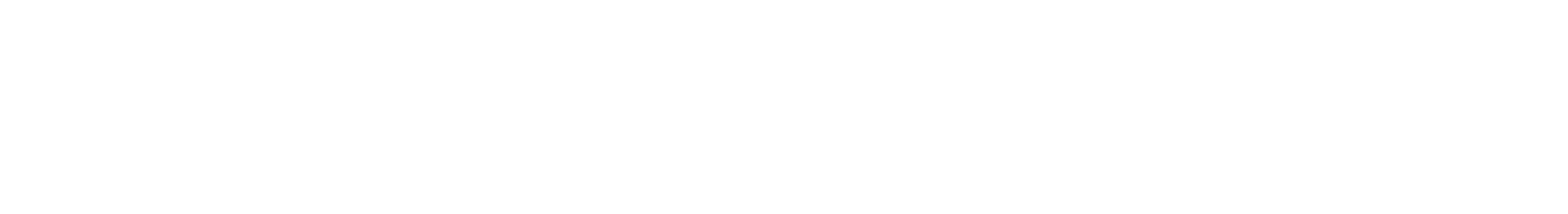


RAPPORT PROJET C

TANKS BATTLE



LAB 3040



GOUNOT Damien | HOANG Jacky

# SOMMAIRE

# SUJET

L’objectif de ce projet est de programmer en langage C, un jeu de bataille de tanks.

Le but est d’éliminer tous les tanks ennemis tout en protégeant le petit oiseau (Titi).

Le projet est à réaliser obligatoirement en binôme (ancien & nouveau) sous environnement GNU/Linux. L’affichage doit être effectué uniquement sur console.

Au lancement du jeu, le joueur doit pouvoir choisir entre deux modes de jeu :

Facile : moins de tanks seront générés (ex. 15), les tanks avancent lentement, les tirs sont

moins fréquents, etc.

Difficile : plus de tanks seront générés (ex. 30), les tanks avancent rapidement, les tirs sont

plus fréquents et plus rapides, plus de tanks ultra-blindés, etc.

# PRINCIPE D’EXECUTION

## 

## Lancement

Au lancement du programme, l’utilisateur est dans le menu principal (voir Figure 1 en Annexe) où il a la possibilité de sélectionner la fonctionnalité qu’il souhaite utiliser :

* Mode Facile (Easy)
* Mode Difficile (Hard)

Pour sélectionner le mode facile, il suffit d’appuyer sur la touche E du clavier.

Pour sélectionner le mode difficile, il suffit d’appuyer sur la touche H du clavier.

La touche Echap quand à elle, est utilisable pour quitter l’exécution du programme.

## Choix du mode

Le déroulement d’une partie est dans le principe, le même, quel que soit le mode choisi.

Cependant notons les spécificités propres à chaque mode.

Concernant le mode Facile :

* Titi est protégé dans un Bunker
* Les ennemis sont uniquement de niveau 1
* Nombre d’ennemis inférieurs

Concernant le mode Difficile :

* Titi n’est plus protégé, il est livré à lui même
* Les ennemis sont de niveau 1, 2 et 3
* Nombre d’ennemis supérieur (triplé)

## Initialisation de la carte

C’est lors du lancement de la partie que la carte est initialisée.

Pour cette génération, il suffit de s’imaginer 3 Matrices de tailles identiques.

La première matrice (appelons la « map ») est utilisée pour générer les cases du plateau (l’aspect visuel).

La seconde matrice (appelons la « collisions ») est utilisée pour attribuer ou non une collision à une case.

Enfin la troisième matrice (appelons la « destruction ») est utilisée afin de définir si une case du terrain à la possibilité d’être détruite.

Prenons l’exemple d’un fleuve, il ne peut être détruit, ainsi que la bordure de notre carte.

Un mur quand à lui possède une collision et peut être détruit. (NB : il en va de même pour les tanks et titi)

Tout est géré à partir de fichiers .txt



Le joueur ainsi que les tanks apparaissent sur la carte de façon aléatoire, il est cependant vérifié qu’il soit impossible qu’ils apparaissent sur une case présentant une collision (et oui, il serait dommage d’apparaître dans un mur ou au milieu de l’océan…)

Ils ne peuvent également pas apparaître sur un autre tank (car eux même possèdent leur propre collision !)

## Déroulement de partie

Maintenant que tout a été correctement initialisé intéressons nous au déroulement d’une partie.

Tout d’abord, il n’y a que 2 possibilités de finir une partie :

* Perdre (si Titi ou le joueur viennent à mourir)
* Gagner (si le joueur détruit l’intégralité des tanks présents sur la carte)

Il n’y a aucune possibilité de match nul.

Tout au long de la partie, chaque tank indépendamment les uns des autres vont se déplacer aléatoirement sur la carte, ils auront un pourcentage de chance de tirer à chaque déplacement.

Leur portée de tir est fonction de leur niveau :

* Un Tank de niveau 1 tir sur la case en face de lui
* Un Tank de niveau 2 tir jusqu’à 2 cases devant lui
* Un Tank de niveau 3 tir jusqu’à 3 cases devant lui

Le joueur est initialisé comme étant un tank de niveau 2 afin de conserver un équilibrage correct.

Le joueur peut diriger son tank avec les touches fléchées, ou Z,Q,S,D , et tirer avec la barre d’espace.

A noter que si titi est en liberté au cours de la partie, il cherchera a s’enfuir (via des déplacements aléatoires)

# REPARTION DES TACHES

Pour organiser un travail il faut au préalable identifier l’ensemble des tâches qui lui sont liées, les planifier et attribuer les responsabilités nécessaires à leur réalisation.

Une bonne organisation (individuelle et en équipe) permet au projet d’avancer.

Déléguer les différentes tâches dans un groupe permet de se libérer du temps pour solutionner les éventuels problèmes qui pourraient découler du projet. Cela fait partie des avantages du travail d’équipe.

Comment avons-nous réparti le travail ? Et à quel point cela influence notre efficacité ?

|  |  |  |
| --- | --- | --- |
| Acteur principal | Tâche | Assistance |
| Sylvain | Ajout de contact | Jacky |
| Jacky | Suppression de contact | Sylvain |
| Damien | Sauvegarde des contact | Jacky & Sylvain |
| Damien | Recherche de contact | Jacky & Sylvain |
| Sylvain & Jacky | Fonctions d'affichages basiques (menu) | Damien |

La supervision du travail individuel par les autres membres de l’équipe permet alors à chacun d'avoir une vision globale du projet et de son fonctionnement.

Ainsi chaque personne est concernée et joue un rôle conséquent dans l'évolution du projet.

De cette façon, la répartition du travail se justifie et permet au projet une avancée efficace et organisée.

# FONCTIONS & ALGORITHMES

# DIFFICULTES RENCONTREES

# ANNEXES