**Buffy – Labo 4**

Une image contenant habits, Personnage de fiction, Art numérique, personne

Description générée automatiquement

*Durée du travail :* **Du 23.05.2024**

**au 11.06.2024**

*Auteurs :* **Demont Killian   
Graf Calvin**

*Enseignant :* **Krähenbühl Grégoire**

*Assistant :* **Decorvet Grégoire**

*Domaine d’application :* **C++**

*Lieu de travail :* **HEIG-VD | Yverdon-les-Bains**

Table des matières

[1. Introduction 2](#_Toc169048589)

[2. Développement 3](#_Toc169048590)

[2.1 Instruction de compilation 3](#_Toc169048591)

[2.2 Choix d’implémentation 3](#_Toc169048592)

[2.3 Protocole de tests 4](#_Toc169048593)

[2.4 Statistiques 5](#_Toc169048594)

[3. Conclusion 7](#_Toc169048595)

[4. Diagramme de classe 0](#_Toc169048596)

## Introduction

Ce laboratoire vise à concevoir une simulation où Buffy, la tueuse, doit sauver l'espèce humaine de la menace des vampires. L’objectif est d’implémenter la simulation des actions des vampires, des humains et de Buffy (qui sont des humanoïdes), en mode "tour par tour", l’affichage du plateau de jeu et une option permettant d’effectuer des statistiques sur un ensemble de 10’000 simulations.

Buffy (B), les vampires (V) et les humains (h) sont placés aléatoirement sur une grille. La taille de la grille, le nombre d'humains et de vampires sont passés en paramètre du programme. Un menu affiche le numéro du tour courant et permet d’effectuer les différentes opérations disponibles.

À chaque tour de la simulation, chaque humanoïde décide de sa prochaine action. Toutes les actions sont ensuite résolues simultanément pour éviter tout avantage pour les premiers humanoïdes dans la liste, puis on affiche le résultat à l’utilisateur.

## Développement

* 1. Instruction de compilation

Nom du compilateur : Mingw-w64 gcc

Version du compilateur : 13.2.0

* 1. Choix d’implémentation

On a décidé de laisser l’option aux humanoïdes d’être sur la même case car on ne voyait aucune raison ni technique ni dans le principe du jeu de l’interdire.

Pour les statistiques, on a décidé d’utiliser un struct plutôt qu’une classe car on n’a qu’une méthode statique sans méthodes ni attributs. Ce qui nous permet de simplifier et alléger le code

Enfin, on a décidé après la discussion que nous avons eu ensemble de mettre en commentaire la partie qui efface la console à chaque tour afin de ne pas créer de bugs inutiles (et perdre des points pour une partie facultatif) mais je l’ai laissé à disposition si vous souhaitez y jeter un coup d’œil.

* 1. Protocole de tests

|  |  |
| --- | --- |
| **Test** | **Résultat** |
| Buffy se dirige toujours de 2 cases vers le vampire le plus proche | OK |
| Lorsque Buffy est à 1 case ou moins d’un vampire, elle le tue | OK |
| Les vampires se dirigent toujours d’une case vers l’humain le plus proche | OK |
| S’il n’y a plus d’humain, les vampires ne bougent plus | OK |
| Les humains se déplacent d’une case au hasard | OK |
| S’il n’y a plus de vampires, Buffy se déplace comme les humains (d’une case au hasard) | OK |
| Tous les humanoïdes peuvent se déplacer horizontalement, verticalement ou diagonalement. | OK |
| Si un vampire ou un humain est mort, il disparaît du jeu | OK |
| Si un vampire mord un humain, celui-ci se transforme en vampire à son tour | OK |
| Si l’utilisateur entre « n », le jeu passe au tour suivant et chaque humanoïde effectue son actions | OK |
| Si l’utilisateur entre « q » le jeu s’arrête | OK |
| Si l’utilisateur entre « s » il obtient le % de taux de victoire de Buffy avec les conditions de jeux actuels sur 10'000 simulations | OK |
| S’il n’y a plus de vampires mais qu’il reste des humains c’est considéré comme une victoire pour Buffy | OK |
| S’il n’y a plus d’humains c’est considéré comme une défaite pour Buffy | OK |
| Si l’utilisateur entre une autre commande un message d’erreur lui est retourné et rien ne se passe | OK |
| Si la taille du plateau de jeu, son nombre d’humain ou de vampires est inférieur à 1 une erreur est retournée | OK |
| Les humanoïdes ne se déplacent pas en dehors de la grille du jeu | OK |
| Un vampire mord une fois sur deux et tue une fois sur deux | OK |

* 1. Statistiques

Si nous effectuons dix fois la simulation (de 10'000 itérations chacune) avec les conditions initiales soit une grille de 50x50, 10 vampires et 20 humains on obtient un pourcentage de victoire pour Buffy situé entre 53 et 55%.

Il est intéressant de voir qu’en doublant le nombre d’humains (40), le pourcentage de victoire n’augmente que très légèrement (environ +1%) ce qui se justifie par la transformation des humains en vampire lorsqu’ils se font mordre par les autres vampires. Donc la mission n’est pas vraiment facilitée pour Buffy.

En revanche, en divisant le nombre d’humains par deux, le taux tombe aux alentours de 38% en moyenne car les vampires arrivent plus rapidement à tuer tous les humains.

En faisant l’expérience inverse, c’est-à-dire en gardant 20 humains mais en doublant le nombre de vampire (donc 20) on obtient un pourcentage de victoire proche d’1% pour Buffy qui se fait submerger par le nombre de vampires.

Une image contenant ligne, Tracé, capture d’écran, diagramme

Description générée automatiquementPuis si on ajoute uniquement 5 vampires, on arrive à un taux de victoire proche de 98%.

Nous avons réalisé plusieurs simulations avec différents cas mais toujours sur une grille de 50x50 et 50'000 parties. Voici le graphique que l’on en a tiré :

On remarque qu’on tend très vite vers 0% en augmentant le nombre de vampire et ce peu importe le nombre d’humain. Une fois la barre des 20 vampires atteint, c’est presque impossible pour Buffy de gagner la partie.

On peut également voir que l’augmentation d’humain impacte grandement le % de victoire de Buffy au début, mais plus il y a d’humains, plus cet écart se rétrécit rapidement.

Ci-dessous les résultats du graphe avec plus de précision.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

## Conclusion

Toutes les actions et personnages ont pu être implémenté, le jeu ainsi que le menu fonctionnent sans problème. L’affichage est respecté et le calcul de statistiques l’est également. Nous avons factorisé au plus la conception des classes afin de faciliter l’implémentation d’éventuelles nouvelles actions ou personnage.

Tous les points du cahier des charges ont été remplis et testés avec succès.

## Diagramme de classe