



LE MANS UNIVERSITÉ
LICENCE INFORMATIQUE 2ÈME ANNÉE

YAHTZEE

COLLINS SOARES ET FATIH UFACIK

AVRIL 2019

Sommaire

1	Introduction	2
2	Organisation du travail	3
2.1	Comment le travail à été répartie	3
2.2	Outils de travail commun utilisés	3
3	Conception du projet	5
3.1	Règles du jeu avancées	5
3.2	Comment une partie fonctionne	5
3.3	Analyse du travail à faire	6
4	Developpement du projet	7
4.1	Structures de données utilisées	7
4.2	Jeux de tests réalisés	7
4.3	Problèmes rencontrés et réponses trouvées	8
5	Résultats obtenus	9
5.1	Les objectifs sont-ils atteints ?	9
5.2	Le projet à t-il été réalisé dans les temps indiqués	9
5.3	Le projet correspond à nos attentes	9
6	Conclusion	11
6.1	Quelles sont les améliorations possibles ?	11
6.2	Ce que le projet nous a appris ?	11
7	Annexes	12

Chapter 1

Introduction

Le Yahtzee est un jeu de société de hasard dont le but est de réaliser le maximum de points en lançant 5 dés afin de former des combinaisons prédéfini. Nous avons donc choisi de réaliser ce jeu sous forme numérique pour notre projet.

Pour réaliser notre Yahtzee nous avons, dans un premier temps défini clairement les règles du jeu et ce que nous voulions faire.

Nous avons donc décidé de réaliser un jeu fonctionnel sur terminal en 1 contre 1 puis d'accompagner ce jeu avec une interface graphique sous SDL2. Puis pour finir nous avons décidé de réaliser une IA permettant au joueur de jouer une partie contre l'ordinateur.

Chapter 2

Organisation du travail

2.1 Comment le travail à été répartie

Concernant le jeu sur terminal nous avons décidés de travailler dessus à deux pour rapidement avoir une version fonctionnelle du jeu sur le terminal. Pour cela nous nous sommes répartis les fonctions de notre jeu entre nous.

Une fois le jeu codé et fonctionnel sur terminal nous nous sommes répartis le travail restant d'un coté l'interface graphique SDL du jeu et de l'autre le jeu contre l'ordinateur.

On a donc décidé que UFACIK Fatih se chargera de l'interface graphique sous SDL2.0 puis SOARES Collins se chargera du jeu contre l'intelligence artificielle.

2.2 Outils de travail commun utilisés

Pour mener à bien notre projet nous avons utilisés plusieurs outils de travail collaboratif tel que Github ou encore Overleaf ...

Github est un environnement de travail permettant de partager des fichiers parmi le groupe facilement afin d'avancer rapidement sans forcément se voir. Cet environnement nous à donc permis de partager le code du jeu sur terminal notamment.

Pour ce qui est de l'écriture de code nous avons principalement utilisés deux logiciels, Sublime Text et Virtual Studio Code. Sublime Text sur les machines de l'université et Virtual Studio Code sur nos machines personnels permettant de compiler facilement sous Windows.

Pour créer la documentation de notre coden nous avons utilisés le logiciel

Doxygene nous permettant de créer facilement la documentation de notre code directement à partir des commentaires écrits sur le code. C'est une manière simple et rapide de créer une documentation.

Le rapport a, quant à lui, été écrit via Overleaf en Latex. C'est un éditeur de texte en Latex permettant l'affichage du rendu final plus facile et rapidement en même temps que nous écrivons le texte/code.

Pour finir avec les logiciels, nous avons utilisé Google Slides afin de réaliser la présentation orale de notre projet.

Chapter 3

Conception du projet

3.1 Règles du jeu avancées

Le Yahtzee est un jeu de hasard se jouant avec 5 dés et se terminant quand toutes les cases de la fiche de score sont remplies.

Une partie se déroule en tour par tour et chaque joueur dispose de 3 lancers durant son tour. L'objectif d'un tour de jeu est de réaliser une combinaison qui va engranger un nombre de points prédéfini ou défini par les dés obtenus.

Pendant son tour le joueur a le choix, à partir du deuxième lancer, de relancer la totalité ou certains dés selon la combinaison qu'il souhaite réaliser. A la fin de chaque tour le joueur doit obligatoirement choisir une combinaison de la fiche de score quitte à ne mettre 0 points.

Evidemment chaque combinaison ne peut être réalisée qu'une seule fois. Un bonus peut être attribué aux joueurs si le total intermédiaire du joueur atteint ou est supérieur à 63 points. Si c'est le cas, le joueur en question reçoit un bonus de 35 points.

3.2 Comment une partie fonctionne

Les dés du joueur sont lancés, les combinaisons possibles avec les dés obtenus sont déterminés puis le joueur doit décider si il veut relancer certains dés de sa main.

Si le joueur choisit de relancer, la relance des dés sélectionnés est réalisée puis le joueur doit encore faire son choix : relancer certains dés ou attribuer des points à une combinaison.

Ensuite le tour du joueur se termine et l'autre joueur réalise un tour de

jeu. Les tours s'enchainent et quand la fiche de score est remplie la partie se termine.

Le gagnant est le joueur ayant obtenu le plus de points durant la partie.

3.3 Analyse du travail à faire

Pour écrire notre code nous avons eu besoin de plusieurs fonction c'est donc ce que nous avons commencer à voir. Nous avons dénombrer un total de 16 fonctions. Des fonctions permettant le lancement ou la relance des 5 dés ou moins. D'autres fonctions permettant le reperage des combinaisons et l'attribution des points. Des fonctions d'affichage tel que l'affichage des dés du joueurs ou encore l'affichage de la fiche de score de la partie. Nous avons également ecrit une fonction permettant la réalisation d'un tour de jeu puis d'autres fonctions permettant le bon déroulement d'une partie.

Nous avons donc déterminé toutes les fonctions nécessaire au bon déroulement de notre jeu sur terminal puis nous avons commencé à réfléchir sur l'interface graphique et le jeu contre l'ordinateur.

Nous avons donc conclu que UFACIK Fatih se chargera de l'interface SDL et qu'il faudrait un menu avec le titre du jeu et 3 boutons, bouton joueur vs joueur, bouton jeu contre ia et un bouton quitter. Puis c'est SOARES Collins qui se chargera du jeu contre l'ordinateur avec la mise en place d'une partie où l'ordinateur prend le maximum de points des combinaisons réalisables avec les dés du joueur puis une partie ou la relance est gérée par des probabilités.

Chapter 4

Developpement du projet

4.1 Structures de données utilisées

Nous avons ensuite déterminer les structures de données que nous souhaiteront utilisés dans notre programme. Premièrement nous avons mis en place une structure représentant la main d'un joueur (soit les 5 dés) et le nombre de faces identiques (nombre de un dans la main, nombre de deux dans la main etc.)

Ensuite nous avons mis en place une deuxième structure représentant la fiche de score de la partie, contenant chaque combinaison et le nombre de points que le joueur a réalisé et les totaux de la fiche de score (partie supérieure, inférieure et supérieure + inférieure)

Cependant nous avons utilisé des tableaux a la place de structures pour la main et la fiche de score dans le jeu contre l'ordinateur car nous avons trouvé cela plus simple d'utilisation et plus pratique dans le code du jeu contre l'IA.

4.2 Jeux de tests réalisés

Pour teste notre programme afin de le corriger nous avons commencé par des tests simple comme simuler un lancer et afficher la main ou encore simuler une relance avec le choix des dés à relancer.

Nous avons également simulé un tour de jeu pour voir si l'attribution des scores s'effectuer correctement pour la combinaison effectuée lors du choix du joeur. Puis nous avons fait énormement de tests concernant l'affichage des scores, des main/dés ou encore d'un tour de jeu ou du choix des combinaisons.

Il a fallut en effet vérifier tous les affichages des combinaisons disponibles en fonction des dés disponibles ou encore tester l’affichage de la fiche de score etc.

4.3 Problèmes rencontrés et réponses trouvées

Les premiers problèmes que nous nous sommes posés étaient de trouver une méthode efficace pour gérer la relance des dés. Nous avons fini de créer une fonction qui demande l’indice du dé et qui obtient la valeur grâce à cet indice.

Nous avons également eu besoin de gérer l’affichage de score pour se faire nous avons commencé par utiliser une structure contenant toutes les données correspondant au score de la partie. Mais lors de l’affichage, la fiche de score se remettait à zéro à chaque tour.

Pour résoudre cela nous avons utilisé plusieurs fonctions, la structure de la fiche de score et des tableaux stockant les données de la fiche de score. Permettant ainsi un parcours des données plus facile et un affichage sans problèmes.

Concernant le jeu contre l’ordinateur, le premier problème rencontré a été de trouver la méthode permettant de faire jouer l’ordinateur, pour résoudre cela nous avons décidé de faire lancer les dés à l’ordinateur puis dans un premier temps prendre regarder toutes les combinaisons possibles avec les dés obtenus puis de prendre celle qui apporte le plus de points parmi celle encore non réalisée.

Mais cette idée, ne permet pas de faire relancer l’ordinateur ce qui est en contradiction avec les règles du jeu. En effet le joueur doit pouvoir lancer les dés 3 fois maximum.

Pour régler cela, nous avons décidé de prendre en compte des probabilités afin de gérer les relances de dés. Pour cela après le premier lancer la probabilité d’avoir une certaine combinaison en fonction de la meilleure disponible est calculée et si cette dernière est supérieure à 60% alors la relance de dés est effectuée.

Chapter 5

Résultats obtenus

5.1 Les objectifs sont-ils atteints ?

En fin de projet nous avons réussi à avoir un jeu sur terminal en joueur contre joueur fonctionnel ainsi qu'un début d'interface graphique et un début d'IA. En effet la partie avec les probabilités de l'IA n'a pas plus été terminée et la première partie avec le maximum de points pris rencontre encore quelques bugs (dans l'attribution des scores des combinaisons total de A à total de 6).

Ce qui fait que un objectif a été complètement atteint (le jeu joueur contre joueur dans le terminal) et le reste des objectifs a été partiellement atteint.

5.2 Le projet a-t-il été réalisé dans les temps indiqués

Si on considère que les objectifs ne sont pas tous terminés à 100% on peut dire que le projet n'a pas été terminé dans les temps indiqués mais vu que le projet non 100% terminé a été déposé dans les temps on peut également dire que le projet a été réalisé dans les temps indiqués.

5.3 Le projet correspond à nos attentes

Non, le projet ne correspond pas à nos attentes dans sa globalité car nous avons eu une image précise de l'interface graphique et du jeu contre l'ordinateur qui n'a pas été complètement atteinte.

Cependant, nous avons atteint ce que nous souhaitions concernant le jeu sur terminal en joueur contre joueur. En effet toutes les fonctionnalités que nous

avons voulu implémenté est fonctionnel puis une partie peut être jouer sans soucis.

Chapter 6

Conclusion

6.1 Quelles sont les améliorations possibles ?

Les améliorations possibles que nous pouvons envisager de faire pour notre projet sont de mettre un jeu en réseau (jeu en ligne) afin de jouer avec deux ordinateurs différents.

Par la suite nous pouvons améliorer notre interface graphique pour la rendre plus jolie et nous pouvons aussi ajouter le mode ordinateur contre joueur (IA vs JOUEUR).

Quand à l'IA (intelligence artificielle) nous pouvons l'améliorer en ajoutant des probabilités conditionnelles pour ajouter la possibilité de relancer de l'ordinateur pour qu'ils fassent de meilleurs combinaisons possibles avec les dés qui lui sont proposés et nous pouvons aussi corriger l'affichage de notre IA.

6.2 Ce que le projet nous a appris ?

En réalisant ce projet, nous avons appris énormément de choses. Dans un premier temps nous avons appris à réaliser un travail de groupe dans un temps imparti et de le mener à bien. C'est une chose que nous avions pas l'habitude de faire dans le passé.

Ensuite nous avons appris à utiliser de nouvel outils comme par exemple github qui nous a permis de nous partager nos fichiers à distance ou encore nous avons appris à utiliser SDL2 et à créer une interface graphique pour un jeu vidéo en 2D.

Figure 7.2: Exemple d'un tour de l'ordinateur

```

> 6> 4> 1> 5> 5
T > 1 | 1 pts
T > 4 | 4 pts
T > 5 | 10 pts
T > 6 | 6 pts
Ch   | 21 pts
21
-----
Joueur          | 1
Total de 1      | 00
Total de 2      | 00
Total de 3      | 00
Total de 4      | 00
Total de 5      | 00
Total de 6      | 00
Total supérieur | 00
Bonus (35pts)   | 00
Brelan          | 00
Carré          | 00
Full (25pts)    | 00
Petite Suite (30pts) | 00
Grande Suite (40pts) | 00
Yams (50pts)    | 00
Chance          | 21
Total inférieur | 21
Total           | 21
-----
> 2> 6> 3> 5> 3
T > 1 | 1 pts
T > 2 | 2 pts
T > 3 | 6 pts
T > 4 | 4 pts
T > 5 | 15 pts
T > 6 | 12 pts
B > 5 | 15 pts
F > 5 | 25 pts
F > 5 | 25 pts
P 1-4 | 30 pts
Ch    | 19 pts

```