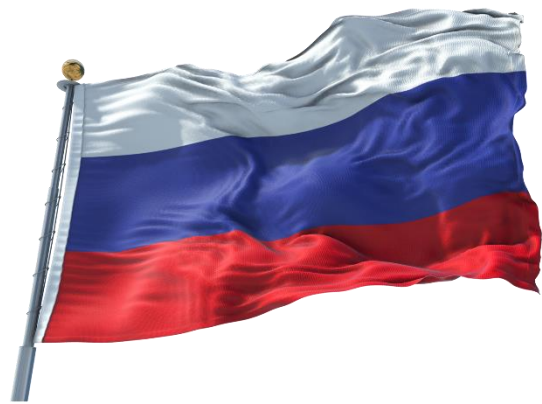


Projet Java

Jeu de roulette Russe



Université Lyon 1 : LP DEVOPS

Equipe : Yann MERLIN, Mostefa ALI KADA

Professeur : Clément BATTARD

Table des matières

I – Description générale du projet.....	3
a. Résumé et contexte	3
b. Utilisateurs ciblés.....	3
c. Objectifs.....	4
d. Livrables.....	4
e. L'équipe.....	4
II – Contraintes	5
a. Contraintes technologiques.....	5
b. Contraintes matérielles et financières	5
c. Contraintes humaines	5
d. Délais.....	6
III – Prestations attendus	7
a. Frontend design	7
b. Backend	7
c. Réseau	8
IV – Identification des risques	9
V – Fonctionnement	10
a. User stories.....	10
b. DCU	10
c. DCL	10

I – Description générale du projet

a. Résumé et contexte

Ce projet est réalisé dans le cadre du cours de Java de la licence DevOps de l'IUT Lyon 1. Notre professeur est Mr BATTARD Clément.

Ce projet consiste en la réalisation d'un programme en langage Java, pour appliquer les différentes notions vues en cours. Nous avons choisi la réalisation d'un jeu de roulette Russe, multijoueur sur le même réseau, composé de deux joueurs (clients) et d'un serveur.

Le jeu de la roulette russe est un jeu traditionnel Russe dans lequel deux joueurs ou plus s'affrontent, généralement pour de l'argent, en pointant chacun leur tour une arme chargée d'une seule balle dans le barillet sur leur tempe ou sous leur gorge. Il pressent alors la détente, et suivant la position de la balle dans le barillet (déterminée aléatoirement en début de partie) meurent ou survivent au tir. Le gagnant et le dernier survivant, qui empoche alors la mise de tous les autres joueurs.

Pour rentrer dans un contexte universitaire, les humains de ce jeu seront ici remplacés par des poulets, et peu d'effets graphiques seront visible. L'objectif est d'utiliser le concept de l'aléatoire, pas celui de la mort.

b. Utilisateurs ciblés

Les utilisateurs ciblés par cette application sont les suivants ; tout d'abord notre professeur, qui réalisera l'évaluation, la notation et la correction du projet, puis dans un second temps tout type de joueur qui souhaite essayer le jeu. Aucun prérequis ou connaissance en informatique n'est requis pour utiliser l'application, puisque celle-ci disposera d'une interface graphique simple et claire.

c. Objectifs

Les objectifs (triés par ordre d'importance) sont les suivants :

1. Créer un serveur
2. Créer un client
3. Permettre à deux clients de se connecter simultanément sur le serveur
4. Permettre aux joueurs de communiquer via un tchat écrit
5. Pouvoir jouer une partie
6. Pouvoir modifier les paramètres de la partie (difficultés, type d'armes)
7. Intégration avec la base de données
8. Identification des joueurs dans la bd
9. Mémorisation des différents scores des joueurs dans la bd
10. Implémenter un joker par joueur (relancer le barillet)

d. Livrables

Les livrables de ce projet sont les suivant :

Tout d'abord ce cahier des charges pour définir le projet.

Ensuite l'application en elle-même (le code Java, JavaFX et ses dépendances) et une documentation de type Java doc pour accompagner l'application.

Un rapport écrit sous forme PDF composés de 3 parties. Une partie de présentation des fonctionnalités, une sur l'architecture générale de l'application et une concernant la gestion de projet.

Et enfin une présentation orale (power point + démonstration).

e. L'équipe

Notre équipe est composée de deux développeurs : ALI KADA Mostefa et MERLIN Yann.

Nous sommes des étudiants en alternance en licence DevOps, de niveau intermédiaire en langage Java.

II – Contraintes

a. Contraintes technologiques

Pour ce projet les technologies à utiliser nous sont déjà imposées. Nous allons utiliser le langage Java, avec la librairie JavaFX pour l'interface graphique, ainsi qu'une librairie de socket pour la partie réseau. L'IDE sera IntelliJ ou Eclipse selon les préférences des membres de l'équipe. La base des données sera administrée avec SQL developer et Oracle puisque c'est le choix de l'IUT.

GitHub sera utilisé pour sauvegarder à distance le projet et effectuer la gestion des versions.

b. Contraintes matérielles et financières

Le matériel est celui de l'IUT, le développement et l'évaluation se feront en grande partie sur les PCs de la salle S01. De même, le réseau sera celui de l'IUT, et puisqu'il est limitant, les deux clients et le serveur seront tous les trois exécutés sur un seul et même ordinateur.

Une partie du développement pourra également s'effectuer hors des cours Java et même à la maison, mais les membres de l'équipe disposent personnellement de tout le matériel nécessaire, donc pas de contrainte à ce niveau-là.

Le budget pour ce projet (si l'on ne compte pas le prix du matériel déjà présent à l'IUT) atteint un grand total de 0\$.

c. Contraintes humaines

L'équipe est composée de seulement deux personnes, ayant un niveau Java assez limité, et ne travaillant pas à plein temps sur ce projet (seulement sur les heures de cours Java). Cette contrainte limite la taille de ce projet à une application assez simple.

Notre méthode de travail sera basée sur du pair programming puisque la nature du projet ne permet pas une répartition des tâches sur deux machines simultanément. Néanmoins il arrivera que l'on collabore à distance via un repo GitHub.

d. Délais

Les deadlines sont les suivantes :

- Rapport PDF : le 1 février 2024 à minuit.
- Code et documentation : le 6 février 2024.
- Présentation orale et démo de l'application : le 9 février 2024.

Le planning prévisionnel est le suivant :

Date:	04/01/24 - 4h	11/01/24 - 4h	12/01/24 - 4h	26/01/24 - 4h	01/02/2024	06/02/2024	09/02/24 - 2h
Tache	Cahier des charges	Créer un serveur Créer un client Deux clients peuvent se connecter simultanément sur le serveur. Permettre aux joueurs de communiquer via un tchat écrit. Pouvoir jouer une partie via l'interface graphique	Modifier les paramètres de la partie via interface graphique Intégrer la base de données Identification des joueurs dans la bd	Scores des joueurs dans la bd Implémenter un Joker Faire le rapport PDF	Rapport PDF	Code et doc	Oral

III – Prestations attendus

a. Frontend design

L'interface sera la plus simple possible, composée des éléments suivants :

- Un écran d'accueil pour identifier les joueurs.
- Un menu pour rejoindre ou créer une partie.
- Des boutons pour tirer sur soi ou sur l'adversaire.
- Des poulets vivant ou mort pour visualiser l'état de la partie.
- Une arme.
- Un écran de fin de partie.
- Un tableau des scores.

b. Backend

Le backend contiendra toute la logique de l'application et permettra le déroulement d'une partie de la manière suivante :

Joueur1

1. Lancement de l'application
2. Le joueur s'identifie en tapant son nom
 - a. S'il existe, on récupère ses précédents scores
 - b. Sinon, on crée le joueur dans la BDD
3. Choix dans le menu
 - a. Créer une partie
 - b. Rejoindre une partie
 - c. Quitter
4. Le joueur choisit de créer une partie
5. Le joueur attend que le second rejoigne la partie avant de commencer le paramétrage.
6. Le joueur paramètre la partie
 - a. 1^{er} Indicateur de difficulté : La taille du barillet/Modèle de Revolver : 8-6-5
 - b. 2^{ème} indicateur de difficulté : Le nombre de balle dans le barillet

Joueur 2

1. Lancement de l'application
2. Le joueur s'identifie en tapant son nom
 - a. S'il existe, on récupère ses précédents scores
 - b. Sinon, on crée le joueur dans la BDD
3. Choix dans le menu
 - a. Créer une partie
 - b. Rejoindre une partie
 - c. Quitter
4. Le joueur choisie de rejoindre une partie
5. Le joueur attend que l'hôte finalise les paramètres de la partie

Commun

1. Lancement de la partie
2. Lancement d'une pièce Pile -> J1 ou Face -> J2
3. Le gagnant du Pile ou Face choisi le joueur qui commence.
4. 2 choix d'action possible
 - a. Se tirer dessus
 - i. Cas 1 : *click, la partie continue et on passe à l'autre joueur
 - ii. Cas 2 : *BANG, la partie est terminée et l'autre joueur gagne
 - b. Tirer sur l'autre joueur
 - i. Cas 1 : "click, la partie est terminée et l'autre joueur gagne
 - ii. Cas 2 : *BANG, la partie est terminée et le joueur gagne
5. Affichage du tableau des scores
6. 2 choix d'action possible :
 - a. Relancer avec les mêmes paramètres
 - b. Relancer avec de nouveaux paramètres

c. Réseau

L'application sera capable de faire communiquer deux clients (joueurs) via le réseau pour leur permettre de jouer une partie ensemble.

Dans le cas d'utilisation de l'application à l'IUT, le réseau étant limitant pour des raisons de sécurité, les deux clients et le serveur seront exécutés sur le même ordinateur, et donc utiliseront l'adresse réseau localhost 127.0.0.1. Le numéro de port utilisé n'importe pas tant qu'il est dans la plage des ports non réservés. Il faut juste qu'il soit plus grand que 1024.

IV – Identification des risques

Les risques identifiés pour ce projet sont les suivants :

		Niveau de gravité			
		1 - Indolore	2 - Limité	3 - Grave	4 - Dramatique
Probabilité	4 - Très forte	x	Bugs	Difficultés à respecter les délais	x
	3 - Forte	x	x	x	x
	2 - Moyenne	x	Un membre de l'équipe tombe malade	Corruption du code	Réalisation d'un projet non fonctionnel
	1 - Faible	Logiciel inadapté	Manque de compétence	Mauvaise entente entre les membres de l'équipe	Covid 24 Catastrophe naturel affectant l'IUT

Les traitements envisageables de ces risques sont les suivants :

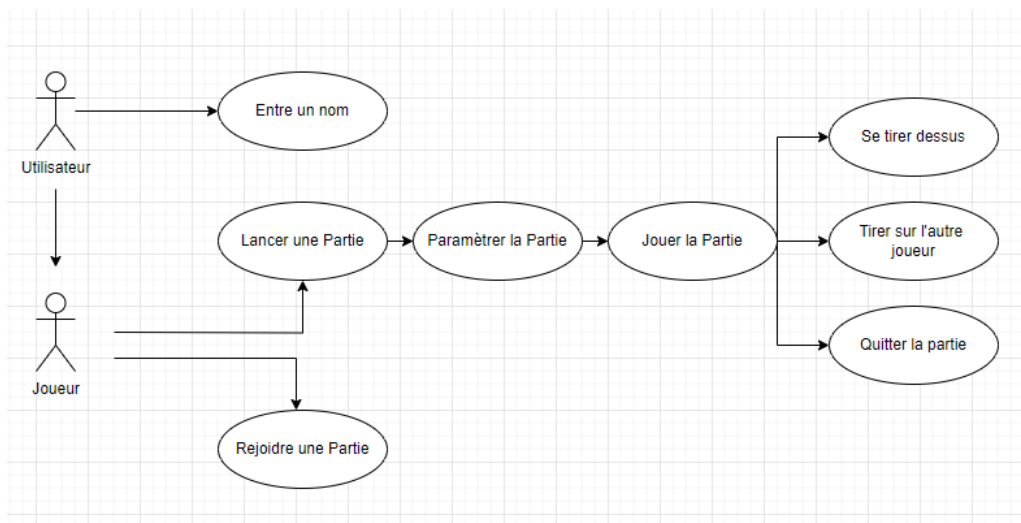
- Bugs :
Effectuer des tests du code pour identifier puis traiter les bugs.
- Difficultés à respecter les délais :
Travail supplémentaire chez soi, en dehors des heures de cours.
- Un membre de l'équipe tombe malade :
Travail à distance depuis la maison, ou, s'il y a incapacité totale de travail, augmenter la charge de travail pour le membre restant et demander au professeur de réévaluer le barème d'évaluation en conséquence.
- Corruption du code :
Récupération d'une version précédente du projet sur GitHub.
- Réalisation d'un projet non fonctionnel :
Multiplier les tests pour prévenir ce risque.
- Logiciel inadapté :
Changement d'IDE, ou de logiciel.
- Manque de compétence :
Poser des questions au professeur, utiliser le cours, travail à la maison.

- Mauvaise entente entre les membres de l'équipe :
Changer la méthode de travail, passer du pair programming à une division nette des tâches.
- Covid 24 et catastrophe naturelle affectant l'IUT :
Travail à distance à la maison.

V – Fonctionnement

a. User stories

b. DCU



c. DCL

