TONY BUI

MARC CHARLEBOIS

Veille Technologique – Développement d’applications

420-6D9, gr. 00001

Session 6

**Bilan de livraison 2**

**Projet de jeux vidéo de combat 1v1 en ligne**

Travail présenté à

M. Mohamed Guesmia

Département de l’informatique

Collège de Maisonneuve

Le 16 février 2023

Note : Ce livrable (Livrable 2) consiste à fournir un BILAN de la deuxième itération du projet, ainsi que le livrable 3 (pour la 5ème semaine) un logiciel fonctionnel, testable et exécutable. La durée de cette itération sera de 3 semaines.

**Date de remise : Livrable 2 pour 16 février et le livrable 3 pour le 23 février.**

1. Vision du produit

**Notre objectif principal est de créer un jeu de combat 1 contre 1 en ligne compétitif qui soit divertissant, passionnant et accessible à tous. Nous mettrons en place des systèmes de matchmaking. Nous travaillerons également sur une mécanique de jeu fluide et intuitive, pour permettre aux joueurs de se concentrer sur leur stratégie et leur technique plutôt que sur les commandes.**

2. Objectif de l’itération 2

Vous précisez l’objectif de chaque itération (Sprint) pour l’atteinte de la vision du projet. Dans ce document, précisez uniquement l’objectif de l’itération 2. L’objectif doit être SMART, en respectant les indicateurs mentionnés suivants :

* Spécifique : un objectif spécifique doit être en lien direct avec le projet du PO
* Mesurable: un objectif mesurable doit être quantifié ou qualifié.
* Acceptable : cet objectif doit être atteignable et partagé par les membres de l’équipe
* Réaliste : un objectif réaliste est un objectif pour lequel le seuil du travail est défini et réalisable dans le délai prévu.
* Temporellement: un objectif temporellement défini est délimité dans le temps : une date butoir avec, éventuellement, des dates intermédiaires. L’objectif doit être clairement défini dans le temps par des termes précis. Par exemple, l’itération 1 doit se réaliser dans un intervalle de 2 à 4 semaines.

Notre objectif pour l’itération #2 est d’avoir un « Game Loop » complet puisque notre projet est un jeu. Un « Game Loop » signifie que nous devons pouvoir jouer au jeu en entier et pouvoir faire toutes les actions de bases du jeu sois les suivantes : Se connecter à une partie contre un adversaire, contrôler un personnage qui peut attaquer, faire du dommage à l’adversaire et pouvoir gagner ou perdre la partie avec un message de victoire ou de défaite. Le reste des fonctionnalités seront réservées à la prochaine itération qui permettra de donner une finition au jeu. Finalement, tout cela doit être fait pendant la période de 2 semaines qui se terminera le 16 février.

3. Projection de coûts et durée à partir des réalisations faites à date

Graphical user interface, text, application

Description automatically generatedDans cette section, vous démontrez à l’aide d’un graphe l’avancement du projet, en montrant les itérations déjà réalisées (itérations et quantité User stories y associées) et celles restant, tout en précisant la vélocité optimiste et pessimiste.

Selon nos avancements, nous aurons le temps au moins de terminer la base du jeu qui est de pouvoir se connecter à distance contre un autre joueur et jouer avec lui. S’il nous reste assez de temps à la fin, nous peaufineront les détails tels que le terrain de combat, la vitesse de mouvement et d’attaque des personnages et tout ce qui est visuel.

4. Critères de succès

1. Une grande base de joueurs : Le succès de votre jeu dépendra en grande partie du nombre de joueurs qui y jouent régulièrement. Ainsi, le nombre d'utilisateurs actifs mensuels est un critère important pour mesurer la réussite de votre jeu.
2. Un taux d'engagement élevé : En plus de compter un grand nombre de joueurs, il est important de veiller à ce que les joueurs s'engagent dans le jeu de manière régulière et prolongée. Un taux d'engagement élevé peut se mesurer par le nombre de parties jouées par utilisateur ou par la durée moyenne de jeu par session.
3. Des retours positifs des joueurs : Les commentaires des joueurs peuvent être une source précieuse de retour d'information pour améliorer votre jeu. Un taux de satisfaction élevé des joueurs, mesuré par des sondages ou des critiques positives, est un critère important de succès.
4. Un matchmaking efficace : L'un des éléments clés de votre jeu sera le système de matchmaking qui permet aux joueurs d'être appariés avec des adversaires de niveau similaire. Un taux de succès élevé de matchmaking, mesuré par des temps d'attente courts et des adversaires équilibrés, est un critère important de succès.
5. Une mécanique de jeu équilibrée : Dans un jeu de combat compétitif, il est essentiel que la mécanique de jeu soit équilibrée pour permettre une expérience de jeu juste et satisfaisante pour tous les joueurs. Un taux de victoire équilibré entre les joueurs et des retours positifs sur la mécanique de jeu sont des critères importants de succès.

5. Résumé et statistiques sur les fonctionnalités à couvrir (nombre par catégorie et total)

Vous précisez la quantité des User Stories déjà réalisées par catégories (Essentiel, important etc.) (sinon dites « Aucun »)

Users Stories déjà réalisées

Essentiel : Camera qui suit le personnage, animations de mouvement, Hitbox des personnages, personnage peut attaquer, Terrain de combat adapté au 1 contre 1 (map)

Important : Menu Principal, Personnage peut bloquer des attaques,

6. Description des besoins produit

Par exemple :

1. Être capable de consulter des formations, leurs cours et télécharger la documentation gratuite
2. Pouvoir créer une formation et des cours associés à cette formation en tant qu’administrateur ou formateur
3. Pouvoir se connecter à un serveur contre un adversaire
4. Pouvoir contrôler un personnage
5. Pouvoir attaquer et bloquer des attaques
6. Être capable de battre l’adversaire en lui enlevant des points de vie (attaquer)
7. Pouvoir voir des animations synchronisées du côté des deux clients

6.1. Description des cas d’utilisation

Vous décrivez les cas d’utilisations pour l’itération en question, tout en incluant les cas d’utilisation de la première itération, en suivant une approche par triplet. Cette approche vise à rédiger les cas d’utilisation ou récits utilisateurs sous forme (sujet, prédicat, objet) afin de réduire l’ambiguïté des exigences du produit. Voici un exemple de cas d’utilisation rédigé sous forme de triplet.

Titre : Déroulement du jeu de combat

Acteur : Joueur

1. Le joueur lance le jeu avec l’exécutable généré par Unity

2. Le joueur appuie sur le bouton « Find Match » pour démarrer une partie contre un adversaire

3. Si aucun « Lobby » n’est déjà créé, le système crée un « Lobby » et attendra qu’un joueur rejoigne. Si un autre joueur avait déjà créé un Lobby et attend pour un adversaire, le joueur rejoindra sa partie.

4. Au moment où les deux joueurs seront prêts, la partie commencera et un compteur de 3 minutes sera lancé.

5. Les deux joueurs auront chacun un personnage contrôlable à partir de leurs ordinateurs distants

6. Le joueur pourra se déplacer avec les touches W / A / S / D, peut attaquer avec la clique gauche de la souris et bloquer une attaque avec le clique droit de la souris.

7. Lorsque le joueur va se déplacer, attaquer ou bloquer une attaque, le système s’occupera de synchroniser les personnages, leurs animations et toutes autres données telles que les points de vie.

Titre : Ajouter Produit

Acteur : Directeur de vente

1. Le directeur de vente accède à la fonctionnalité d’ajout d’un nouveau produit.

2. Le système demande à l’utilisateur (directeur de vente) de saisir les informations.

3. Le directeur saisit les informations du nouveau produit et demande au système d’enregistrer.

4. Le système vérifie les informations

5. Le système enregistre les données ou retourne un message d’erreur

6. Le système confirme l’enregistrement du nouveau produit

6.3. Modèle ou diagramme de cas d’utilisation

Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système.

6.4. Diagrammes de classes (sinon dites « Aucun »)

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations.

Aucun

6.5. Classes de tests (TDD) (sinon dites « Aucun »)

Aucun

6.6. Technologies utilisées

- Unity

- Netcode for GameObjects

- Unity Relay

- Unity Lobby

7. Faits saillants

Indiquez quelques faits saillants qui marquent l’équipe de développement au tout long de déroulement de l’itération 1 et 2

8. Obstacles méritant d'être portés à l'attention de la direction (s'il y en a, sinon dites "Aucun").

Aucun

9. Rapport de rétrospective

Examiner et améliorer ses façons de faire via une **rétrospective** --> Rapport de rétrospective (peut être intégré au bilan de livraison ou pas: choix de l'équipe).