UNIVERSITÉ DU QUÉBEC À MONTRÉAL

RAPPORT DU SPRINT 23

PHILIPPE LEBLANC

BRYAN CÔTÉ

YASSINE H.

TRAVAIL PRÉSENTÉ

À

MONSIEUR HAFEDH MILI

DANS LE CADRE DU COURS

INM5151 PROJET D’ANALYSE ET DE MODÉLISATION

21 AVRIL 2020

Table des matières

[1. Introduction 3](#_Toc38129186)

[1.1 Objectifs 3](#_Toc38129187)

[1.2 Vue d’ensemble du produit 3](#_Toc38129188)

[1.3 Définitions, acronymes et abréviations 3](#_Toc38129189)

[1.4 Documents de références 4](#_Toc38129190)

[1.5 Aperçu du document 4](#_Toc38129191)

[2. Description générale du logiciel 5](#_Toc38129192)

[2.1 Vue d’ensemble des fonctions du produit 5](#_Toc38129193)

[2.2 Contenu des sprints 7](#_Toc38129194)

[3. contenu du sprint actuel 8](#_Toc38129195)

[3.1 Fonctionnalités implantées dans le sprint 8](#_Toc38129196)

[3.2 Modèle de classes 9](#_Toc38129197)

[4. Revue 11](#_Toc38129198)

[4.1 Revue du sprint 23 11](#_Toc38129199)

[4.2 Revue du projet 12](#_Toc38129200)

[5. Annexe 14](#_Toc38129201)

# 1. Introduction

## 1.1 Objectifs

L’objectif de ce document est essentiellement de décrire et communiquer les différents avancements du projet en cours et les fonctionnalités présentement implantées et celles à venir. Ainsi, cette nouvelle approche a été implémentée en réponse aux différents manquements dans l’industrie du FPS. Ce document s’adresse au commanditaire de notre projet afin de transmettre, de façon transparente, une bonne compréhension des besoins des usagers. Il s’adresse aussi aux développeurs, tout comme aux usagers, pour s’assurer que le document décrit leurs besoins de façon fidèle et prendre en compte tout changement mineur potentiel, ou bien pour une restructuration des priorités du projet.

## 1.2 Vue d’ensemble du produit

L’objectif du système Alpha Strike vise à combler le besoin d’entraînement au tir (à l’arme à feu) qu’un utilisateur pourrait avoir avant qu’une partie de jeu officielle se lance. Il permet à l’utilisateur d’améliorer ses réflexes et la qualité de son tir, et ce peu importe l’environnement de jeu FPS. Cet entraînement pourrait s’appliquer de près ou de loin à n’importe quel jeu de tir présent sur le marché. Ceci est justement dû au sous-système « Sandbox » que notre jeu aura. Ce qui démarque Alpha Strike de ses compétiteurs est majoritairement le côté personnalisable des différentes variables d’environnements. On entend ici la vitesse du mouvement, le recul de l’arme, la gravité ou toute autre fonctionnalité permettant de recréer l’atmosphère d’un jeu précis.

## 1.3 Définitions, acronymes et abréviations

**FPS** : Le jeu de tir à la première personne ou en vue subjective, souvent appelée Doomlike et FPS, sigle pour l’expression anglaise first-person shooter, est un genre de jeu vidéo de tir fondé sur des combats en vision subjective (« à la première personne »), c’est-à-dire que le joueur voit l’action à travers les yeux du protagoniste.

**Sandbox** : Le gameplay non linéaire désigne dans un jeu vidéo la possibilité qu’ont les joueurs de compléter des épreuves selon des séquences variables. Chaque joueur peut compléter qu’une partie des épreuves possibles, lesquelles pouvant être jouées dans un ordre aléatoire. Inversement, un jeu vidéo avec un gameplay linéaire confronte le joueur à une séquence fixe d’épreuves : chaque joueur rencontre chaque épreuve, et ce, dans le même ordre.

**Low poly** : Le fait d’utiliser un faible nombre de polygones lors de la modélisation d’éléments 3D, ce qui a l’avantage de prendre moins de ressources machine en optimisant les performances, mais d’un autre côté en créant des textures beaucoup moins détaillées. Pour certains, le style de textures qui résulte de cette technique est attrayant autant au niveau de la performance que de l’esthétique.

## 1.4 Documents de références

**Unity**.<https://unity.com/>

**Screen Tearing**,<https://en.wikipedia.org/wiki/Screen_tearing>

## 1.5 Aperçu du document

Ce document passe en revue les différentes fonctionnalités du projet qui ont été mises en place par l’équipe durant le premier sprint de développement. Un rappel des différentes fonctions du jeu Alpha Strike sera d’abord effectué ainsi que la répartition de ces tâches dans les différents sprints. Les différents éléments implémentés dans le projet seront mis en lumière comme suit :

* Modèle de classes
* Une revue du sprint actuel

# 2. Description générale du logiciel

## 2.1 Vue d’ensemble des fonctions du produit

La première version qui sera développée comprendra une physique fonctionnelle de base. Un personnage pouvant se déplacer et sauter. La fonctionnalité de tirer avec une arme spécifique sera incluse dans cette version. L’environnement d’entraînement sera une chambre fermée avec des cibles simples. Un score sera calculé par rapport au nombre de cibles touchées selon le temps chronométré. Le joueur pourra aussi accumuler des crédits à chaque partie jouée selon un système de devise pour ainsi magasiner des personnalisations d’armes (« skins »).

Les fonctions développées dans Alpha Strike par catégorie seront les suivantes :

●        Environnement :

* Restreindre le joueur à un bac à sable fermé : l’environnement doit être simpliste et personnalisable au goût du joueur (parmi une liste de choix).
* Avoir un système d’éclairage simulant le soleil : un objet dans la scène doit simuler l’éclairage du soleil et ainsi créer les ombrages rendant l’environnement plus vivant.
* Avoir un mur avec des cibles : l’environnement doit être constitué au minimum d’un mur principal sur lequel des cibles apparaissent.
* Ajouter les textures des différents composants de la carte : selon les goûts du joueur, une carte pourrait avoir un environnement futuriste (« sci-fi »), réaliste, loufoque ou même du genre dessin animé (« low-poly »).
* Impacts de tirs : l’environnement sera impacté par les balles tirées par le joueur, par exemple par des textures de trous causés par les munitions.

●        Fonctions utilisateur (joueur) :

* Physique de base : le joueur doit être en mesure de se déplacer dans la carte, sauter et bouger sa vision à l’aide de la souris.
* Tirer une arme : en cliquant sur le bouton gauche de la souris, le joueur doit pouvoir tirer son arme où il vise.
* Recharger une arme : en appuyant sur la touche « R », le joueur peut recharger son arme avec un nouveau chargeur rempli de munitions.

●        Fonctions de l’arme choisie :

* Tirer un projectile selon une trajectoire : l’arme doit pouvoir tirer des munitions selon la trajectoire qui correspond à l’endroit où le joueur vise avec son réticule.
* Se recharger : lorsque l’arme est rechargée par le joueur, une animation appuie l’action pour améliorer l’expérience utilisateur.
* Effets de tir : les différentes armes du jeu ont des comportements différents lorsqu’elles sont tirées par le joueur incluant les effets de recul, sonores et visuels.

●        Menu

* Lancer une partie : lorsque cette option sera choisie, le joueur sera placé dans une instance du champ de pratique et il pourra s’exercer sur les cibles.
* Consulter son meilleur score : le joueur aura l’option de voir son plus haut score selon le mode de jeu.
* Magasin : ce menu permettra au joueur de consulter à la fois son nombre de crédits selon le système de devise du jeu et aussi acheter des « skins », parmi ceux qui s’affichent avec le nombre de crédits possédés, pour personnaliser ses armes.

## 2.2 Contenu des sprints

Tout d’abord, le contenu du premier sprint a servi majoritairement à se familiariser avec l’environnement de développement pour Alpha Strike, c’est-à-dire Unity. Une recherche plus approfondie a permis dans un premier temps de trouver quelques librairies utiles à l’implémentation. Ensuite, un environnement simpliste de tir a été développé par l’équipe afin de recréer un univers sandbox voulu pour la zone de tir. Finalement, les déplacements du joueur ainsi que sa capacité de tirer ont pu être ajoutés au jeu.

Pour le deuxième sprint, il est prévu de voir apparaître des cibles pouvant être tirées par le joueur, ainsi que l’effet de recul sur les armes à feu, afin d’y ajouter un aspect plus réaliste. La partie plus optionnelle de ce sprint sera orientée vers le côté texture de l’environnement et sur le détail des armes, tant par leur couleur que par les types d’armes disponibles pour l’utilisateur.

Pour le troisième et dernier sprint, un système de devise devrait être mis en place, afin de permettre au joueur d’avoir une certaine progression dans l’application. Cette nouvelle monnaie pourrait lui permettre d’acheter à la fois de nouvelles armes dans le magasin du jeu, mais aussi lui permettre d’acheter des skins d’environnement pour le sandbox et ses armes. Finalement, un menu de jeu sera mis en place afin que l’utilisateur soit en mesure de naviguer entre une partie de tir et le magasin du jeu.

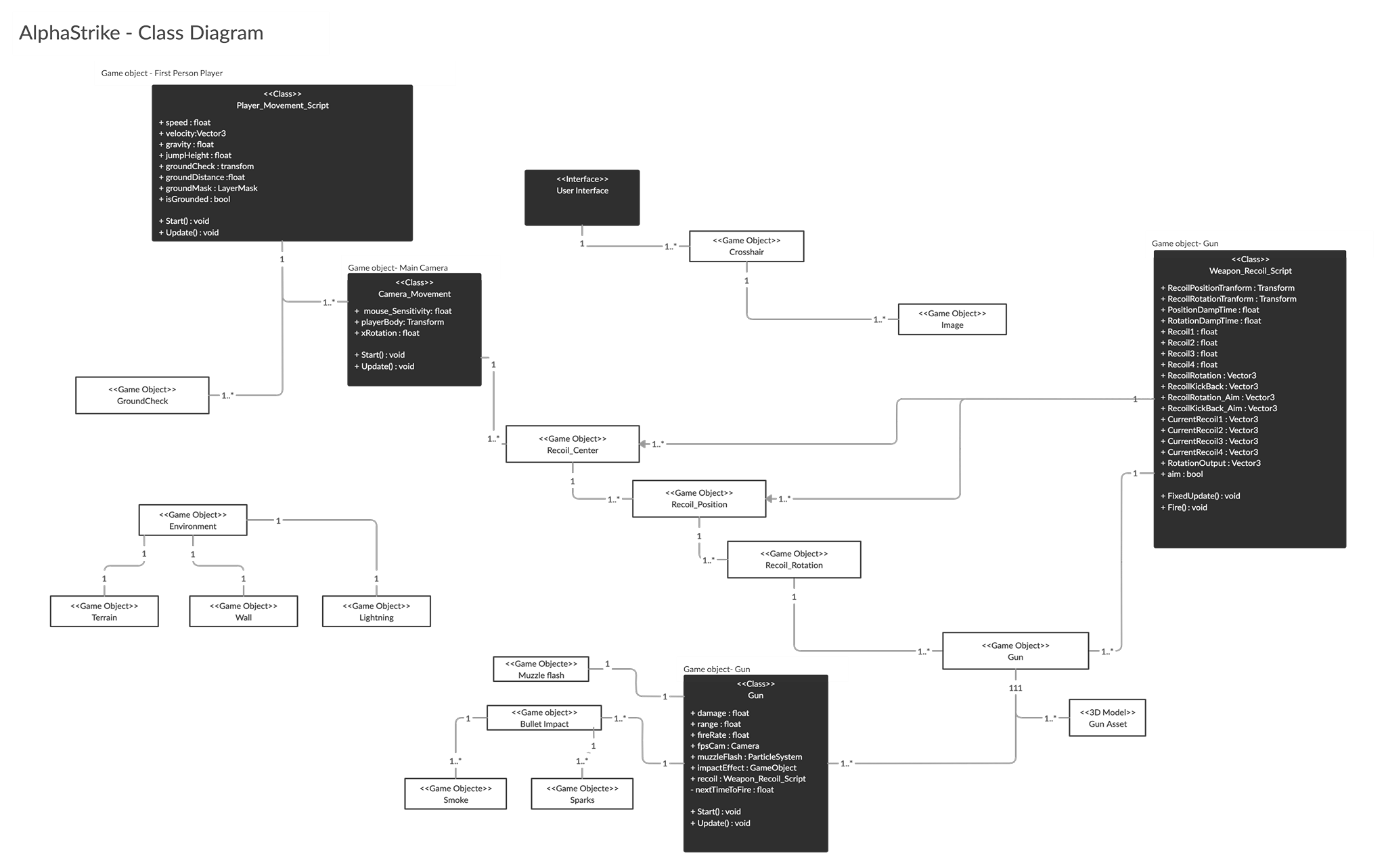
# 3. contenu du sprint actuel

## 3.1 Fonctionnalités implantées dans le sprint

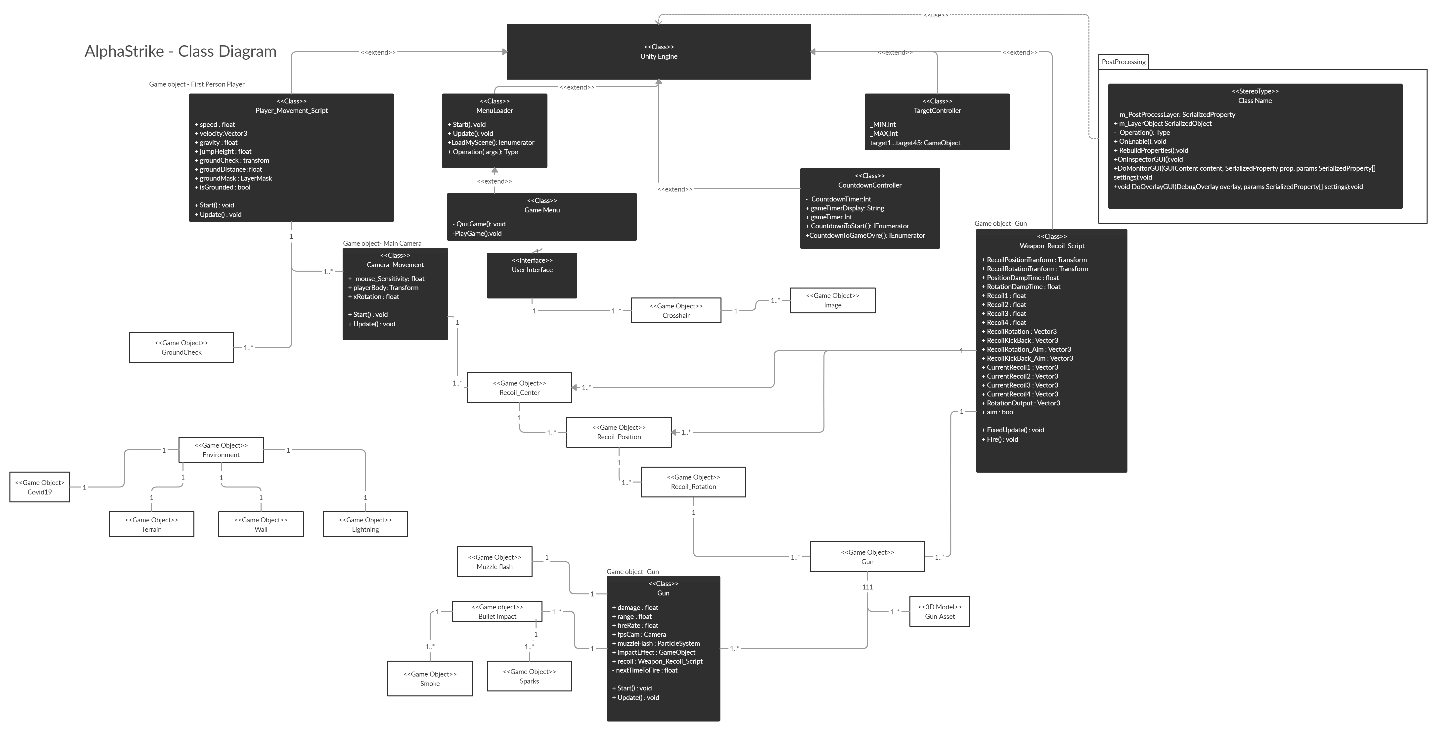
1. Améliorations
   1. Ajout d’une plus belle texture à l’environnement du jeu.
   2. Correction au niveau de bogues de mouvement de l’usager.
   3. Correction d’un bogue visuel infligeant des dégâts à l’usager avec sa propre arme à feu
2. Use case implantés
   1. La principale fonctionnalité ajoutée pour ce sprint est le système de tir sur cibles. L’usager peut démarrer une session de tir et accumuler des points selon ses performances en jeu.
   2. Ajout du menu, permettant à l’usager de naviguer de manière plus fluide sur l’application.
3. Autres "fonctionnalités" implantées
   1. Implantation du Menu de jeu. Permets à l’usager de faire le pont entre le jeu et l’instance du sandbox. Cette fonctionnalité permet aussi à l’usager de quitter une partie en cours pour retourner au menu principal.
   2. Détection d’une cible touchée par le système. De cette façon, ce dernier peut accumuler les points de l’usager.
   3. Ajout du mur de tir à l’intérieur de l’environnement Alpha Strike.
   4. Menu à l’intérieur de la zone de jeu, permettant à l’usager de sortir d’une partie en cours.

## 3.2 Modèle de classes

1. Sprint 1



1. Diagramme des classes global



* Les principaux ajouts
  + Le compte à rebours pour les cibles à tirer (CountDownController),
  + Les cibles au mur (TargetController)
  + Le menu du jeu (MenuLoader et GameMenu). Cette classe qui fait le pont entre l’application et le champ de tir
  + Un PostProcessing, qui modifie certains objets après instanciation a également été ajouté au diagramme de classe.
* Modifications
  + Game Engine de Unity, car la majorité de nos scripts héritent de ces fonctionnalités offertes par le cadriciel de développement.
  + L’héritage entre nos classes personnelles et le Game Engine
  + Petite modification sur l’environnement, qui contient maintenant une texture faisant référence à la Covid-19

# 4. Revue

## 4.1 Revue du sprint 23

Revue technique

* Ce qui a été complété :
  + Ajout de texture à l’environnement
  + Afficher les cibles sur le mur du sandbox
  + Détecter la touche de la cible par le joueur
  + Accumuler le pointage du joueur
  + Menu du jeu
* Ce qui n’a pas été complété :
  + Recharger son arme
  + Ajout des détails sur l’arme du joueur
  + Système de skin
  + Système de devise
* Ce qui a bien fonctionné :
  + Tester les changements dans l’application
* Ce qui n’a pas bien fonctionné :
  + Intégration du menu à partir d’un gabarit Unity
* Identifier les obstacles rencontrés et les solutions adoptées :
  + L’équipe a pris un certain temps avant de déterminer comment implanter un mur de tir dans l’application. Pour régler ce problème, nous nous sommes inspirés de développeurs ayant implémenté ce genre de jeux auparavant.

Revue gestion

* Évaluer les personnes
  + Bonne séparation du travail, équitable
* Bon déroulement
  + Tâches terminées à temps pour tous les membres de l’équipe
  + Accessibilité à des tutoriels en ligne pour aider au développement
* Améliorations potentielles
  + Établir des échéanciers plus clairs
* Créer un plan pour la mise en œuvre/adoption de ces améliorations
  + Mettre en place des plages communes de travail pour s’assurer de l’avancement du projet

## 4.2 Revue du projet

Revue technique

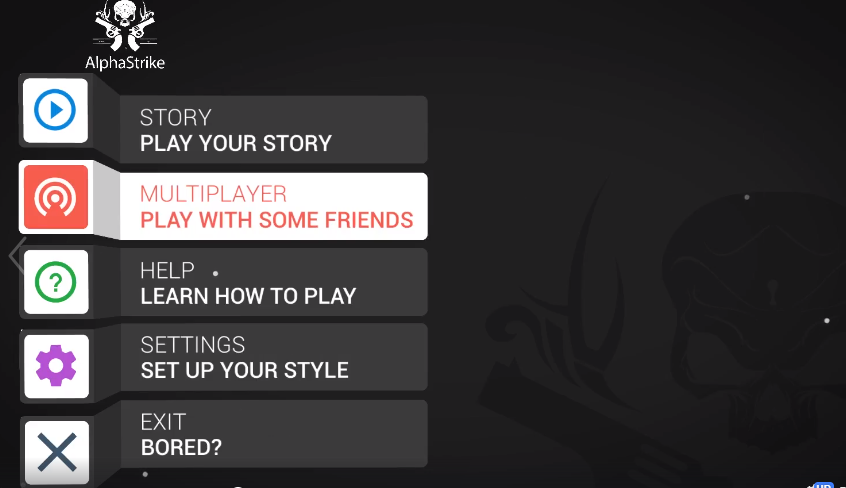
* Ce qui a été complété :
  + La totalité des fonctionnalités jugées essentielles
  + L’aspect réaliste de l’environnement de jeu
  + Menu du jeu
* Ce qui n’a pas été complété :
  + Les statistiques du joueur
  + Côté réaliste de l’arme à feu (recharger, couleurs)
* Ce qui a bien fonctionné :
  + Pour le projet, l’accès à de la documentation a été bénéfique pour s’orienter tout au long du développement
  + Tester l’ajout de code à l’application pouvait être testé immédiatement grâce au bouton play fourni dans l’environnement Unity
* Ce qui n’a pas bien fonctionné :
  + Travailler sur le même projet avec deux systèmes d’exploitations différentes (Apple, Windows) a souvent causé des conflits entre les versions de code
* Identifier les obstacles rencontrés et les solutions adoptées :
  + La collaboration via git a été difficile pour ce projet avec Unity, nous avons donc décidé d’utiliser Collab, un outil de contrôle de version fourni par l’environnement de développement

Revue gestion

* Évaluer les personnes
  + Bonne collaboration
  + Respect des échéanciers
* Bon déroulement
  + Communication simple et rapide entre les membres via Messenger
  + Bonne disponibilité des membres de l’équipe
* Améliorations potentielles
  + Ajouter une forme d’équité entre la séparation du travail (Écriture du rapport, développement de l’application)
* Créer un plan pour la mise en œuvre/adoption de ces améliorations
  + Établir la séparation des tâches en début de sprint afin que tous les membres de l’équipe touchent à tous les aspects du projet

## 5. ****Annexes****

Menu



In Game

