**Projet ISN 2017/2018: Cahier des charges**

**Nom du projet :** Mad Runner

**Date de début de rédaction du présent document :** 01/01/2018

**Date de dernière modification du présent document : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Elèves faisant partie de l’équipe de développement :** ADAM Ahmet, CIMBARO Mathéo.

**I/ Partie Générale :**

**Problématique du projet :** Création d’un jeu de plateforme d’athlétisme

**Quel est le but du programme ? Qu’apporte-t-il de différent par rapport aux programmes déjà existant ?**

Le but du programme est de créer un jeu divertissant, drôle et unique. Un jeu dont le principe est simple, mais qui est pourtant addictif.

**Introduction /Fonctionnalités/Mode de fonctionnement du programme :**

**Introduction:** Le jeu incarne un coureur Le joueur doit faire évoluer son personnage, peu sportif au début à travers diverses courses, avec différentes difficultés et sur différents terrains de courses.

**Fonctionnalités:**

Nom du jeu : Mad Runner (coureur fou)

Faire une introduction au jeu : Affichage du logo du jeu ainsi que de son titre de manière progressive

Menu avec 3 choix : Jouer, Statistiques, Paramètres

2 choix de jeu : 1 joueur ou 2 joueurs

3 modes de jeu : "400m", "400m haie", "Course infinie"

3 difficultés : facile, moyen, difficile

Appuyer sur des touches du clavier pour obtenir des avantages lorsque cette touche s'affiche sur l'écran

Avantages possibles : courir plus vite (mais énergie qui se finit plus vite), plus d'énergie, plus d'endurance (énergie qui se finit plus lentement)

Le coureur sera affiché en haut de l'écran (environ 3/4 de la hauteur), et les touches à appuyer en bas (prend le reste de la hauteur)

Barre d'énergie en haut au milieu

Vitesse en km × h-¹ en haut à gauche

Les avantages disparaissent au bout d'un certain temps, et plus la difficulté est élevée, plus ils disparaissent vite

Les avantages disparaissent plus vite au fur et à mesure qu'on avance dans la course

Vers la fin de la course, mettre plein d'avantages à la suite pour confondre le joueur surtout dans 400m haie

Dans "Course infinie", mettre les avantages de plus en plus vite

Mettre tout de même une limite du nombre d'avantages qui peuvent être mis à la fois

Si on se trompe de touche, on fait le contraire de l'avantage

Personnage qui évolue avec l'expérience : personnage en surpoids et non-sportif au début (court pas vite et dépense plus d'énergie), puis amélioration sur le personnage (perdre du poids, être plus sportif et donc utiliser moins d'énergie)

Différent décors pour la course (arrière-plan) : Jeux Olympiques, Athènes, Forêt

Dans "Course infinie", le coureur est coursé par quelque chose qui dépend du décor

Idées de courseurs: pour le décor Jeux Olympiques: une foule de personnes en colère, pour Athènes: un combattant avec une épée, pour la forêt: des loups

Dans "Course infinie", le jeu s'arrête lorsque le coureur est rattrapé par le courseur

Le courseur court de plus en plus vite

Le mode "Course" infinie" aura des obstacles qui dépendent du décor

Dans le "400m haie" et "Course infinie", appuyer sur Espace pour sauter et éviter les haies ou les obstacles, mais cela consomme de l'énergie. Si on rate une haie, on perd de la vitesse

Dans le mode 2 joueurs, le gagnant est celui qui arrive en premier à l'arrivé, et dans "Course infinie", c'est celui qui survie le plus longtemps

Le score dans "400m" et "400m haie" est le temps mis pour tout parcourir, et dans "Course infinie", c'est la distance parcourue

Tableau des scores en ligne (dans le jeu + dans un site web) du meilleur score parmi tous les joueurs), dans la section Statistiques (au menu principal)

Tableau des scores local (meilleur score personnel)

Possibilité de régler la musique du jeu (le désactiver ou l'activer)

**Mode de fonctionnement : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Matériels et logiciels nécessaire à la mise en œuvre du projet :**

**Logiciels:**

**-** PyCharm : programmer en Python

- Git : gestion des versions du projet

- ROBLOX Studio : Créer, animer et modéliser les personnages

- GIMP 2 : Édition es images des personnages

**Autres ressources nécessaires (documentation, apprentissages spécifiques etc…) :**

- Site internet GitHub pour envoyer les versions du projet

- Module Pygame de Python (non intégré)

- Vidéos sur Youtube (pour apprendre la Programmation Orientée Objet), le site Pixabay (images libres de droits)

**Ressources documentaires :**

* **Sites internet :**
* **Youtube.com**
* **OpenClassRooms.com**
* **Magazines :**
* **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**
* **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**II/ Partie algorithmique :**

* **Fonction principale :**

**Fonction qui gère le lancement du jeu (initialisation) et sa fermeture**

* **Objectifs de la fonction :**
* Initialisation du model, de la vue, et du contrôleur.
* Détermination de l’état de la fenêtre pour savoir si celui-ci est fermée.
* **Description du mode de fonctionnement :**
* Importe les modules et scripts nécessaires, initialise les scripts (Model, View, Controller)
* Communique avec le Controller pour la gestion des contrôles
* Communique avec le View pour la mise à jour des éléments graphiques

**Fenêtres utilisées :**

PyGame

* **Contraintes :**

Faire un chargement rapide et efficace

* **Répartition de cette tâche au sein du groupe :**

1 seul personne, Ahmet, a coder la base du script puisqu'il est très court, avec éventuellement des optimisations apportés par Mathéo

**Structure suivante à copier autant de fois que nécessaire :**

* **Sous fonctions :**

**Model**

* **Objectifs de la fonction :**

La fonction a pour objectif de s'occuper des informations sur les objets, de les traiter, les communiquer vers la partie "View", de recevoir les changements causés par la partie "Controller"

* **Description du mode de fonctionnement :**

Classe "Model" qui reçoit grâce à des méthodes, les changements à faire au niveau des informations sur les objets (coureur, boutons, état du jeu)

* **Fenêtres utilisées :**

PyGame

* **Contraintes :**

Être organisé, bien gérer les objets (ne pas oublier de les effacer quand ils ne sont plus nécessaires, ne pas mettre d'attributs inutiles), bien communiquer avec le "View" pour la modification d'éléments dans la fenêtre.

* **Répartition de cette tâche au sein du groupe :**

**Mathéo**: Gérer les changements d'attributs, d'état du jeu

**Ahmet**: Communication des changements vers la partie "View"

* **Sous fonctions :**

**Controller**

* **Objectifs de la fonction :**

La fonction a pour objectif d'afficher dans la fenêtre tout élément visible par

l'utilisateur, c'est la partie "View" du pattern Model View Controller

* **Description du mode de fonctionnement :**

Classe "View" qui contient des méthodes permettant de créer les éléments visibles sur l'écran (boutons, arrière-plan, personnage)

* **Fenêtres utilisées :**

PyGame

* **Contraintes :**

Être organisé, soigner l'apparence des objets, bien communiquer avec le "Model" pour la modification d'éléments dans la fenêtre.

* **Répartition de cette tâche au sein du groupe :**

**Ahmet**: Détections des boutons cliqués, gérer la communication entre le "Model" et le "Controller"

**Mathéo**: Création des boutons, design des boutons

* **Sous fonctions :**

**Controller**

* **Objectifs de la fonction :**

La fonction a pour objectif de détecter toute interaction du joueur avec le jeu (clics de souris, fermeture de la fenêtre, appuies sur les touches du clavier)

* **Description du mode de fonctionnement :**

Classe "Controller" qui détecte ces interactions, puis communique au Model pour savoir si cette interaction change quelque chose au "Model", puis au "View**"**

* **Fenêtres utilisées :**

PyGame

* **Contraintes :**

Être organisé, bien communiquer les informations, ne rien faire d'inutile, bien communiquer avec le "Model" pour la modification des attributs des objets.

* **Répartition de cette tâche au sein du groupe :**

**Mathéo:** Détection des touches appuyées

**Ahmet:** Communication de l'information