

Mini projet

Table des matières :

I) Partie Générale

1) Page de titre..... page 2

- Titre du projet
- Nom des étudiants/ membre de l'équipe
- Date de soumission
- Nom du cours et de l'enseignant

2) Résumé.....page 2

- Brève description du projet et des principaux résultats

3) Introduction.....page 2

- Présentation du contexte
- Objectif du logiciel

4) Analyse des besoins.....page 2-3

- Description détaillée des besoins du logiciel
- Spécifications fonctionnelles
- Spécifications non fonctionnelles

5) Planification et répartition des tâches.....page 3

- Utilisez et commentez le diagramme de gantt

II) Partie personnelle

1) Conception.....page 4-5

- Diagramme UML
- Architecture général du logiciel (fonctionnement de git hub)
- choix technologiques

I)Partie générale

1) Page de titre

- Nom de projet : STEVE
- Nom des étudiants : TISON Elise MARECHAL Jose
- Date de soumission : ... decembre
- Nom du cours et de l'enseignant : Développement d'application avec Monsieur Tomczak

2) Résumé

- Description du projet : un jeu 2D où un personnage doit sauter à temps pour éviter les obstacles qui arrivent vers le personnage.

3) Introduction

- Présentation du contexte : créer un logiciel divertissant et amusant dans le style du jeu « Steve »
- Objectif du projet : créer un jeu similaire à STEVE qui soit fonctionnel et le plus fluide possible

4) Analyse des besoins

- Descriptions détaillée des besoins du logiciel :
 - Interface de lancement du jeu
 - Créer personnage
 - Créer obstacle
 - Créer points
 - Créer les mouvements du perso
 - Créer le placement des obstacles
 - Créer les mouvements des obstacles
 - Changement de points selon le score
 - Interface de fin
- Spécifications fonctionnelles :
Perso :
 - Saute
 - Animation de course (boucle d'image pour une impression de mouvement)
 - Apparaît tout à gauche
 - Meurt quand il touche un obstacle

Obstacles :

- Change de taille aléatoirement (augmente la difficulté)
- Apparaît de façon irrégulière
- Apparaît tout à droite
- Mouvement de droite à gauche

Score :

- Dès qu'on franchit un obstacle, on ajoute 1 au score

- Spécifications non fonctionnelles

Perso :

- Ne peut pas se déplacer

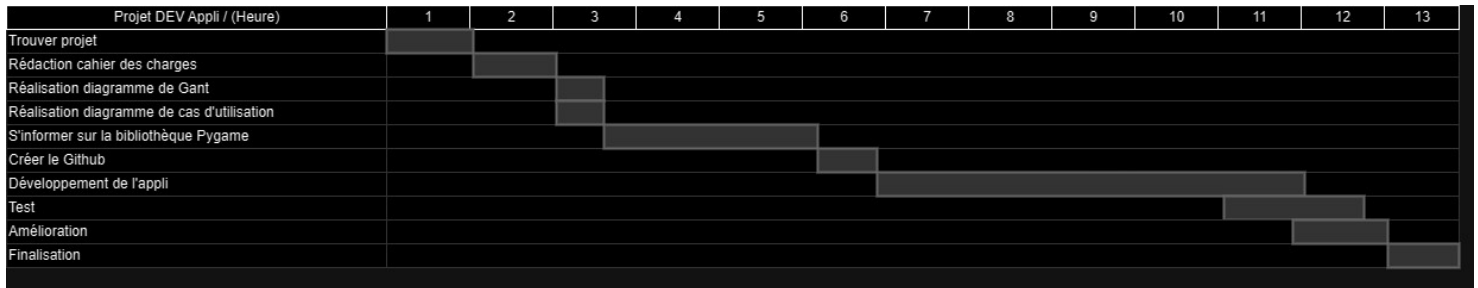
- Ne peut être positionné qu'à gauche de l'écran
- Ne peut pas effectuer de double saut

Obstacle :

- Ne peut pas avoir une trop grande taille sinon le perso ne peut pas passer
- Apparaît forcément à gauche

5) Planification et répartition des tâches

Diagramme de gantt :

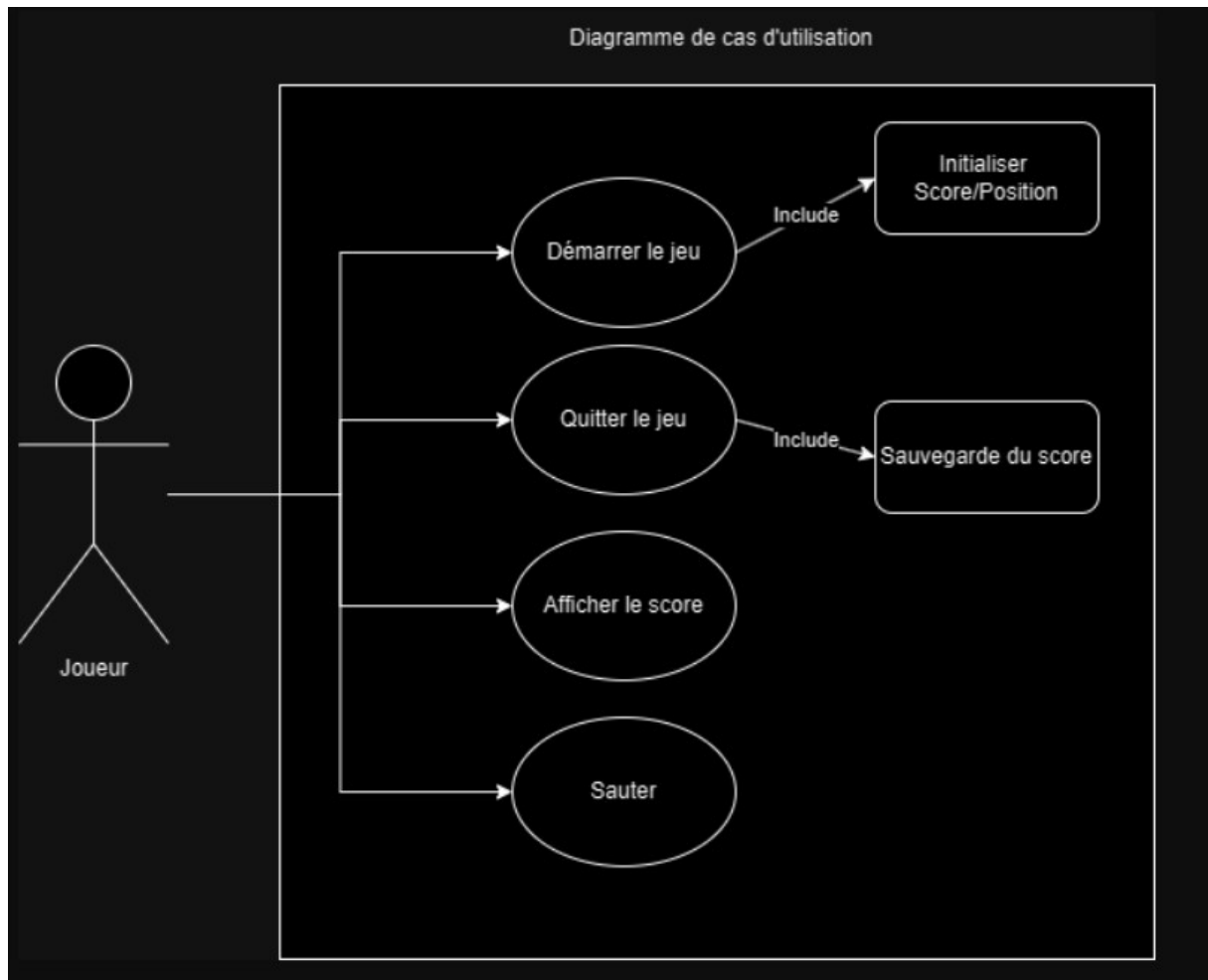


Afin de répartir les différentes tâches du projet, nous avons effectué certaines tâches ensemble et d'autres séparément dans l'optique d'optimiser le temps de projet. Le diagramme de gantt réalisé ci-dessus et séquencé en heure pour une répartition plus précise. Nous avons d'abord trouvé une idée de projet et rédiger le cahier des charges à 2. Ensuite, pendant que l'un réalise le diagramme de Gantt, l'autre s'occupe de réaliser le diagramme UML. L'information de la bibliothèque pygame a pris 2h30 car c'était une nouvelle notion à connaître et que nous avons encore jamais vu donc la découverte de celle-ci a pris quand même pas mal de temps. Idem pour le fonctionnement de git/github. Enfin, il ne manque plus que la partie programmation qui sera la plus longue car c'est le cœur même du projet. Pour finaliser le projet, il ne reste plus qu'à effectuer des tests et essayer de l'améliorer au maximum pour avoir le meilleur résultat possible. La finalisation du projet consistera à la relecture et aux dernières modifications de celui-ci.

II) Partie personnelle

1) Conception

Diagramme UML :



Description:

1. Cas d'utilisation : Démarrer le jeu
-Description : Permet au joueur de démarrer le jeu.
-Inclusion : Initialiser score et position.
2. Cas d'utilisation : Quitter le jeu
-Description : Permet au joueur de quitter le jeu.
-Inclusion : Sauvegarder le jeu.
3. Cas d'utilisation : Afficher le score
-Description : Permet au joueur de visualiser son score actuel.
4. Cas d'utilisation : Sauter
-Description : Permet au joueur de faire sauter le personnage dans le jeu.

Chaque cas d'utilisation est représenté graphiquement dans un diagramme de cas d'utilisation, avec des acteurs (comme "Joueur") et des relations d'inclusion pour montrer les dépendances entre les différents cas d'utilisation.

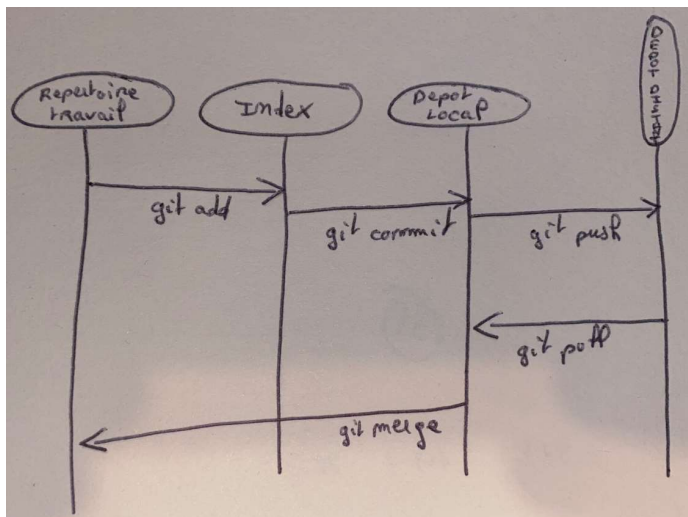
Architecture général du logiciel fonctionnement de git hub :

Git permet de suivre et enregistrer les modifications de notre projet au fil du temps dans un dossier distant (cela évite de tout perdre et de partager plus facilement entre personne du groupe).

Il faut au préalable avoir créé un compte github, et par la suite télécharger git sur son pc.

Dans le git bash:

- Il faut d'abord commencer par initialiser un dépôt sur notre machine avec "git init".
 - Ensuite ajouter des fichiers dans le dépôt initialisé avec "git add".
 - Et enregistrer les modifications avec "git commit -m 'message'".
 - Et finalement envoyer sur un serveur distant avec "git push" (en relation avec le github grâce à "git remote add origin https://github.com/nom_utilisateur/nom_depot.git").
- Pour récupérer les dernières modifications, on utilise un "git pull".
Le "git status" permet de voir l'état de nos fichiers.



Choix technologiques :

Langage de programmation utilisé : Python (avec utilisation de la bibliothèque pygame sur le logiciel pycharm)