Documentation intermédiaire du projet

Orlann Ferreira, Tommy Brisset, Yann Gellner, Thibault Choubry

Sommaire

1. Bilan de ce qui a été réalisé
2. Fonctionnalités à développer
3. Justification des choix
4. Répartition des tâches
5. Outils de travail et de communication au sein du groupe
6. Bilan de ce qui a été réalisé

Table

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generatedPour commencer, nous nous sommes concentrés sur la création des pages du jeu ainsi que de leurs apparences visuelles.  
Ce travail a donc été réalisé sur la page d’accueil du jeu, la page du choix des options et la page contenant le plateau de jeu.

Tout d’abord, pour la page d’accueil, nous avons réfléchi au type de graphisme à utiliser. Nous avons donc choisi un graphisme de type PIXEL ART.  
Une image de fond a donc été inséré ainsi qu’un bouton « start » permettant d’aller vers la page des « settings » soit les paramètres du jeu.   
Ensuite, sur la page des choix des paramètres de jeu, plusieurs options sont disponibles pour adapter le jeu à nos envies, donc plusieurs configurations de parties possibles.  
Nous pouvons choisir le nombre de joueurs, la taille du plateau, ainsi que le nombre de barrières total de la partie. Nous avons implémenté ces différents choix sous la forme de « boutons » ou il faut cliquer dessus et une animation se déclenche pour montrer visuellement que le bouton a bien été activé. On retrouve, également, un bouton « reset » qui permet de remettre les différents paramètres par défaut.   
Pour pouvoir lancer une partie, nous avons implémenter un bouton « jouer » qui permet de passer à la page de jeu avec le plateau. Le bouton « réseaux » permettra de lancer une partie en réseau. (Cette fonctionnalité n’est pas encore au point).

Après avoir réalisé ces pages qui nous ont servies de base pour la réalisation du code, nous sommes passé à la création de l’algorithme de jeu, avec son interface graphique. Il s’agit donc des différentes fonctions qui servent à appliquer les règles du jeu.   
Nous pouvons donc nous déplacer de cases en cases, selon les règles établies, grâce à un clic droit sur la case voulue. Nous pouvons aussi placer des barrières sur les emplacements libres, (l’algorithme de pathfinding n’a pas encore été réalisé). Une fois que l’un des joueurs est arrivé sur la ligne adversaire, le programme s’arrête et indique le vainqueur dans la console.   
Nous avons aussi créé une fonction qui permet de naviguer entre les pages et de conserver les informations entre celles-ci, comme par exemple lorsque l’on passe de la configuration de la partie vers la page de jeu.

1. Fonctionnalités à développer

Les différentes fonctionnalités du code qu’ils nous restent à développer sont :

- Faire la partie réseau du projet, c’est-à-dire création d’un serveur permettant à différents ordinateurs de se connecter entre eux et pouvoir jouer au jeux ensembles.

- Création d’une IA afin de pouvoir jouer contre l’ordinateur. (si possible nous essayerons de développer une IA qui fait des mouvements aléatoires et une IA plus avancée et qui joue intelligemment)

- Une optimisation du code afin que les performances du jeu soit efficace (pas de ralentissement lors du jeu), éviter d’avoir des bugs et garantir un bon fonctionnement du jeu.

- Faire un algorithme de « pathfinding » qui permet de vérifier si un joueur peut toujours jouer (un joueur ne doit pas être bloqué entre 4 murs, etc…), il est important qu’une partie ne soit pas bloqué pendant que les joueurs jouent.

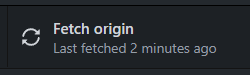
1. Justification des choix

Pour le langage, nous avons choisi le langage de programmation python car il s’agit du langage le mieux maitrisé dans le groupe par rapport aux autres langages utilisables.

Pour la librairie utilisée, nous avons choisis pygame car elle nous semblait être la plus intuitive et pratique pour notre code et qu’elle permet une meilleure gestion des éléments affichés à l’écran. De plus, les fonctions pré implémentées au module pygame nous semblaient être un avantage dans la réalisation de notre projet.

Nous voulions utiliser pygame car il nous permettait aussi d’utiliser une nouvelle librairie et donc de découvrir une autre manière de coder graphiquement.  
L’utilisation de pygame, nous mettait face à un défi car nous n’avions presque jamais utilisés cette librairie graphique.

Pour le partage du code entre les membres du groupe, nous avons choisi de mettre le code à disposition sur GitHub car il s’agit d’une des méthodes les plus simple et efficace pour se partager un code. Tous les membres du groupe ont installés GitHub desktop pour que le partage et la mise à jour du code soit accessible à tous rapidement. En effet, il suffit juste de faire un « commit » puis un « pull » du code, qui permet d’envoyer son code lorsqu’on a modifié celui-ci ou bien, de récupérer le nouveau code quand quelqu’un l’a corrigé.



Il suffit juste d’ouvrir notre IDE via l’application afin d’avoir le nouveau code.

1. Répartition des tâches

Pour la répartition des tâches, nous avons déterminé des points à régler afin de faciliter le travail de chacun, et, aussi pour que l’organisation du travail soit le plus simple possible. Nous avons aussi fait en sorte de se répartir les activités en fonction des facilités et des connaissances de chacun.

Orlann s’occupe de la partie réseau et de l’algorithmie des différentes règles du jeu qui permet donc de jouer et lancer une partie de jeu.

Tommy s’occupe de l’affichage d’une grande partie des fenêtres du jeu, il a réalisé la page du choix des paramètres de partie et il est aussi à l’origine du premier modèle du jeu (squelettes, maquette) et des transitions entre les différentes pages du programme.

Thibault s’occupe de la partie IA afin de pouvoir jouer une partie contre l’IA, de l’affichage du plateau de jeu par rapport à la fonction game et du fonctionnement des barrières et le pathfinding avec Yann.

Yann s’occupe de l’identité graphique du projet (images, designs…), de l’affichage des autres fenêtres du jeu et aussi de l’algorithme de pathfinding.

La répartition des tâches n’était pas vraiment au point au début du projet ce qui était un peu difficile quand on a dû commencer à coder mais nous avons discuté et nous nous sommes organisé comme il fallait afin que celle-ci soit au point et que nous puissions avancer efficacement.

1. Outils de travail et de communication au sein du groupe

Pour optimiser la communication dans le groupe, nous avons mis en place des solutions comme le serveur discord et le GitHub. Nous avons aussi désigné un chef de projet Tommy, qui s’occupe de la communication entre le groupe et l’étudiant référent.

On peut tous communiquer entre nous à tout moment grâce au serveur discord que nous avons créé pour le projet. Le GitHub nous permet de synchroniser le code sur nos IDE tous ensemble, simplement en l’uploadant via l’application GitHub desktop.

Dès que quelqu’un à avancer sur le projet, on le dépose sur le GitHub ou le serveur discord afin que les autres puissent observer les avancées de chacun et que l’on puisse aussi se baser sur le travail des autres pour avancer sur le projet dans sa globalité. Il faut aussi informer Tommy afin qu’il soit au courant des avancées du projet et qu’il puisse poser les questions nécessaires par rapport aux problèmes que l’on a rencontré pendant le développement du jeu.

Grâce au discord et au GitHub, on peut travailler à distance sur le projet mais le meilleur aspect pour s’organiser en groupe reste lors des journées dédiées au module qui nous permettent de travailler ensemble sur le projet.

Nous avons aussi pu communiquer directement entre nous dans le groupe discord afin de faire le point à deux ; ou pour résoudre un problème de l’un ou de l’autre grâce à l’aide d’un membre du groupe , qui lui pouvait apporter une solution et permettre de continuer d’avancer sur le projet.