

Rapport de développement/test
Nathan Menny
08/06/2020

V.2
Groupe 2.1

Antoine Tartare
Sophie Scheidt
Karina Teyssere
Nathan Menny

Rapport de développement/test

Sommaire :

I. Synthèse du développement

- 1. Comparaison objectif / résultat*
- 2. Utilisation de l'application*

II. Descriptions des tests unitaires réalisés

I. Synthèse du développement

1 – Comparaison objectif / résultat

Au début du projet nous voulions créer un jeu de plateforme en 2D avec la possibilité de déplacer notre personnage, de le faire sauter ainsi que de pouvoir avoir autant de niveaux que nous le souhaitons. Nous avons donc réussi sans difficulté à implémenter ces différentes fonctionnalités.

Cependant, nous avons dû abandonner plusieurs fonctionnalités ; la principale étant l'option de menu qui nous aurait permis de changer la taille de l'écran ainsi que le volume du jeu. La cause principale étant le manque de temps induit par la crise du coronavirus.

Nous avions également prévu d'inclure en bonus dans notre jeu divers éléments de *gameplay* comme des ennemis, des trampolines pour sauter plus haut et des ventilateurs qui nous auraient poussé dans certaines directions mais par manque de temps nous n'avons pas pu les réaliser.

Malgré tout, nous avons quand même un jeu fonctionnel avec les fonctionnalités de base implémentées.

2 – Utilisation de l'application

Afin de lancer l'application, il faut exécuter la méthode *main* de la classe *GameLauncher* (aucun paramètre n'est à fournir). Une fois l'application lancée, une fenêtre va s'ouvrir montrant l'écran d'accueil du jeu. Si vous souhaitez commencer à jouer, sélectionnez le menu « start ».

Vous incarnerez dorénavant un carré vert et votre objectif est d'atteindre le carré bleu qui vous permettra d'aller au niveau suivant. Sur votre chemin vous trouverez des carrés rouges symbolisant des fraises, récoltez en 10 afin de gagner une vie.

Les contrôles :

→ Dans les menus :

- . *Flèches directionnelles* : navigation dans les options de menu
- . Touche *entrée* : ouvrir l'option de menu sélectionnée
- . Touche *echap* : retour au menu principal

→ Dans le jeu :

- . *Flèches directionnelles* : déplacer le joueur à l'écran
- . Barre *espace* : faire sauter le joueur
- . Touche *echap* : retour au menu principal

II. Description des tests unitaires réalisés

Nous avons testé principalement la classe *Game* car c'est elle qui permet de faire le lien en l'interface et l'application.

Afin la tester nous avons effectué des tests sur ses différentes méthodes :

→ La méthode *checkCollisions()* a probablement été la plus testée car il s'agit de la plus complexe ; en effet, c'est elle qui vérifie la collision du joueur avec le décors, il était donc nécessaire de la tester. Afin de vérifier qu'elle fonctionne correctement, nous avons créé un niveau avec toutes les dispositions possibles de blocs ce qui nous permis de vérifier les collisions sur tous les côtés (nous n'avons pas pu la tester avec Junit).

→ La grande majorité des méthodes nous permettant de rentrer des valeurs (*chooseLevel()*, *displayLevel()*, *setCurrentSelection()*) ont été testé en essayant d'entrer des valeurs trop grandes ou trop petites ; c'était important de les tester car l'utilisateur y a accès (vu qu'elle sont publiques) et qu'il peut fournir des valeurs à la méthode.

L'autre classe ayant été testée est *Level*, qui regroupe toutes les méthodes permettant de charger, d'afficher un niveau et de placer le joueur. On a dû la tester car elle contient des méthodes lisant le contenu de fichiers pouvant être modifiés par l'utilisateur.

→ La méthode *loadAllLevels()* a été testée car elle permet de charger tous les niveaux en lisant un fichier. Afin de s'assurer qu'elle réagit correctement, nous avons modifié le fichier dont elle a accès (fichier vide, fichier avec trop de lignes...) et regardé si elle générerait bien les bonnes erreurs.

→ Nous avons effectué le même test sur la méthode *load()* qui permet de charger un niveau en particulier pour les mêmes raisons que la méthode précédente.

→ La méthode *getInitialPlayerPosition()* a été testé en long, en large et en travers afin de s'assurer que le joueur serait bien toujours présent et qu'il apparaîtra toujours dans l'écran peu importe la position donnée par le niveau. En effet, vu que c'est le fichier *leveln.txt* (où n est le numéro du niveau) qui contient la position initiale du joueur ; l'utilisateur peut décider de le mettre n'importe où, il a donc fallu s'assurer que peu importe la position initiale du joueur, on devait l'avoir à l'écran tout le temps. Nous avons donc testé toutes les configuration possibles pour la position du joueur et vérifié qu'il était toujours visible.

La dernière classe à avoir été testée est *IHM_Player* qui gère l'affichage et les interactions avec l'utilisateur.

→ Tout comme dans *Game*, on a testé la méthode qui permettait à l'utilisateur de rentrer certaines valeurs : *displayMenu()*. Pour cela, on a donc tester les cas où la valeur donnée était trop petite, trop grande et quand elle était correcte.