|  |
| --- |
| **15 juin 2020**  **Développement d’un jeu vidéo en java**  BLIER Lukas  CARTERET Thomas  MENUDIER Jeoffrey  Sous la tutelle de M. Creput |





Sommaire

Table des matières

Sommaire 3

I – Introduction 4

II – Analyse du problème 5

**1)** **Diagramme de cas d’utilisation** 5

**2)** **Diagramme de classe** 6

**3)** **Diagramme d’état-transition** 7

III – Choix de conception 8

IV – Présentation du jeu 9

V – Améliorations possibles 10

VI – Conclusion 11

I – Introduction

Dans le cadre de l’UV LO43 visant à apprendre la programmation orientée objet à travers les langages JAVA et C++, nous avons eu l’opportunité de réaliser un projet de fin de semestre. En effet, il s’agit d’un petit projet ayant pour but de mettre en application nos connaissances en JAVA. C’est pour cela que nous avons choisi de développer un jeu. Il s’agit d’un projet assez ludique dans lequel nous étions très intéressés. Avec les années nous avons pu jouer à divers jeux sans vraiment appréhender en profondeur leur fonctionnement, cela a donc été l’opportunité parfaite. De plus, JAVA est un langage très utilisé dans ce domaine ce qui a renforcé notre choix.

Cependant, il y a une très grande variété de jeu, nous avons choisi de créer un jeu de plateforme 2D. Il s’agit en effet d’un type de jeu abordable à des programmeurs débutants et qui n’implique pas l’utilisation de bibliothèques graphiques très développées ce qui veut dire que l’on peut plus facilement tout faire nous-même et donc mettre en jeux les notions abordées en cours.

Le cadre plus précis du jeu se construit autour d’un univers utopiste parodiant le fonctionnement de l’UTBM à travers un jeu de plateformes traditionnel. Nous avons cependant décidé de ne pas beaucoup développer cet aspect-là car il ne représente pas le but initial du projet.

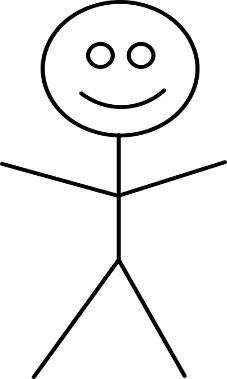
II – Analyse du problème

Après avoir défini notre projet et son contexte, il a fallu l’approcher plus en profondeur et notamment identifier les contraintes auxquels il devra répondre ou encore définir une architecture optimale. Pour gagner du temps et éviter les erreurs de conception, nous avons utilisé dans cette démarche préliminaire le model de représentation UML.

1. Diagramme de cas d’utilisation

Ce diagramme a pour but de finaliser l’étape de définition des et des exigences pour analyser le système et prévoir la conception. Il permet de déterminer les manières avec lesquelles les acteurs utiliseront le système.

Jeu 2D platformer



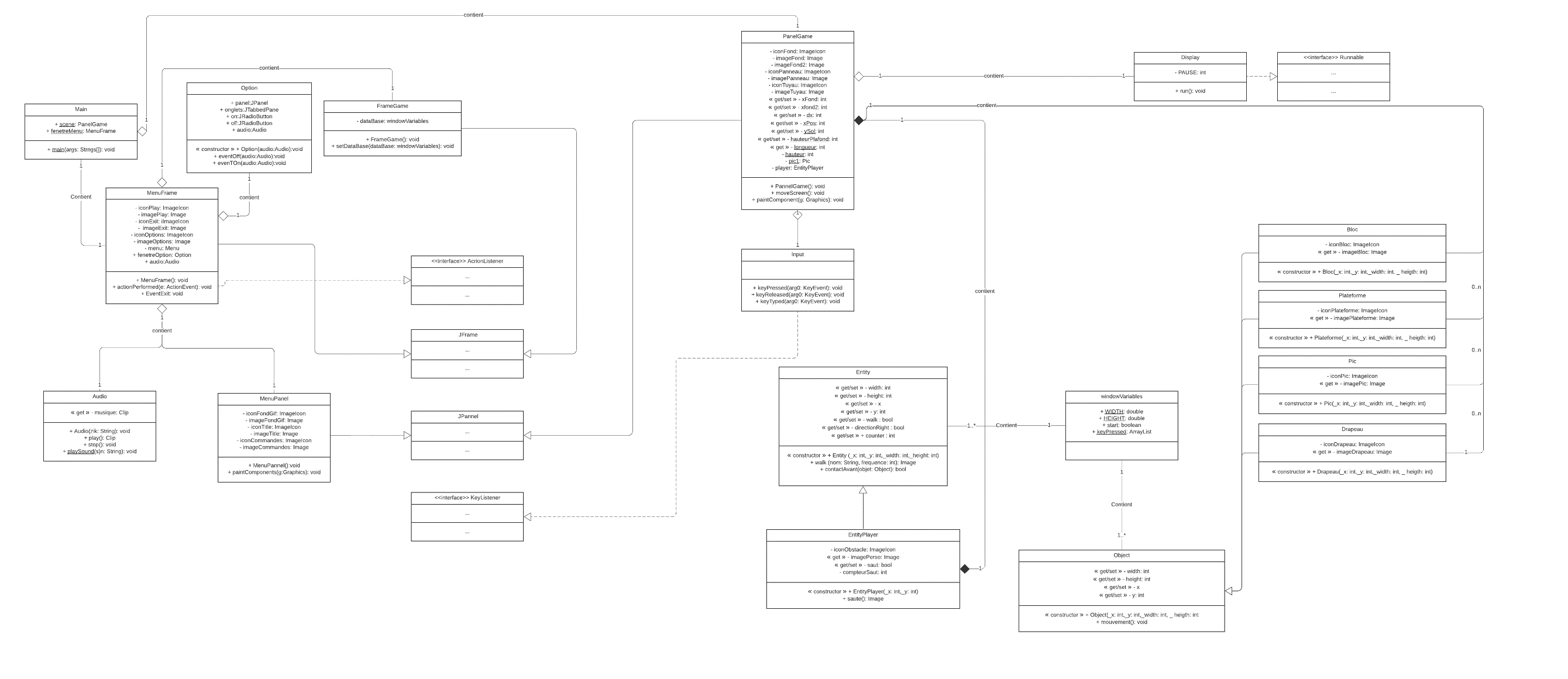
Joueur

Régler les paramètres

Jouer au jeu

1. Diagramme de classe

Après avoir terminé cette étape préliminaire, afin de concevoir le système, nous avons réalisé le diagramme de classe. Il permet en effet de préciser les informations qui seront nécessaires à l’utilisateur. De plus, il permet de définir l’architecture globale du jeu, il fournit ensuite une ossature à compléter permettant la création du logiciel. Il nous a notamment permis de mettre en valeur et de respecter la contrainte d’une architecture MVC. Cette aide visuelle nous a permis d’éviter certaines erreurs de logique globale comme par exemple mettre une variable nécessaire à tout le logiciel en dehors de la partie ‘model ‘.



1. Diagramme d’états-transition

D’une part pour mettre en pratique ce type de diagramme, d’une autre pour faciliter la compréhension globale du programme, nous avons choisi de réaliser un diagramme d’états-transition. Il permet en effet de bien appréhender l’évolution d’une instance de classe clef au fonctionnement du programme. Contrairement au diagramme de classe il est plus abstrait, plus éloigné du code.

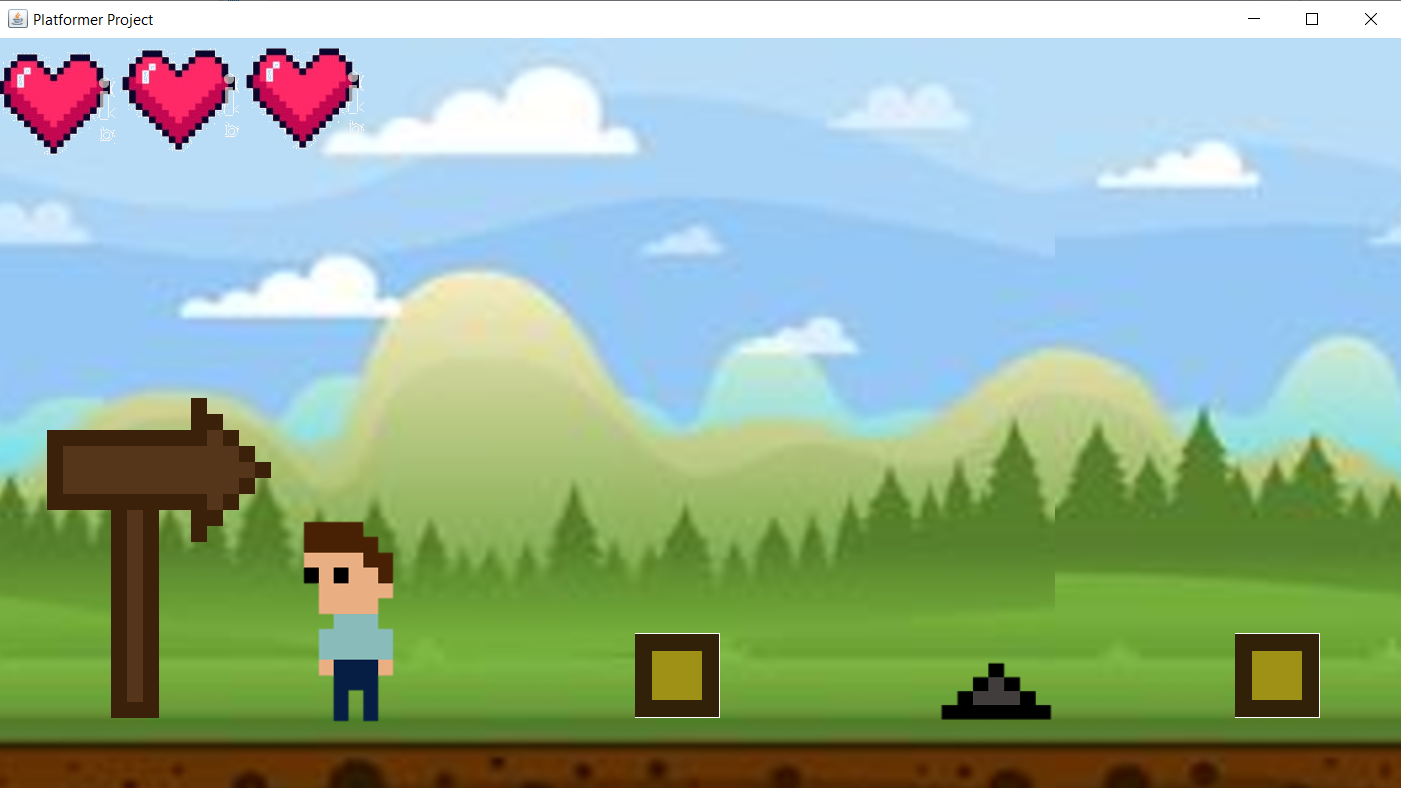
III – Choix de conception

IV – Présentation du jeu

Lors du développement du jeu, nous avons choisi de suivre les codes des platformers usuels. Le temps étant assez court, nous n’avons pas pu nous lancer dans la création d’une charte graphique pour créer un univers. C’est pour cela, que nous avons pris des images sur internet en ajoutant quelques éléments de personnalisation. Le seul élément que nous avons créé entièrement est le personnage qui a été dessiné grâce à Piskel.

Une fois le jeu lancé grâce à La\_Quete\_Du\_Graal.jar, nous nous retrouvons sur la page d’accueil suivante.

S’offre alors à nous une fenêtre composée de trois boutons cliquables. Le premier ‘Play’ permet de lancer le jeu, le second n’a malheureusement pas encore abouti, il est censé amener à une page de paramétrage inexistante pour le moment, Le dernier ‘Exit’ permet de quitter le jeu. On peut aussi apercevoir le nom du jeu, un .gif donnannt un apperçu du jeu ainsi que les commandes pour pouvoir y jouer.

Une fois sur la fenêtre de jeu, nous pouvons nous déplacer grâce à ‘Q’ et ‘D’ et sauter grâce à la barre d’espace, pour un saut en forme de cloche il est recommandé d’appuyer simultanément sur le saut et la direction pour faire un saut en forme de cloche. Le but est alors de rejoindre le drapeau en fin de niveau en évitant les piques faisant perdre des points de vie.

V – Améliorations possibles

A cause des cours délais pour réaliser ce projet, la version du projet livrée n’est pas tout à fait fini, il reste des bugs à corriger. Nous avons pu en traiter la majorité notamment autour des collisions mais il en reste encore. Contrairement à ce que nous avions pu penser, les collisions ne consistent pas simplement à un set de vérification, il est très facile avec cette méthode de se coincer dans l’objet à cause de l’inertie du personnage. Il faudrait donc se pencher plus profondément sur ce projet en anticipant les collisions par exemple ou alors en utilisant des méthodes de développement de jeu vidéo plus professionnel, c’est-à-dire en utilisant des bibliothèques plus complètes permettant notamment d’enregistrer un terrain et donc de ne pas se préoccuper des collisions. Avec cette méthode il suffit d’instancier des classes correspondant à des entités qui ont certaines propriétés modifiables grâce à l’héritage.

De la même façon, pour perfectionner le jeu, nous pourrions envisager l’ajout d’un menu de gestion de paramètres pour par exemple gérer les FPS pour éviter un rafraîchissement trop important ou trop faible. Afin d’améliorer l’immersion, il faudrait aussi gérer des événements qui correspondraient à des bruitages ou à des animations suite à certains évènements. Dans ce même esprit, et du au peu de temps que nous avons eu, il aurait été souhaitable de créer un univers, une charte graphique. Sans cela, il s’agirait d’un projet où on perdrait le fil conducteur à savoir la pensé orientée utilisateur, s’agissant d’un projet scolaire, cet aspect n’a pas été traité.

VI – Conclusion

A travers ce projet, nous avons pu retirer de très nombreuses connaissances et compétences complémentaires au cours. En effet, dans cette partie du semestre orientée sur la programmation orientée objet en Java, le projet constituait l’essentiel des TP et donc de la mise en pratique du langage. Nous avons aussi pu constater à quel point les langages orientés objets se ressemblaient. C’est pour cela que nous avons si rapidement pu être opérationnelle dans ce langage. Nous avons pu aussi profiter de la plus grande souplesse de ce langage pour se focaliser sur l’apprentissage des nouvelles structures et surtout des bibliothèques. Nous avons découvert à nos dépend à quel point les bibliothèques de jeux vidéo pouvaient être complexes à appréhender. De plus, nous avons réellement constaté à travers l’utilisation de ces bibliothèques l’intérêt de la programmation orientée objet et à quel point elle était utilisée. Tous les objets de jeux tel que les ‘Frame’ ou encore les ‘ Pannel’ utilisaient ce mécanisme.

Pour ce qui est de l’expérience retirée, nous en avons tous retiré une bonne expérience pour des raisons très variées. Parmi les plus importantes nous retrouvons évidemment le contexte particulier dans lequel s’est déroulé le projet. La crise du covid-19 a bien sur eu un impact important sur le déroulement du semestre et donc du projet. Nous avons pu développer des compétences autres à la programmation notamment au niveau de l’autonomie qui pourront nous être très utile en stage. D’un point de vue de l’apprentissage du langage Java, notre projet concernant des bibliothèques très spécifiques nous avons pu apprendre à nous en servir notamment sur internet avec divers tutos. Pour ce qui est du langage en règle générale, nous n’avons pas eu de difficulté particulière car grâce au Tps de C++ effectués au préalable, nous n’avons dans une globalité pas eu de difficultés particulières.

Le fait que les cours soient à distance pour une matière telle que l’apprentissage de la programmation orientée objet