



PRESENTATION PROJET

CRYPT OF THE JAVADANCER



OUT OF BOUNDS

IUT DIJON-AUXERRE

TABLE DES MATIERES

| | |
|---|----------|
| I/ Presentation du projet : | 2 |
| II/Objectifs du projet : | 4 |
| 1. Objectifs principaux : | 4 |
| 2. Objectifs secondaires : | 5 |
| 3. Objectifs tertiaires : | 6 |
| III/ Choix de la technologie | 7 |



I/ PRESENTATION DU PROJET :

Notre projet consiste à reproduire le jeu *Crypt of the Necrodancer*. Nous allons réaliser un *copycat* du jeu en intégrant les fonctionnalités principales du modèle utilisé et en se rapprochant au maximum de celui-ci.



Crypt of the Necrodancer est un jeu de rythme de type *Dungeon Crawler*. Un *Dungeon Crawler* est un jeu qui consiste à explorer des donjons. Dans ce jeu-ci, le personnage se déplace au rythme de la pulsation de la musique. Ces dites pulsations sont affichées à l'écran et si le personnage rate un battement, son tour est passé tandis que les ennemis, eux, peuvent agir. Le joueur doit constamment rester attentif car une erreur peut coûter très cher.

En plus d'être un *Dungeon Crawler*, *Crypt of the Necrodancer* répond aux codes du *Rogue Lite*. Cela signifie que chaque partie sera différente de la précédente. En effet, le joueur sera amené à récupérer et à acheter divers objets et équipement qui le rendront plus fort. C'est ce qu'on appelle couramment dans un jeu de ce type, un *build*. Votre *build* dépendra de votre façon de jouer, de votre situation à un moment t, de vos besoins et surtout, de votre chance. C'est cette notion qui rend chaque partie unique et qui fait tout le charme des *Rogue Lite*. Dans le même genre, on pourra citer les excellents *Binding of Isaac* ou *FTL : Faster Than Light*.

Cadence, le personnage principal, est une jeune fille dont le père a disparu depuis quelques années. Elle part donc à sa recherche en parcourant les donjons et en espérant le retrouver. Pour progresser dans sa quête, le personnage dispose de différents objets et pièces d'équipement tels que des armures, des armes, des consommables, des bombes, des sorts,... L'outil principal de Cadence est la pelle. Elle vous permettra de détruire des blocs de terre

(ou de roche) afin de vous frayer un chemin dans le donjon. Le joueur peut aussi acheter des effets permanents dans le *lobby* tel que des cœurs supplémentaires, de meilleurs objets obtenables dans les coffres, une augmentation de la fréquence d'apparition des coffres,... Le jeu comporte plusieurs niveaux, et chaque niveau est découpé en plusieurs étages. Chaque étage possède une musique différente et donc, un rythme différent. A chaque étage se trouveront des monstres, des pièges, un marchand, des coffres et quelques fois, des pièces secrètes qui donnent accès à de l'équipement plus puissant ou des objets rares. Il y a trois façon de passer à l'étage suivant : la plus commune est de prendre les escaliers gardé par un mini-boss (il est nécessaire d'en venir à bout pour débloquent l'accès aux escaliers). La seconde est d'attendre la fin de la musique, le joueur sera alors téléporté de force à l'étage suivant. Enfin, il est possible d'emprunter une trappe menant à un étage éloigné.

Chaque niveau se conclut par un affrontement contre le boss de la zone. Une fois que le joueur en vient à bout, il peut passer au niveau suivant.



II/OBJECTIFS DU PROJET :

L'objectif est de reproduire *Crypt of the Necrodancer*. Ou plus précisément, de recréer le moteur du jeu afin de permettre au client de l'exploiter pour un projet.

Nous avons recensé tous ces objectifs dans un *Backlog Product* que nous avons intégré dans le dossier.

Nous décomposerons les objectifs en 3 grandes familles.

1. OBJECTIFS PRINCIPAUX :

Les objectifs principaux consistent à concevoir et à développer la base, le cœur du moteur qui permettra d'ajouter de nouvelles fonctionnalités au fur et à mesure du développement.

Tout d'abord, le jeu devra pouvoir afficher un écran-titre. Depuis cet écran, il devra être possible de quitter le jeu ou de lancer une partie.

Ensuite il nous faut définir les fonctionnalités mères de notre jeu:

- ❖ Une carte doit pouvoir être générée aléatoirement. La carte comportera des cases classiques, des murs et un escalier. Une fois la carte créée elle sera affichée.
- ❖ Un joueur sera créé à chaque début de partie et placé sur la carte. Le joueur possèdera des points de vie et pourra se déplacer en haut, en bas, à gauche, et à droite. La gestion des collisions devra être intégrée afin qu'on ne puisse pas passer à travers les murs.
- ❖ La vie du joueur sera affichée à l'écran.
- ❖ Le programme lancera une musique en fonction de si on se trouve dans l'écran titre ou si on est en pleine partie.
- ❖ Un écran de victoire s'affichera si le joueur atteint la sortie.

2. OBJECTIFS SECONDAIRES :

Les objectifs que nous allons nous fixer dans cette partie vont nous permettre de mettre en place les bases du *gameplay* de *Crypt of the Necrodancer* :

- ❖ Le joueur pourra creuser à travers des blocs de terre à l'aide d'une pelle dont il pourra s'équiper.
- ❖ Le blob vert et le blob bleu seront intégrés au jeu, chacun de ces ennemis auront un pattern différent et infligeront des dégâts aux joueurs s'ils lui foncent dessus.
- ❖ Le joueur pourra attaquer en fonçant dans un ennemi s'il est équipé d'une arme.
- ❖ Le joueur pourra s'équiper d'une armure qui réduiront les dégâts subis et d'un consommable qui lui permettra de se soigner.
- ❖ Des coffres que le joueur pourra ouvrir seront générés aléatoirement sur la carte.
- ❖ Le joueur pourra récupérer de l'or dans les coffres ou après avoir tué un ennemi.
- ❖ La partie se terminera si le joueur perd tous ses points de vie et un écran de *Game Over* sera affiché.
- ❖ L'*HUD* affichera l'or possédé par le joueur ainsi que son équipement.
- ❖ Il sera possible de mettre le jeu en pause et d'accéder au menu des options. A partir de ce menu, l'utilisateur pourra modifier le volume de la musique et des bruitages.

3. OBJECTIFS TERCIAIRES :

Ces objectifs vont nous permettre de nous rapprocher au maximum du jeu original. Les fonctionnalités qui vont suivre ne sont pas nécessaires pour concevoir le moteur du jeu mais permettront d'étendre les possibilités offertes par celui-ci :

- ❖ Le joueur devra se déplacer AU RYTHME de la musique. S'il loupe un battement, le jeu passera son tour l'exposant ainsi aux assauts ennemis. Une barre de rythme permettra au joueur de suivre la cadence.
- ❖ Chaque action exécutée au bon moment initialisera un combo ou le préservera.
- ❖ Plus de pelles, d'armes, d'armures et de consommables.
- ❖ Le joueur pourra récupérer des diamants.
- ❖ Des pièges seront générés aléatoirement sur la carte.
- ❖ Le joueur pourra ramasser et poser des bombes
- ❖ 2 niveaux supplémentaires seront ajoutés. Le joueur changera de niveau en prenant les escaliers ou à la fin de la musique.
- ❖ Des ennemis seront rajoutés : le blob jaune, le squelette, la chauve-souris, le dragon, l'armadille, le clone et le zombie.
- ❖ A la fin des 3 niveaux, le joueur devra affronter un boss. Le vaincre déblocuera l'escalier qui mènera à l'écran de victoire.
- ❖ Des magasins et un marchand seront générés dans chaque niveau, le joueur pourra y dépenser ses pièces d'or contre des objets.
- ❖ Le joueur commencera la partie dans un *lobby* dans lequel il pourra : dépenser ses diamants dans des magasins et sélectionner le niveau auquel on veut jouer. Le joueur ne pourra accéder aux deux autres niveaux qu'une fois ceux-ci atteints.
- ❖ Il y aura une certaine probabilité pour qu'un marchand emprisonné apparaisse dans le donjon. Le joueur devra acheter la clé au magasin pour délivrer le marchand. Ce dernier établira son magasin dans le *lobby*.
- ❖ L'*HUD* affichera les diamants, la clé et les bombes possédés par le joueur.
- ❖ Il sera possible de passer le jeu en plein écran et de modifier les contrôles dans le menu des options.

III/ CHOIX DE LA TECHNOLOGIE

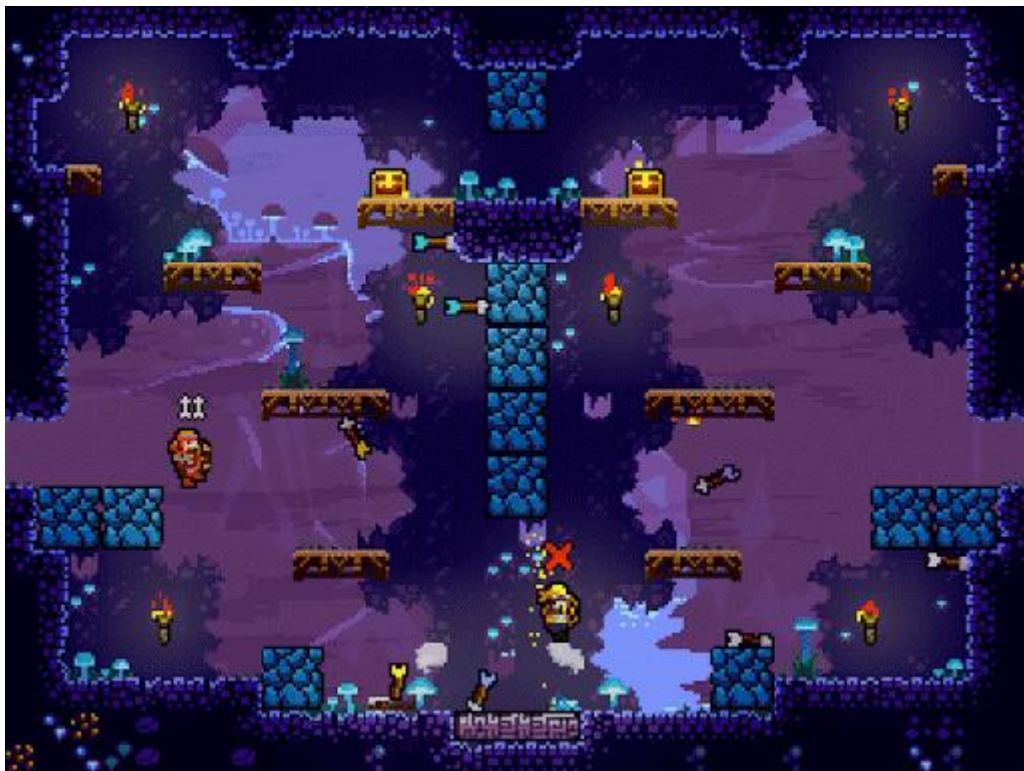
Le client nous a imposé la technologie à utiliser pour le développement du moteur du jeu : Java.

Cependant, en tant que techniciens, nous allons présenter d'autres technologies qui seraient susceptibles de servir de support au projet :

❖ Le C++ :

Les + : Ce langage permet d'utiliser des *Framework* reconnus tel que Monogame. Ce dernier a notamment servi à développer des jeux comme *Towerfall Ascension*, un jeu 2D indépendant.

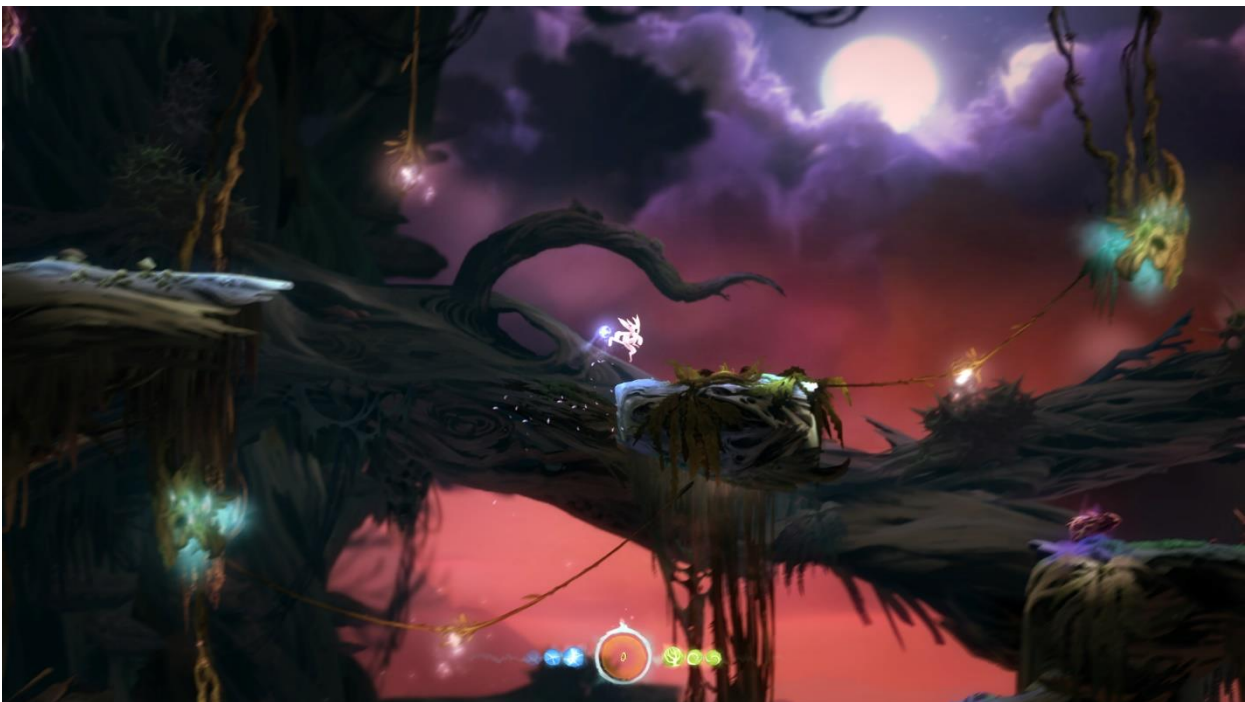
❖ Les - : Cependant, cette alternative bien que très adéquat au projet est moins pratique pour la réalisation car les *Framework* sont moins documentés qu'en Java. De plus Java propose déjà des *Framework* permettant la réalisation simple de ce type de projet (JavaFX).



❖ Le C# :

Les + : Nous aurions également pu choisir le C#. Langage très proche du Java, il permet de travailler avec le Framework Unity, très reconnu également et souvent comparé à Monogame. Il a notamment servi à développer le jeu indépendant *Ori and the Blind Forest*.

Les - : Si ce langage permet effectivement de travailler avec Unity, ce dernier ne semble pas très adapté à notre projet. En effet, nous n'avons pas trouvé beaucoup d'exemple de jeu avec un design *16 bits* ce qui indique qu'il serait préférable de se tourner vers autre chose. Cependant, il est tout de même possible de concevoir notre moteur en C#.



CRYPT OF THE NECRODANCER