Rapport intermédiaire

SAE

Tower of Babel

# Introduction

## Rappel du contexte

Dans le cadre de notre année en BUT3, il nous a été demandé de réaliser un jeu pendant l’ensemble du premier semestre. Ce projet de groupe doit respecter plusieurs contraintes : utiliser une API complexe (Framework, …) interroger une base de données et s’appuyer sur le serveur. Le projet que notre équipe a décidé de réaliser s’appelle Tower Of Babel

## Rappel du pitch du jeu

Tower of Babel est un jeu d'exploration de donjon et de combat au tour par tour. Le joueur prépare et compose son équipe avant d'explorer les étages d'une tour infinie remplie de monstres et de trésors. Le but du jeu est d’aller le plus haut dans la tour possible et d’amasser le plus de butin. Cette tour est générée aléatoirement à chaque exploration, chaque partie est donc unique. L’équipe du joueur est composée de quatre héros, chacun avec des forces, faiblesses et ses compétences uniques.

Ce jeu a pour but d’être abordable mais intéressant en utilisant une mécanique « easy to learn, hard to master » : le jeu est facile à comprendre et à prendre en main même pour des inhabitué, mais va demander une réflexion plus poussée pour atteindre des étages plus haut dans la tour.

La conception de Tower of Babel s'inspire principalement du jeu Darkest Dungeon, dont elle reprend la mécanique de placement des personnages au sein de l'équipe : positionner un héros à l'avant ou à l'arrière de l'équipe a une importance pour les compétences du héros. Le jeu simplifie et facilite de nombreux aspects du jeu de base en éliminant la mécanique du stress ou de la santé mentale, en rendant la fin de partie non punitive et en réduisant le facteur aléatoire. La principale innovation de ce projet par rapport à Darkest Dungeon réside dans l'exploration infinie d'une tour générée de manière aléatoire, contrairement à Darkest Dungeon où l'exploration se limite à un donjon suivi d'une sortie.

# Changement apporté au jeu

## Modification de ce qui a été planifié

Certaines fonctionnalités prévues ont été retravaillées pour mieux correspondre aux attentes des joueurs :

• Le système de forge est plus consistant que prévu : initialement, il s'agissait de pouvoir améliorer quatre éléments (arme ou armures) de héros au hasard. Désormais, le joueur peut choisir ce qu'ils améliorent sur chaque héros. L'élément de loterie, qui était recherché à la base, n'est pas ludique et empêche le joueur d'utiliser ses héros favoris, le poussant davantage à utiliser les héros que le jeu propose d'améliorer.

Une image contenant texte, Jeu PC, capture d’écran, Compositing numérique

Description générée automatiquement

*Ci-dessus, la nouvelle interface de la forge. On peut choisir jusqu’à quel niveau on souhaite améliorer l’arme et l’armure d’un héro séléctionné.*

• Le système de diligence est plus souple : selon ce qui était initialement planifié, seul un petit nombre de héros serait disponible à l'achat de manière aléatoire entre chaque partie, rendant la possibilité d'acheter un héros spécifique peu fiable. Désormais, tous les héros sont disponibles à l'achat en tout temps. Les principales raisons de ce changement sont le fait qu'il n'y aura pas suffisamment de héros dans le jeu final pour que ce système fonctionne d'un point de vue ludique. De plus, ce jeu vise à être très abordable. Accorder davantage de liberté gratuitement au joueur va dans ce sens.

## Suppression de ce qui a été planifié

D’autres fonctionnalités ont été supprimées de ce qui a été planifié, la plupart d’entre elle pour des raisons de temps et parce qu’elles étaient trop ambitieuses :

• Le système de potion utilisable en combat ne sera pas implémenté dans la première version du jeu car trop complexe à implémenter dans le temps impartie.

• Le système de provision utilisable en exploration ne sera pas implémenté dans la première version du jeu car trop de retard sur la partie « exploration » en raison de la complexité de la tâche de base.

• Le nombre de classe jouable va être revu à la baisse.

# Avancement

## Ressentie global

De manière générale, le projet avance moins vite que prévu en raison de difficulté à prendre les nouveaux outils en main.

## Détail sur le travail de Lothaire

La majeur de partie de mon travail s’est concentrée sur l’implémentation du combat dans le jeu. J’ai commencé par créer les mécanismes qui permettent d’animer les personnages correctement, afin de ne pas avoir à gérer ça en même temps que la partie logique. Ce travail a pris environ 2 semaine.

Ensuite, je me suis lancé dans la partie sur la partie logique du combat et j’ai rencontré plusieurs problèmes dès la conception de l’architecture, je ne savais pas comment aborder le problème avec Phaser. Après de nombreux test, j’ai décidé d’utiliser le design patern « state machine ». Ce choix semble être évident car il convient parfaitement à la création d’un jeu de combat en tour par tour (aucune physique à gérer, il faut seulement gérer chaque état du combat), mais son implémentation dans Phaser a nécessité un effort de réflexion.

Phaser fonctionne avec un système de scène, chaque scène est un écran : un menu de pause, un écran de sélection, un niveau, etc… Dans mon cas, la scène n’a besoin que d’un combat pour marcher. On peut placer des images, ou sprites, sur les scène. Les scènes ont par défaut plusieurs fonction, mais notamment « create() » qui permet d’initialiser la scène et « update() » qui permet de mettre à jour la scène. Ma scène de combat n’a pas de fonction update() pour la mettre à jour. A la place, une classe « arbiter » va la manipuler. De manière visuelle, on peut voir la scène de combat comme un terrain de foot, ses sprites comme des joueur et ma classe arbiter comme l’arbitre du jeu qui indique qui doit faire quoi en fonction des entrée du joueur et de l’intelligence artificielle des ennemis.

Ce système n’est pas un usage traditionnel de Phaser, mais est parfaitement valide d’après un consensus général de la communauté Phaser.

Actuellement, 2 classe de héros et 2 classe d’ennemis sont implémenté afin de finir le prototypage du jeu. Les personnages attaque les un à la suite des ordres en fonction d’une statistique de vitesse, ont chacun plusieurs attaques, les héros ont plusieurs compétence, qu’ils peuvent utiliser ou non en fonction de leur position dans l’équipe et de l’existence d’une cible valide, leur points de vie restant est représentée par une barre, les personnages se replacent correctement après la mort d’un alliée, etc…

Les seules mécaniques qui ne sont pas encore implémentée sont les attaques visant plusieurs cibles et les états de statuts (saignement, empoisonné, assommé).

Le nombre de héros et d’ennemis actuel n’est pas impressionnant, mais il est très facile d’en rajouter grâce à mon architecture. Pour les héros, la partie la plus longue de création est la création des ressources visuelles, et pour les ennemis, la création de leur intelligence artificielle car elle est unique à chacun.

## Détail sur le travail de Nuoxi

Voici l'état d'avancement détaillé de mes travaux jusqu'à présent. J'ai accompli plusieurs tâches, notamment l'ajout de la barre de jeu, le développement de la page du hameau, et la finalisation de la composition de l'équipe avant le début de l'exploration. Veuillez trouver ci-dessous les détails de chaque étape de mon travail :

* Ajout de la barre de jeu comprenant le nom d'utilisateur, le nom du jeu et la quantité de pièces d'or. Mise en place d'une fonctionnalité de modification en temps réel des données des pièces d’or.
* Ébauche de la classe d'informations utilisateur, incluant le compte, les données de jeu, etc. et création des fonctions correspondantes de mise à jour des données : ajout de nouveaux héros, amélioration d'équipements, achat de potions, etc.
* Développement de la page du village avec les bâtiments associés : l'alchimiste (achat de potions, affichage des stocks et des informations détaillées), le chariot (présentation détaillée des compétences des héros, achat de héros) et le forgeron (amélioration des équipements spécifiques des héros dans l'équipe). Ajout d'événements réactifs pour chaque page, stockant les opérations de l'utilisateur dans localStorage.
* Finalisation de la page de configuration de l'équipe avant le début de l'exploration : chaque exploration nécessite la configuration de quatre coéquipiers. Transmettre les informations sur les héros inclus dans l'équipe via localStorage.

## Détail sur le travail de Haroun

Plusieurs tâches ont été réalisées jusqu’à présent. L'ensemble des fonctionnalités primaires liées à une salle et à un couloir ont été complétées. Cela inclut l’ouverture d’un coffre, le lancement d’un combat, le passage d’une salle vers un couloir et inversement, ainsi que le déplacement des personnages au sein du couloir. La base de données a aussi été mise en place.

Une équipe peut se déplacer dans un couloir et utiliser une porte pour passer à la salle suivante en test. Pour l'instant, une équipe peut se déplacer d'un couloir à une salle, et d'une salle à un couloir. Chaque salle a un type qui permet de définir son contenu, notamment s'il y a un combat ou non. Pour l'instant, la gestion de l'inventaire durant l'exploration n'est pas encore gérée. L'exploration se fait linéairement pour l'instant, mais cela évoluera pour permettre de choisir quelle salle visiter.

Il reste encore à relier les salles contenant un combat au combat réel. Il existe actuellement un étage 0, qui servira de tutoriel. Il reste encore à développer la création d'étages de façon aléatoire et le passage d’un étage à une autre. Les couloirs peuvent contenir des curios ou des combats. Cette fonctionnalité a en cours d’implémentation.

Concernant la base de données, on peut y accéder à distance. Cependant, l'ensemble des requêtes la concernant n'a pas été effectué ; seules celles nécessaires ont été faites pour l'instant. Il reste également à ajouter une transition entre deux étages afin de passer d'un étage à l'autre de manière visible.

# Mise à jour du planning de Gantt