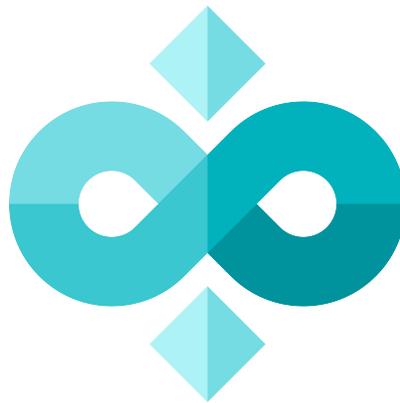


Dungeon BattleRush

par InfiniTeam

Rapport de Soutenance 2



Maxime Buisson
Oriane Margelisch
Léa Margery
Florent Sagot

Table des matières

1	Introduction	3
2	Présentation générale	4
2.1	Avancement général	4
2.2	Avancemant site Web	4
2.3	Problèmes rencontrés	7
3	Avancement	8
3.1	Graphismes	9
3.1.1	Map (Carte de jeu)	9
3.1.2	Personnages et Objets	12
3.1.3	Animations	13
3.1.4	Design	14
3.2	Interfaces	15
3.2.1	Menus	15
3.2.2	HUD	16
3.3	Audio	17
3.3.1	Musiques	17
3.3.2	Bruitages	17
3.4	IA	18
3.5	Réseau	20
3.6	Mécaniques de jeu	21
3.6.1	Déplacements	21
3.6.2	Interactions et Combats	21
3.6.3	Randomisation	23
3.6.4	Effets des statistiques	23
3.7	Git repository	24
4	Avances et retards	25
4.1	Rappels sur ce qui a été fait	26
4.2	Ce qui reste à faire	26
5	Conclusion	27

1 Introduction

Ce rapport de soutenance a pour but de dépeindre l'avancement du projet *Dungeon BattleRush* depuis la première soutenance.

Le projet *Dungeon BattleRush* est né le 5 décembre 2020, et a pour objectif le développement d'un jeu vidéo éponyme en C#, sous le moteur de jeu Unity. La réalisation du jeu vidéo que nous avons imaginé, inspiré de nos expériences vidéoludiques personnelles mises au goût du jour, représente un vrai défi que nous voulons surmonter pour notre premier projet à l'EPITA. C'est ainsi que nous avons imaginé *Dungeon BattleRush*, un jeu 3D sur ordinateur permettant les modes solo et multijoueur, mêlant fantastique, combat et aventure dans un décor médiéval.

Pour rappel, *Dungeon BattleRush* est la combinaison d'un "Dungeon crawler", d'un "Battle Royal" et d'un jeu d'arène, en 3D, dont le but est d'être le dernier survivant du donjon. À travers les différentes salles, vous affronterez des monstres, surmonterez des obstacles et résoudrez des énigmes afin de récupérer des items à mettre dans votre inventaire. Ils vous permettront d'affronter, dans une arène, le boss final. Dans ce jeu dynamique, seul ou à plusieurs, vos principaux ennemis ne sont pas uniquement les monstres, mais également le temps !

En **mode solo**, la partie se déroule dans un donjon, à travers plusieurs salles remplies d'ennemis et d'énigmes. L'objectif du joueur est de récupérer des armes, des boucliers et des potions, afin d'augmenter ses statistiques. À la fin du temps imparti, il se retrouve ensuite dans une arène face à un boss final aléatoire qu'il doit battre pour gagner la partie.

En **mode multijoueur**, seul l'affrontement final diffère. Deux alternatives sont possibles. Premièrement, chaque joueur joue en concurrence avec les autres. Sinon, ils peuvent être en équipe de deux, la totalité de la partie se jouant alors en duo. Toute place laissée vacante dans les modes de jeu décrits précédemment sera occupée par des intelligences artificielles (alliées ou ennemies). Au bout d'un certain temps, les équipes sont lâchées dans une arène et la plus forte est déclarée gagnante.

2 Présentation générale

2.1 Avancement général

L'organisation que nous avons développée pour la première soutenance nous a permis et nous permet encore à l'heure actuelle d'être coordonnés lors de nos séances de travail. L'avancement du jeu reste cohérent avec l'idée initiale que nous avons et les contraintes dressées dans le cahier des charges. En effet, la conception de *Dungeon BattleRush* apporte à l'ensemble du groupe de nouvelles compétences et connaissances autant sur le plan technique que relationnel. Ainsi, ce projet nous satisfait pleinement et, bien que ce soit le premier, nous avons su nous adapter et surmonter les difficultés rencontrées.

2.2 Avancement site Web

Le responsable du web est Florent Sagot, et la suppléante est Léa Margery. Le site web, comme énoncé dans le cahier des charges, se fait sans outils interactifs de création de site web. Il a donc été créé avec Sublime Text 3, et hébergé localement grâce à Xampp. Le site possède donc un menu de navigation, menant sur les pages d'accueil, d'actualité, de téléchargement, l'historique, les documents, nous contacter, et enfin à propos. L'accessibilité à la langue anglaise se fera à la fin, et n'est donc pas encore faite.

Les pages qui ont été complétées sont les rubriques : Accueil, Télécharger, Documents, et InfiniTeam. Manque donc encore les rubriques Actualités, Historique, Nous contacter, et A propos. Nous avons finalement opté d'inscrire notre site dans aspect classique, mais tout de même captivant grâce à son style épuré, avec une vidéo d'arrière-plan.

Le site web est pour le moment hébergé sur le service Byethost, un hébergeur gratuit. Mais ce service manque de fiabilité et de performance. C'est pourquoi nous comptons à terme migrer sur le service d'hébergement de GitHub. Mais ce changement de service nécessitera une réécriture du site du langage PHP, vers de l'HTML.

Les copies de page écrans des rubriques faites sont sur les pages ci-dessous :

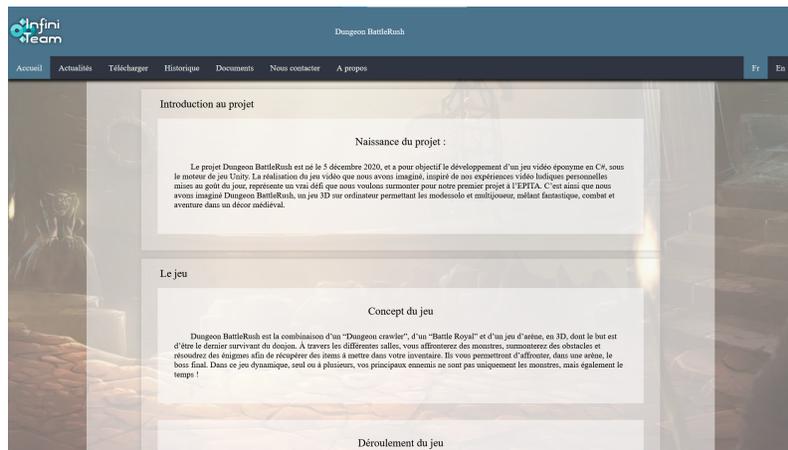


FIGURE 1 – Site Web - Accueil

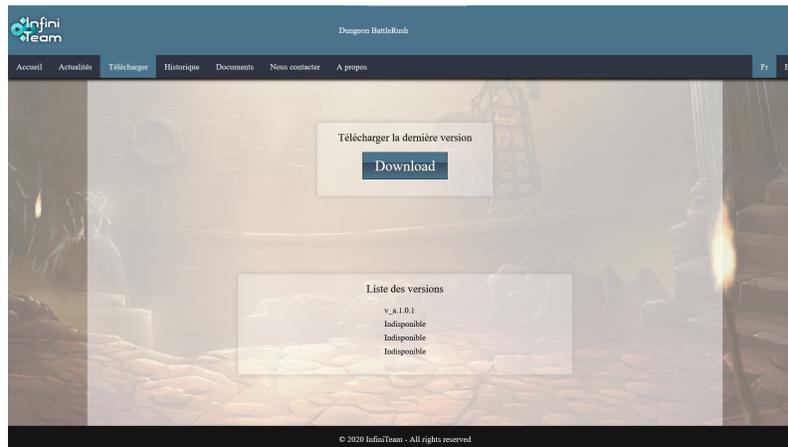


FIGURE 2 – Site Web - Télécharger

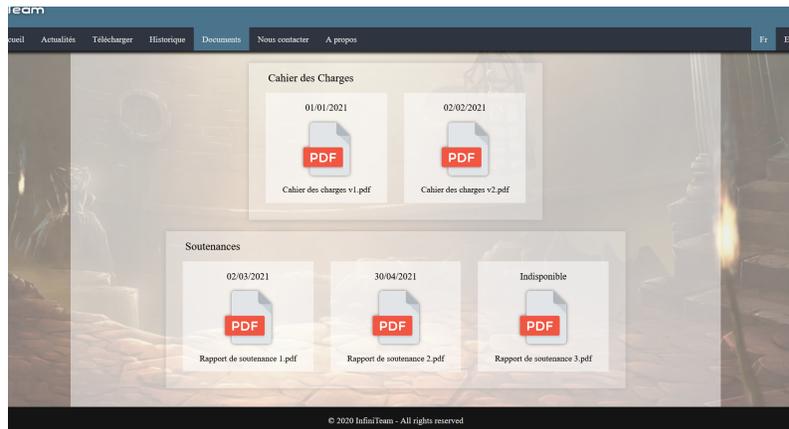


FIGURE 3 – Site Web - Documents

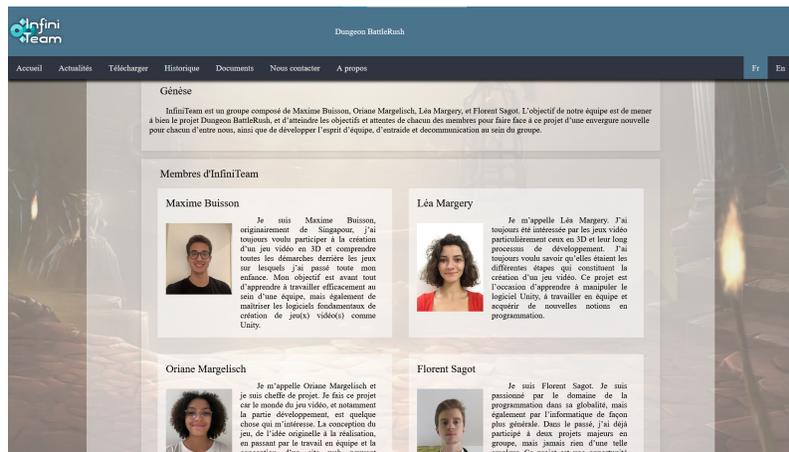


FIGURE 4 – Site Web - InfiniTeam

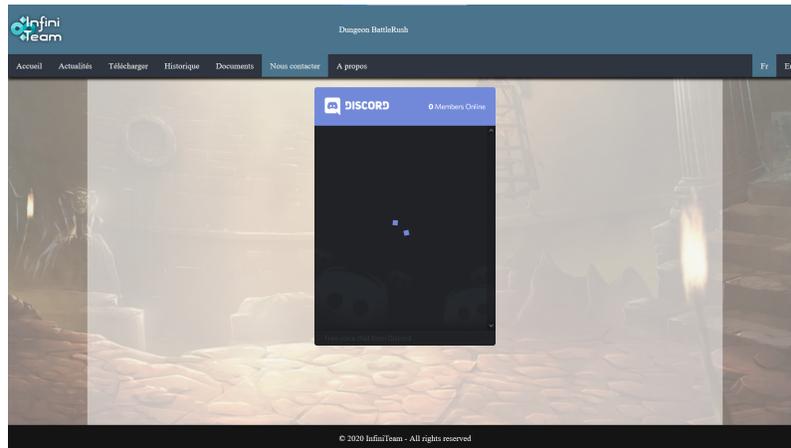


FIGURE 5 – Site Web - Documents

2.3 Problèmes rencontrés

Initialement, nous avons eu quelques problèmes de communication dans le groupe qui ont vite été réglés par la mise en place de réunions hebdomadaires, permettant à chaque membre de rester informé de l'avancement du jeu et du travail de ses collègues. Depuis, la communication au sein de l'équipe constitue une véritable force, nous permettant d'évoluer le plus sereinement possible.

En outre, nous rencontrons continuellement des problèmes plus ou moins importants dans la conception du jeu, ce qui est inévitable dans le développement d'un projet d'une telle ampleur. Cependant, que ce soit seul ou en groupe nous arrivons à les surmonter et parfois même à dépasser nos attentes.

3 Avancement

Pour rappel, voici la répartition des rôles telle qu'elle est prévue dans le cahier des charges, que nous avons su respecter dans le développement de *Dungeon BattleRush* :

	Maxime	Léa	Oriane	Florent
Graphismes				
Map (Carte de jeu)	S	R		
Personnages et Objets	S		R	
Animations	R		S	
Design (logo, icônes, ...)		S		R
Interfaces				
Menus (accueil, options, ...)	S		R	
HUD	S		R	
Audio				
Bruitages	X	R	X	X
Musiques	R	S		
IA				
Mobs (ennemis, boss, ...)	S		R	
Faux Joueurs	R		S	
Réseau				
Multijoueur		R		S
Mécaniques de jeu				
Déplacements	S		R	
Interactions et Combats		S		R
Effets des statistiques		S		R
Randomisation			S	R
Modes de jeu		R		S
Git Repository				
Gestion du Git			R	S
Web				
Site Web		S		R
Documentation				
Mise en page LaTeX				R
Contenu	X	X	X	X

3.1 Graphismes

3.1.1 Map (Carte de jeu)

La responsable du design de la carte de jeu est Léa Margery et le suppléant est Maxime Buisson.

Depuis la dernière soutenance, nous avons décidé d'augmenter le nombre de portes par salles, afin de faciliter la circulation du joueur dans le donjon généré aléatoirement. Ainsi toutes les salles sauf la salle d'apparition et celles situées en bordure disposent de quatre entrées, une par mur. De plus, nous avons créé plusieurs nouvelles salles, en utilisant les assets importés auparavant, et les avons classées selon quatre catégories :

- Les salles de combat : ce sont les salles dans lesquelles se trouvent les IA (un type d'IA par salle) et les coffres qui permettent d'augmenter la puissance via les armes ou l'armure avec les boucliers.
- Les salles d'énigme : nous disposons à l'heure actuelle d'une salle d'énigme dans laquelle le joueur doit trouver une clé cachée dans la salle pour ouvrir la porte localisée au Nord qui est verrouillée. Les salles d'énigmes contiennent quant à elles des potions augmentant les statistiques d'endurance, de santé et de vitesse.
- Les salles situées aux extrémités : ces salles constitueront les bords du donjon généré aléatoirement. Il y aura donc des salles qui formeront les quatre coins du donjon et d'autres pour les quatre bords.
- La salle d'apparition : celle-ci est plus petite que les salles d'énigme et de combat puisqu'elle sert uniquement au début de partie. C'est dans cette salle que le joueur apparaît.

Afin de pouvoir générer le donjon aléatoirement, nous avons réalisé des "*prefabs*" (salles préfabriquées). Pour que celui-ci soit le plus parfait possible, ces salles préfabriquées doivent avoir exactement la même taille, être centrées et l'ensemble des murs doit être parfaitement symétrique, ce qui n'était pas le cas dans les salles préfabriquées réalisées pour la première soutenance. De même, nous avons dû modifier les différentes arches qui permettent de passer d'une salle à une autre, puisqu'un léger décalage de ces entrées pourrait empêcher le passage dans une autre salle.

Enfin, nous avons aussi créé la salle dans laquelle se déroulera le combat final contre le "*boss*". La salle finale, que nous appellerons arène, est plus grande que les salles qui constituent le donjon et a la forme d'un amphithéâtre romain.



FIGURE 6 – Salle - 1.1



FIGURE 7 – Salle - 1.2

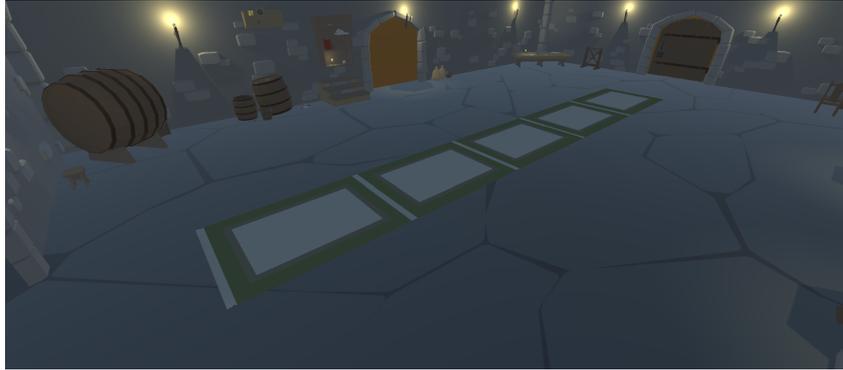


FIGURE 8 – Salle - 2

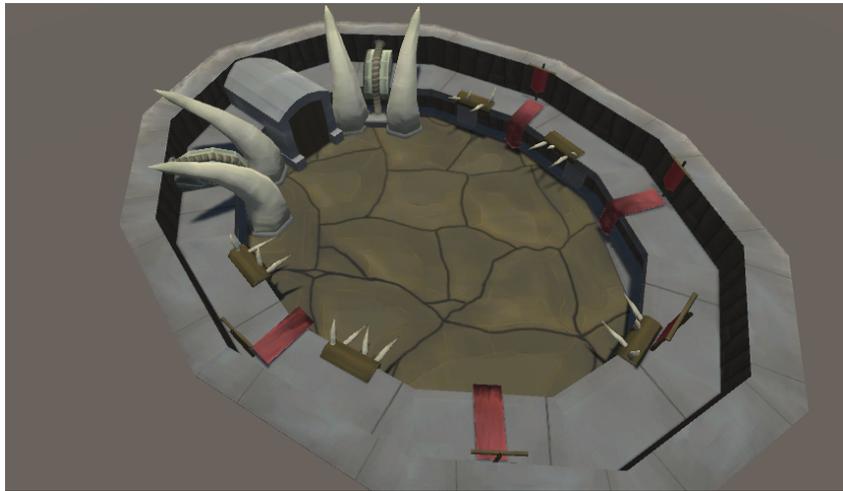


FIGURE 9 – Salle - Arène finale

3.1.2 Personnages et Objets

La responsable de la partie graphique des personnages et objets est Oriane Margelisch et le suppléant est Maxime Buisson.

Les héros du jeu proviennent d'un asset que nous avons acheté 5 euros sur le Unity Assets Store. Nous en avons gardé quatre sur cinq présents dans le pack, afin de respecter le nombre de personnages disponibles dans le jeu que nous avons prévu initialement. Ils sont en texture lowpoly, en accord avec le thème de notre jeu. Concernant les modèles des ennemis (intelligences artificielles) et les boss de fin, ils proviennent également pour certains du Unity Assets Store, et pour d'autres de sites mettant à disposition en téléchargement libre des assets.

Les objets que l'on peut voir sont également issus du Unity Assets Store, et nous les avons obtenus gratuitement.

Nous avons depuis la soutenance précédente changé nos coffres afin qu'ils conviennent mieux au jeu, et avons également importé un pack d'armes complet, incluant toutes celles que nous utilisons.



FIGURE 10 – Coffre



FIGURE 11 – Porte

3.1.3 Animations

Le responsable des animations est Maxime Buisson et la suppléante est Oriane Margelisch.

Les animations choisies pour les personnages sont restées les mêmes depuis le début du jeu, mais nous avons, depuis la première soutenance, réussi à régler un problème que nous avions avec le saut.

Concernant les intelligences artificielles, seule l'animation de course était implémentée lors de la première soutenance. Nous avons ainsi rajouté une animation d'attaque. Les deux IA ont leur propre "*animator*", ce qui permet à chacune d'avoir des animations qui leur sont propres (elles attaquent et ne se déplacent pas de la même manière, diversifiant ainsi les interactions avec les joueurs).

De plus, les salles d'énigmes ouvrent une porte, pour laquelle nous avons nous-mêmes créé l'animation d'ouverture de celle-ci dans Unity.

Enfin, l'ajout de coffres dans le jeu implique l'implémentation d'une animation associée à leur ouverture, que nous avons récupérée dans le pack de coffres de l'assets.

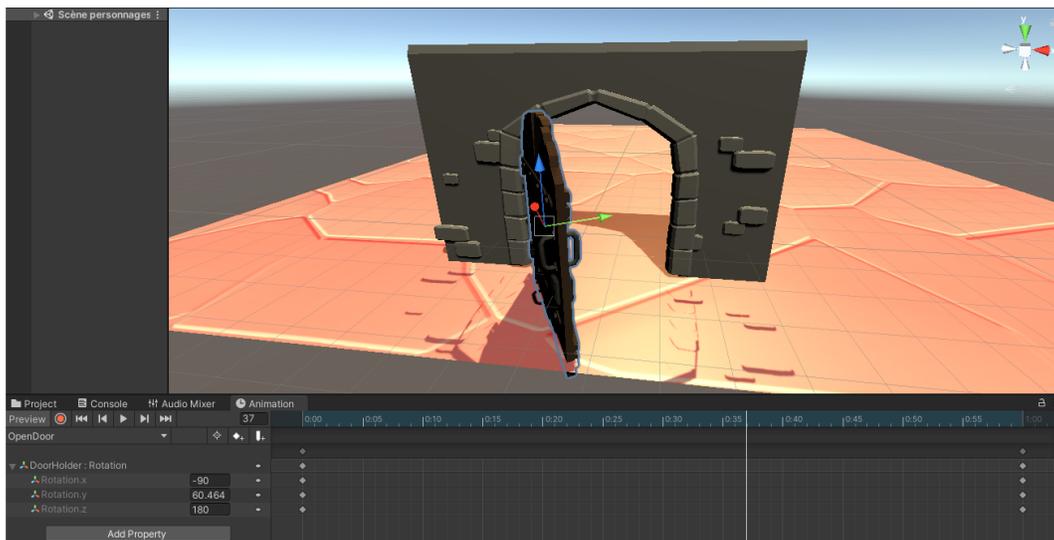


FIGURE 12 – Animation porte

3.1.4 Design

Le responsable des designs et logos est Florent Sagot et la suppléante est Léa Margery.

À la dernière soutenance, nous disposions uniquement du logo de notre équipe puisque notre vision du thème du jeu n'était pas encore entièrement complète, ce qui n'est plus le cas maintenant. C'est pourquoi nous avons pu nous pencher sur la création d'un logo pour le jeu. Pour cela nous avons dessiné le logo en nous inspirant d'images libres de droits. Nous avons fait attention à ce que le logo reste fidèle au thème du jeu médiéval et fantastique.

De plus, nous avons récupéré des icônes libres de droits pour le menu déroulant et les statistiques de jeu du joueur. En ce qui concerne les statistiques du joueur, nous avons modifié les coloris des icônes. Ainsi, la statistique de vitesse associée à une potion verte est représentée dans le HUD par une icône verte et nous avons fait de même pour toutes les statistiques. Ces changements de coloris permettront au joueur de comprendre rapidement laquelle des statistiques est modifiée par une potion d'une couleur spécifique.

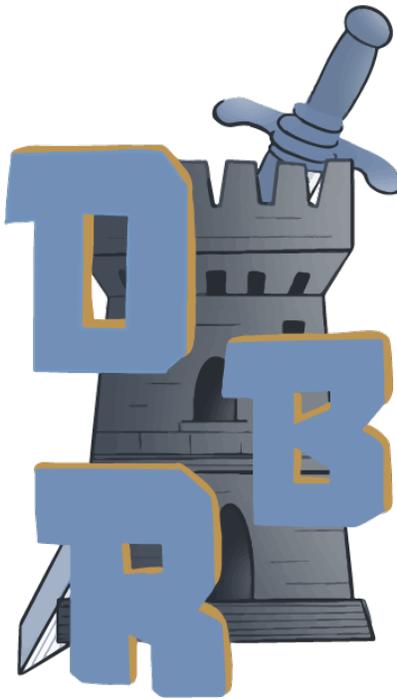


FIGURE 13 – Logo du jeu

3.2 Interfaces

La responsable des interfaces comprenant la conception, le fonctionnement et les graphismes des menus et HUD est Oriane Margelisch et le suppléant est Maxime Buisson.

À l'heure actuelle, les menus et le HUD en jeu sont fonctionnels, incluent toutes les options modifiables prévues initialement, augmentent les interactions entre le jeu et le joueur, et permettent une meilleure compréhension de la partie par ce dernier.

3.2.1 Menus

Lors du lancement de l'application, le jeu débute par un menu fonctionnel, et trois options s'offrent au joueur :

Page du menu principal :

- Play : Mène à une autre page du menu permettant de choisir son mode de jeu
- Options : Mène à la page des options du jeu
- Quit : Quitte l'application

Page Option :

- Mettre la fenêtre en plein écran ou non
- Choisir la résolution du jeu (les dimensions proposées sont calculées automatiquement en fonction de l'écran du joueur)
- Choisir la qualité de jeu (six options disponibles, de *"very low"* à *"ultra"*)
- Contrôle du volume du jeu

Page de choix du mode de jeu :

- Solo : lance une partie en mode solo
- Multijoueur : créer ou rejoindre une salle pour une partie en mode multijoueur

Page de choix du personnage :

- Quatre héros en lowpoly sont disponibles



FIGURE 14 – Menus

3.2.2 HUD

En jeu, de nombreuses informations doivent être communiquées au joueur. Ici, grâce au HUD, le joueur peut, en temps réel, suivre les statistiques de son personnage : sa vie, son armure, sa puissance, sa vitesse et son endurance. Chaque barre représentative est colorée en fonction de la potion associée, et une icône parlante permet une meilleure analogie.

Il est également possible de voir le temps restant avant la téléportation dans l'arène finale depuis le donjon, ou bien la fin du jeu face au boss final. Le compte à rebours se met également en rouge pendant les dix dernières secondes de jeu.

De plus, il a accès à un menu déroulant, qui lors de son actionnement, affiche un bouton d'accès aux paramètres dans la partie (les mêmes que dans le menu options) et un autre quittant la partie et menant directement au menu principal.

En mode multijoueur, le HUD étant directement relié au joueur, cela permet à chacun d'avoir ses propres caractéristiques à l'écran.



FIGURE 15 – HUD

3.3 Audio

3.3.1 Musiques

Le thème retenu pour nos différentes musiques est donjon ou héroïque fantasy, avec un florilège de musiques calmes et épiques pour les niveaux à énigmes et de combats, ainsi que pour les menus.

Nous avons utilisé les audios libres de droits pour créer trois musiques distinctes :

- Menu principal.
- Jeu : cette musique est composée d'une première plutôt calme, puis d'une seconde stressante et enfin à nouveau une musique calme.
- Arène finale : il s'agit d'une musique épique qui permet de mettre le joueur dans l'ambiance lors du combat final contre le "boss".

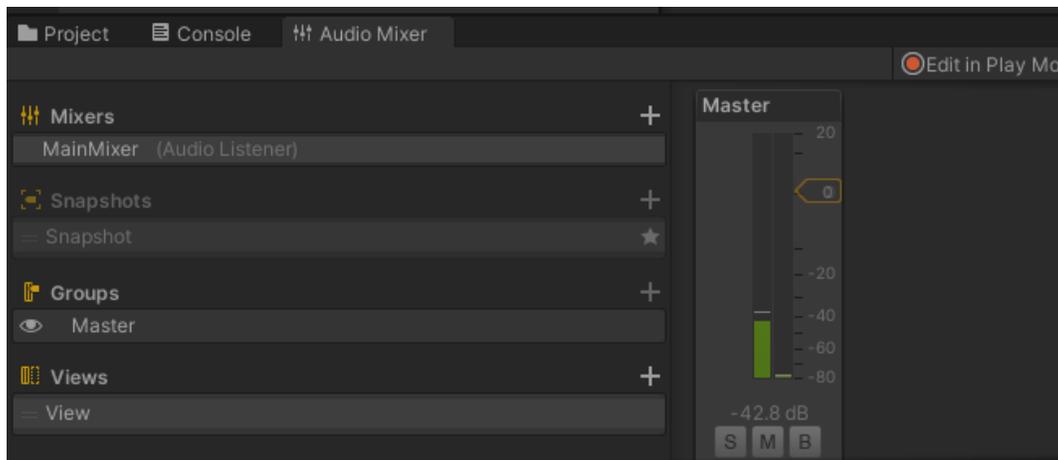


FIGURE 16 – Audio Mixer

3.3.2 Bruitages

Les bruitages seront faits réalisés par le groupe, afin d'apporter encore plus de création personnelle au projet. Mais nous avons prévu de les faire ensemble, nous avons dû repousser cela, à cause des contraintes du confinement.

3.4 IA

La responsable de l'intelligence artificielle est Oriane Margelisch, et le suppléant est Maxime Buisson.

L'IA est actuellement capable de :

- Patrouiller dans le donjon lorsqu'il n'y a pas de joueur dans son périmètre de détection
- Poursuivre le joueur le plus proche qui pénètre dans son périmètre de détection, jusqu'à ce que celui-ci en sorte
- Suivre du regard ce même joueur, avec une possibilité de rotation pour s'adapter en permanence à sa position, toujours dans ledit périmètre de détection
- Et enfin d'attaquer lorsque le joueur est dans sa zone d'attaque

Pour cette deuxième soutenance, nous avons amélioré l'implémentation de l'IA. Cette dernière vérifie à chaque *"frame"* la distance entre le joueur et elle-même. Pour cela, le joueur est automatiquement détecté puis placé en paramètre depuis son script, en tant que "cible". Les déplacements de l'IA se font à travers une fonction qui calcule le trajet le plus court pour atteindre sa cible.

De plus, les déplacements dans la salle sont permis grâce aux *"Component NavMesh"* qui spécifient les zones dans lesquelles l'IA (*"NavMesh agent"*) peut se déplacer (zones bleues ci-dessous). Ainsi, celle-ci aura une nouvelle position en fonction de sa cible et du lieu de navigation établis. Cependant, nos salles, qui ne sont générées qu'au démarrage du jeu (elles ne sont pas instanciées avant), n'ont donc pas de *"NavMesh"* puisqu'il est impossible d'en ajouter un prédéfini à un objet *"prefab"*.



FIGURE 17 – Navmesh

Ainsi, pour que l'intelligence artificielle puisse se déplacer dans notre jeu, nous avons rajouté une surface (objet de type "plane") suffisamment large sur laquelle la disposition des salles du donjon est générée aléatoirement.

Lorsqu'aucun joueur n'est dans son périmètre de détection, l'IA entame une procédure de patrouille sur l'entièreté de la carte du donjon. Pour cela, un point est calculé aléatoirement, puis elle s'y rend, et cette procédure se répète tant qu'aucun joueur n'est croisé. De plus, pour pallier un problème lié au "NavMesh" provoquant des déplacements inattendus (par exemple, l'IA se retrouve parfois bloquée face à un obstacle), nous avons dû modifier cette fonction afin que les coordonnées du point où l'IA se rend, qui sont calculées aléatoirement, soient actualisées si le problème est rencontré.

Si un joueur se trouve dans le périmètre de détection, l'IA se met alors à le suivre. Enfin, si celui-ci se trouve dans son périmètre d'attaque, l'IA s'arrête et l'attaque.

A l'heure actuelle, deux IA fonctionnelles ennemies sont implémentées dans le jeu. Celles-ci sont reliées au même script, que nous avons rendu le plus général possible, mais leurs caractéristiques de jeu (vie, attaque, défense, armure, etc.) sont assignées au type d'IA, et diffèrent donc. Le joueur peut ainsi se retrouver face à deux adversaires différents.

3.5 Réseau

La responsable du réseau est Léa Margery, et le suppléant est Florent Sagot.

Le jeu multijoueur est hébergé sur le Photon Cloud et peut accueillir jusqu'à vingt joueurs sachant que nous nous limitons à quatre joueurs maximum. L'utilisation de PUN nous a permis d'avoir une mise en relation des différents joueurs simple, puisqu'il n'y a pas besoin de rentrer un code, mais simplement de cliquer sur le bouton permettant de lancer une partie.

Dans le hall, nous pouvons :

- Rejoindre une salle aléatoirement
- Créer une salle aléatoirement (nom aléatoire et options de la salle)

Dans un hall, si le joueur est connecté aux serveurs Photon, il peut rejoindre une salle multijoueur aléatoire. S'il ne parvient pas à rejoindre une salle multijoueur, c'est parce qu'aucune n'a encore été créée. Alors une nouvelle salle aléatoire est créée. Si la création de la salle échoue, la tentative est réitérée. Lors du chargement d'une salle multijoueur, on peut à tout moment annuler le chargement de celle-ci et ainsi retourner au menu qui permet de choisir entre le mode solo et le mode multijoueur.

La salle multijoueur :

Si le joueur rejoint une salle multijoueur et qu'il s'agit du créateur de cette salle, alors le joueur se retrouve dans la scène multijoueur. Si le créateur de la salle quitte la partie en cours, alors ce rôle sera redistribué à un joueur encore présent dans la salle. Tout autre joueur qui essaye de rejoindre cette salle, après l'arrivée du master client dans la scène multijoueur, d'arriver dans cette même scène.

Initiation du joueur dans la partie :

Pour que les joueurs puissent se voir l'un l'autre, nous utilisons les composants "*Photon View*" et "*Photon Transform View*" sur les modèles des joueurs préfabriqués (prefabs). Pour créer un joueur, celui-ci est instancié directement à partir du dossier ressource Photon contenant ces modèles. A l'heure actuelle, nous pouvons créer jusqu'à 4 personnages.

Cependant, en implémentant la possibilité de choisir son joueur tout comme dans le mode solo du jeu, nous avons rencontré des difficultés pour les déplacements des joueurs. En effet, il semble que le mouvement d'un joueur entraîne celui des autres joueurs.



FIGURE 18 – Multiplayer

3.6 Mécaniques de jeu

3.6.1 Déplacements

Le personnage se dirige en rotation suivant le curseur de la souris : il le suit en temps réel, ce qui permet un déplacement fluide et offrant de nombreuses possibilités. Il peut de plus, à l'aide du clavier, se déplacer vers l'avant, l'arrière, les côtés et en diagonale, tout cela en marchant ou bien en courant, et relatif à l'orientation de la souris. Depuis la première soutenance, nous avons amélioré les déplacements, notamment ceux en diagonale du joueur.

Le personnage se dirige en rotation suivant le curseur de la souris : il le suit en temps réel, ce qui permet un déplacement fluide et offrant de nombreuses possibilités. Il peut de plus, à l'aide du clavier, se déplacer vers l'avant, l'arrière, les côtés et en diagonale, tout cela en marchant ou bien en courant, et relatif à l'orientation de la souris. Depuis la première soutenance, nous avons amélioré les déplacements, notamment ceux en diagonale du joueur.

3.6.2 Interactions et Combats

Interactions

L'implémentation du combat fut un grand défi pour nous, car cela a nécessité une connaissance sans faille du fonctionnement des scripts, de leurs interfaces, et ce dans l'univers de Unity.

Les objets à interaction se décomposent en deux classes, que sont :

- Les consommables. Ces objets sont stockables sur la ceinture, en mains, et possiblement autre part par la suite. Ils sont récupérés par le joueur uniquement s'il est suffisamment proche, et ne sont utilisés que lorsqu'ils sont nécessaires. Les exemples de tels objets dans notre jeu sont les suivants :
 - Les armes et boucliers, portés chacun dans sa main respective, permettent au joueur de faire varier ses statistiques de puissance et d'armure.
 - Les clés, portées au nombre maximum de 3 sur la ceinture, sont utiles pour ouvrir les portes fermées à clé.
 - Le totem, ou potion de résurrection, s'utilise quand le joueur meurt, et a pour effet sa résurrection instantanée, à l'endroit même où il est mort.
- Les objets à actions directes. Pour ces objets, l'action intrinsèque à l'objet se fait instantanément, automatiquement, dès que le joueur est suffisamment proche. Les exemples de tels objets dans notre jeu sont les suivants :
 - Les coffres, s'ouvrant et faisant apparaître une arme ou un bouclier.
 - Les potions, augmentant les statistiques de santé, de vitesse, ou d'endurance du joueur.
 - Les portes, s'ouvrant avec une clé si verrouillée, s'ouvrant toute seule sinon.

Maintenant, parlons de l'implémentation des interactions dans le jeu de manière plus factuelle. Étant donné un fonctionnement très proche, les scripts des objets à interactions déclinent tous d'une classe "*Interactable*". Cette classe permet de savoir si un joueur peut interagir avec l'objet. Seules les fonctions abstraites nommées "*Action*" et "*Used*" sont écrasées dans les scripts respectifs des objets.



FIGURE 19 – Equipement

Combat

Abordons désormais l'ajout du système de combat dans *Dungeon BattleRush*. L'implémentation des statistiques du joueur et de l'ennemi était un incontournable avant de se lancer dans la création du système d'attaque. Y sont stockés les points de santé, d'armure, de puissance, de vitesse, et d'endurance de l'entité possédant ce dit script. Y sont aussi présentes les fonctions de gains de statistiques, d'attaques et fatalement de perte de points de vie.

Après cela fait, nous dûmes implémenter le système d'attaque à proprement parler. Factuellement, le combat fonctionne de la manière suivante :

- L'arme possédée par le joueur va "s'activer" lorsque l'animation d'attaque de ce dernier va être lancée.
- Dès que l'arme va entrer en contact avec la victime, elle va demander à ce que le script de statistique de l'assaillant attaque les points de vie de la victime. Y seront évidemment déduits les points d'armure de cette dernière.
- La victime est alors stockée dans la mémoire de l'arme pendant une courte durée de temps, afin que la victime ne soit pas attaquée plusieurs fois lors d'une seule animation.

Nous avons choisi un tel fonctionnement, car le fait que ça soit l'interaction entre l'arme et la victime qui fasse l'attaque rajoute du réalisme, ainsi qu'une notion d'esquive et d'anticipation.

3.6.3 Randomisation

Le responsable de la randomisation est Florent Sagot, et la suppléante est Léa Margery. Contrairement à la soutenance dernière, le domaine de la randomisation s'est vu être fortement utilisé dans le développement de *Dungeon BattleRush*. Pour rappel, la randomisation est un aspect aléatoire dans le gameplay d'un jeu. C'est pour cela qu'implémenter cet aspect-ci nous a semblé si important dans notre jeu.

La randomisation possède une place prédominante dans la récupération des butins et objets du joueur. C'est pourquoi les pièces d'équipement apparaissent selon une probabilité aléatoire pondérée lorsque l'on ouvre un coffre. Mais ce système était déjà implémenté lors de la soutenance précédente. Mais entre-temps, il fut énormément optimisé. L'optimisation s'est principalement axée sur la recherche dans le dictionnaire des armes et boucliers. Pour rappel, chaque arme et bouclier sont représentés dans un dictionnaire C#, et sont associés à leurs statistiques respectives, ainsi qu'à leur chance d'apparition dans les coffres.

Pour encore plus assoir la place de la randomisation dans les butins, lorsqu'une pièce d'équipement est instanciée, elle peut être de 5 en dessous ou au-dessus de sa statistique nominale.

De même pour les potions. Elles sont instanciées aléatoirement parmi les 3 types disponibles, et leur bonus de statiques peut varier entre +3% et +7%.

Enfin, la partie la plus avancée depuis la dernière soutenance en termes de randomisation est la création de la carte de jeu de façon tout à fait aléatoire. Nous avons précédemment vu que le donjon était un rectangle, remplis de salles aléatoire. Pour rendre la création d'un tel donjon possible, nous avons procédé de la sorte :

- Une matrice de la taille souhaitée du donjon est créée, chaque case de ce tableau contenant une valeur aléatoire.
- Nous allons ensuite parcourir cette matrice, en instanciant sur la carte du jeu au bon endroit les salles correspondantes aux valeurs contenues dans les cases.

3.6.4 Effets des statistiques

Nous avons précédemment vu que pour rendre le combat possible, nous avons dû créer une classe contenant les statistiques de entités les contenant. Désormais, les potions augmentent les statistiques depuis ce dernier, et les variables des déplacements se retrouvent également ici. Ce script fut alors le nouveau centre, le cœur du système de statistique.

3.7 Git repository

La responsable de la gestion du repository git est Oriane Margelisch, et le suppléant est Florent Sagot.

Nous hébergeons toujours notre projet sur GitLab pour la gestion du partage de notre jeu, car nous sommes toujours satisfaits de ce que cette plateforme nous apporte. Nous rencontrons que peu de difficultés (des conflits de temps à autres en temps de travail intensif), mais la prise en main de GitLab s'est faite naturellement dans le groupe et son utilisation est à l'heure actuelle maîtrisée par l'ensemble des membres d'InfiniTeam. La méthode de fonctionnement initiale est toujours d'actualité : notre repository est composé de six branches, auxquelles nous avons rajouté deux branches de sauvegarde, ayant toutes un intérêt fondamental :

- *master* : branche principale du projet, nous faisons des merge depuis la branche *wip* une fois par semaine, généralement le week-end lors de notre réunion de groupe hebdomadaire, et deux à trois fois par jour en période d'activité intense.
- *wip* (pour *work in progress*) : branche secondaire du projet créée suite à des problèmes en début de projet. Nous l'actualisons depuis nos branches respectives afin de tous être à jour dans la version du jeu que nous modifions.
- branches individuelles (*maxime, oriane, léa, florent*) : à chaque session de travail, qu'elles soient individuelles ou collectives, nous récupérons les éventuelles modifications apportées au jeu par nos collègues de travail sur la branche *wip* en faisant un merge, nous pouvons ainsi travailler sur nos branches attitrées, et en fin de session nous mettons nos modifications sur *wip* en faisant un merge depuis celles-ci. Cette méthode de fonctionnement permet d'éviter la corruption de fichiers et les conflits sur notre branche secondaire, apportant ainsi une première visibilité de nos changements sur git, et dans le cas où une complication surviendrait tout de même, nous avons tous des versions actualisés de *wip* sur nos branches, ce qui permet de régler tous les problèmes sans soucis.
- *soutenance 1* : sauvegarde d'une version de notre jeu au moment de la première soutenance. Celle-ci sera rendue disponible sur le site web du jeu.
- *soutenance 2* : seconde sauvegarde contenant la version actuelle du jeu au moment de la soutenance. Même objectif qu'avec la branche *soutenance 1*.

Le repository git est accessible depuis le site web du jeu.

4 Avances et retards

Au cours de notre avancement dans le projet, nous avons maintenu une organisation qui nous a permis de progresser efficacement. De notre point de vue, les objectifs fixés pour la première soutenance ont été pour la plupart dépassés, excepté le multijoueur pour lequel nous rencontrons quelques difficultés.

	Soutenance 1	Soutenance 2	Soutenance 3
Graphismes	20%	60%	100%
Interfaces	30%	70%	100%
Audio	10%	25%	100%
IA	20%	50%	100%
Réseau	50%	80%	100%
Mécaniques de jeu	30%	70%	100%

TABLE 1 – Planning

	Prévu	Achevé
Graphismes	60%	60%
Interfaces	70%	85%
Audio	25%	50%
IA	50%	70%
Réseau	80%	70%
Mécaniques de jeu	70%	90%

TABLE 2 – Avancement

4.1 Rappels sur ce qui a été fait

Pour résumer, de nombreuses améliorations ont été ajoutées à notre jeu. Concernant la partie graphique, nous disposons aujourd'hui de deux salles de combat, une salle d'énigme, une salle dédiée à l'apparition du joueur et l'arène finale. Les problèmes que nous avons rencontrés à la soutenance précédente ont été réglés. De plus, de nouvelles animations ont été mises en place pour les objets ainsi que pour la seconde IA importée. D'autre part, nous sommes parvenus à dessiner le logo du jeu. Le HUD ainsi que le menu permettant au joueur de choisir son avatar ont été implémentés et sont fonctionnels. Nous avons assemblé des musiques libres de droits pour et les avons importées dans le jeu (menu principal, jeu, arène finale). La première IA est fonctionnelle et une seconde a été implémentée. De plus, les interactions des objets ont été développées et le combat et les statistiques sont implémentés. Le donjon est généré de façon aléatoire en utilisant les salles préfabriquées. Enfin, le Site Web a été mis à jour et est hébergé de façon temporaire.

4.2 Ce qui reste à faire

Pour la prochaine et dernière soutenance, nous devons créer davantage de salles, surtout des salles d'énigme, pour augmenter la diversité de celles-ci dans le donjon généré aléatoirement. D'autre part, nous devons trouver une musique que l'on pourra entendre lorsque le joueur a terminé et gagné la partie et réaliser les bruitages nous-mêmes si la situation nous le permet. Nous devons réparer le problème de déplacement dans le multijoueur et finir d'implémenter le choix d'un personnage dans celui-ci. Il faudra aussi implémenter la troisième IA ainsi que le boss de fin. De plus, nous devons disposer les salles qui doivent former les bords du donjon autour de celui-ci. Il faudra peaufiner quelques détails des IA. Puis nous ferons probablement quelques modifications uniquement esthétiques. Enfin, nous devons remettre à jour le site et le finir.

5 Conclusion

En conclusion, l'avancement du groupe dans la création de *Dungeon BattleRush* respecte à ce jour les objectifs fixés dans le planning du cahier des charges. Nous pouvons même dire que nous les avons dépassés pour la plupart.

Le lancement du jeu amène le joueur à un menu lui permettant de choisir un mode de jeu entre solo et multijoueur, d'accéder aux options du jeu et de le quitter. Un autre menu lui permet de choisir son personnage.

Les modes solo et multijoueur sont fonctionnels. L'instanciation du jeu se fait dans une première salle et le joueur a ensuite la possibilité d'évoluer dans un donjon généré aléatoirement mêlant salles de combat et d'énigmes. Les combats se font face à différentes intelligences artificielles autonomes constamment en mouvement que ce soit en patrouille, pour se rapprocher du joueur et le poursuivre ou encore pour l'attaquer. De plus, le personnage a la possibilité de récupérer des potions (lui augmentant ses statistiques de jeu), de récupérer des armes et des boucliers dans des coffres.

D'autres éléments du jeu sont implémentés, mais ne sont pas directement visibles en tant que tels lors du lancement du jeu, comme des déplacements et animations très complets des joueurs, la gestion du multijoueur à travers la création de salles de jeu (et la possibilité de jouer à distance dans la même salle) et la gestion des objets récupérés.

Nos objectifs pour la prochaine soutenance sont répertoriés dans le planning. Nous projetons de résoudre les actuels bugs et soucis que nous rencontrons, ainsi que de finir le développement du jeu en vue de la dernière soutenance. De plus, nous devons continuer de développer le site web, qui sera régulièrement mis à jour à jour au fil de l'avancement du projet.

Table des figures

1	Site Web - Accueil	5
2	Site Web - Télécharger	5
3	Site Web - Documents	6
4	Site Web - InfiniTeam	6
5	Site Web - Documents	7
6	Salle - 1.1	10
7	Salle - 1.2	10
8	Salle - 2	11
9	Salle - Arène finale	11
10	Coffre	12
11	Porte	12
12	Animation porte	13
13	Logo du jeu	14
14	Menus	16
15	HUD	16
16	Audio Mixer	17
17	Navmesh	18
18	Multiplayer	20
19	Equipement	22

Liste des tableaux

1	Planning	25
2	Avancement	25