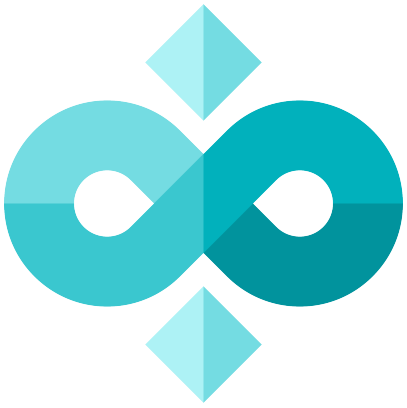
Dungeon BattleRush

par InfiniTeam

Rapport de Soutenance 1



Maxime Buisson
Oriane Margelisch
Léa Margery
Florent Sagot

Table des matières

1 Introduction

Ce rapport de soutenance a pour but de dépeindre l'avancement du projet Dungeon BattleRush depuis sa création.

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 Modification du Cahier des Charges

2.1 Modifications du texte

Une modification a été faite dans l'ensemble du cahier des charges concernant la statistique de protection du joueur. En effet, nous présentions initialement une armure que le joueur porterait liée à ses caractéristiques de protection. Nous avons finalement décidé de changer toutes les occurrences du mot "armure" par "bouclier" dans le cahier des charges, car cela nous semble plus adapté de lier ses caractéristiques de protection à son bouclier, et, par choix esthétique, nos joueurs n'auront pas d'armure interchangeable. La protection du personnage est donc désormais uniquement assurée par un éventuel bouclier.

2.2 Planning

La prévision faite concernant la partie réseau était initialement prévue pour atteindre 40% de la tâche, nous avons modifié cette prévision afin d'atteindre 50% au moment de notre première soutenance.

	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

2.3 Répartition des tâches

Certains rôles de responsables ou de suppléants ont été échangés au fur et à mesure de l'avancement du jeu et des préférences de chacun. La charge de travail individuelle reste toutefois équilibrée sur l'ensemble du projet.

	Maxime	Léa	Oriane	Florent
Graphismes				
Map (Carte de jeu)		R		S
Personnages et Objets	R		S	
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		S		R
Mécaniques de jeu				
Déplacements		S		R
Interactions et Combats		R		S
Effets des statistiques	S		R	
Randomisation	R		S	
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

R : Responsable, S : Suppléant, X : Participant

TABLE 1 - Répartition des tâches

	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

R : Responsable, S : Suppléant, X : Participant

TABLE 1 - Répartition des tâches

TABLE 2 – Ancien puis nouveau tableaux de la répartitions des tâches

3 Avancement

3.1 Graphismes

3.1.1 Map (Carte de jeu)

La responsable du design des salles du donjon est Léa Margery et le suppléant est Maxime Buisson. Nous avons importé des assets soigneusement sélectionnés afin que les objets puissent correspondre au mieux au thème médiéval et fantastique du jeu, de plus tous les assets sont en lowpoly. Nous les avons modifiés afin que les objets puissent être intégrés dans une salle avec nos personnages. Pour le moment, nous n'avons qu'une scène dédiée au combat, qui nous a permis de tester les différentes animations et interactions de notre personnage. Les dimensions de cette salle serviront de prefab pour la création d'autres salles. Puisque nous devrons par la suite randomiser entièrement le donjon, qui sera constitué de plusieurs salles préfabriquées, le fait qu'elles soient de la même taille nous simplifiera la tâche. La prochaine étape consistera donc à créer davantage de salles, dont certaines dédiées à la résolution d'énigmes, et de créer un donjon en randomisant les positions de ces salles.

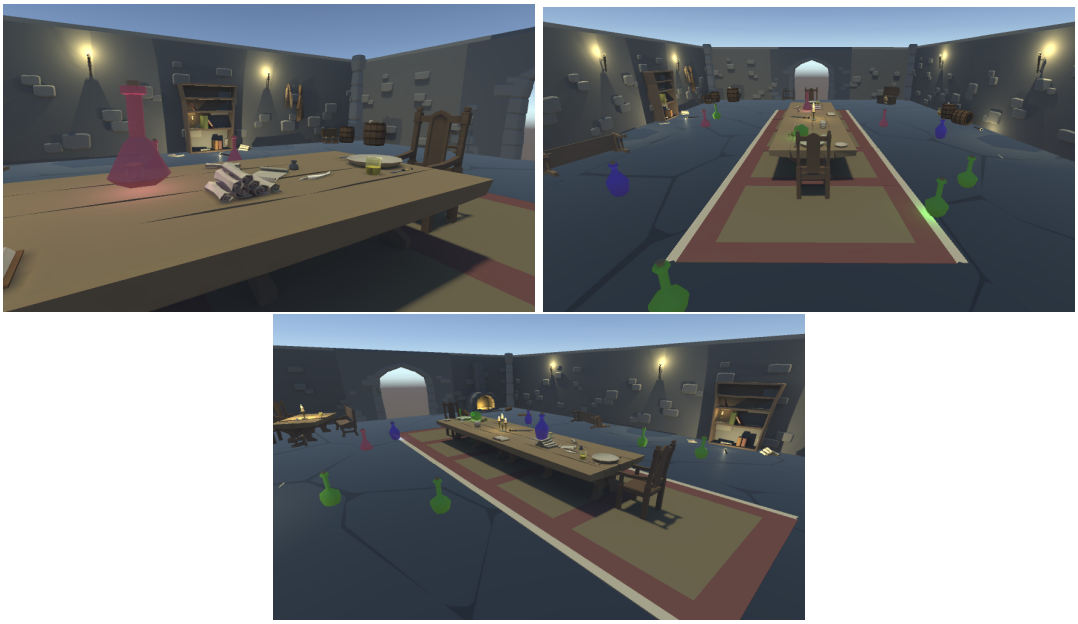


FIGURE 1 – Salle constituant la map

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 personnages principaux 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 avions 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 proviennent également du Unity Assets Store, et nous les avons obtenus gratuitement.

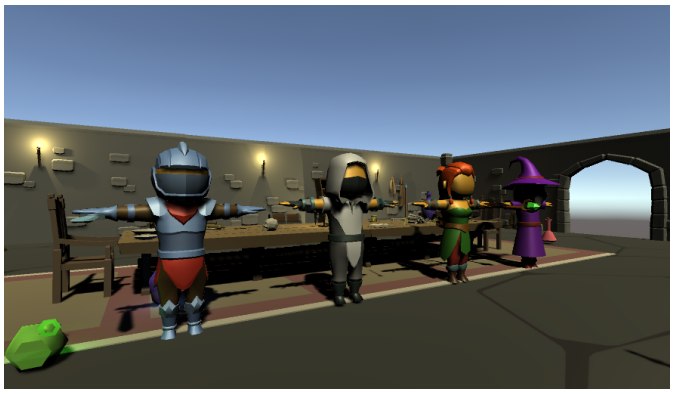


FIGURE 2 – Modèles des joueurs

FIGURE 3 – Modèles des ennemis

3.1.3 Animations

Le responsable des animations est Maxime Buisson et la suppléante est Oriane Margelisch. Nous avons importé des animations de déplacements fondamentaux (telles que la marche avant et arrière, la course, le saut et les déplacements en diagonales), que nous avons liés au personnage afin que les mécaniques de jeu, représentées par les déplacements, les attaques, le saut, la mort du joueur et quand celui-ci reçoit des attaques ennemies puissent prendre une forme concrète et réaliste, lui donnant ainsi une certaine forme de vie. Pour chaque changement d'état (par exemple marcher puis courir), un booléen est nécessaire dans le script ainsi que dans l'onglet "animator", permettant une transition entre les animations.

3.1.4 Design

Le responsable des designs et logo est Florent Sagot et la suppléante est Léa Margery. La recherche du logo de l'équipe InfiniTeam s'est fait au travers d'une recherche approfondie sur un banque d'image nommée FlatIcons, et sur le compte d'une entreprise nommée Freepik, spécialisée dans la création de logo libres d'utilisation. C'est ainsi que nous trouvâmes notre logo. La police d'écriture sur notre logo est une police d'écriture nommée Blue, et trouvée sur le site DaFont.

Cependant, nous n'avons pas encore de logo pour notre jeu, et nous comptons y remédier pour la deuxième soutenance. En effet, étant donné que l'univers du jeu n'est pas encore suffisamment complet, trouver un logo qui le représente peut s'avérer compliqué, voire inutile.

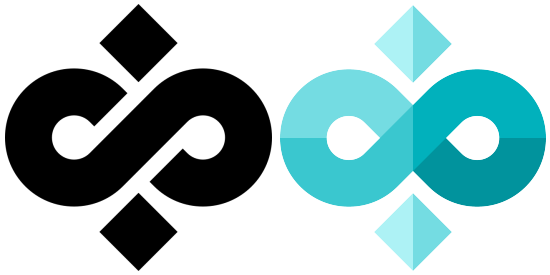


FIGURE 4 – Ancien puis nouveau logo simple

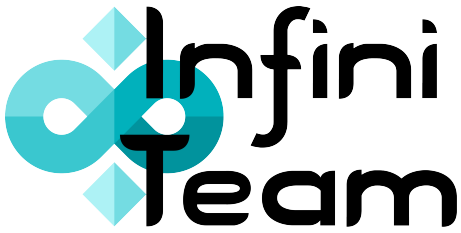


FIGURE 5 – Logo final complet

3.2 Interfaces

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

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 Options :

  — 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

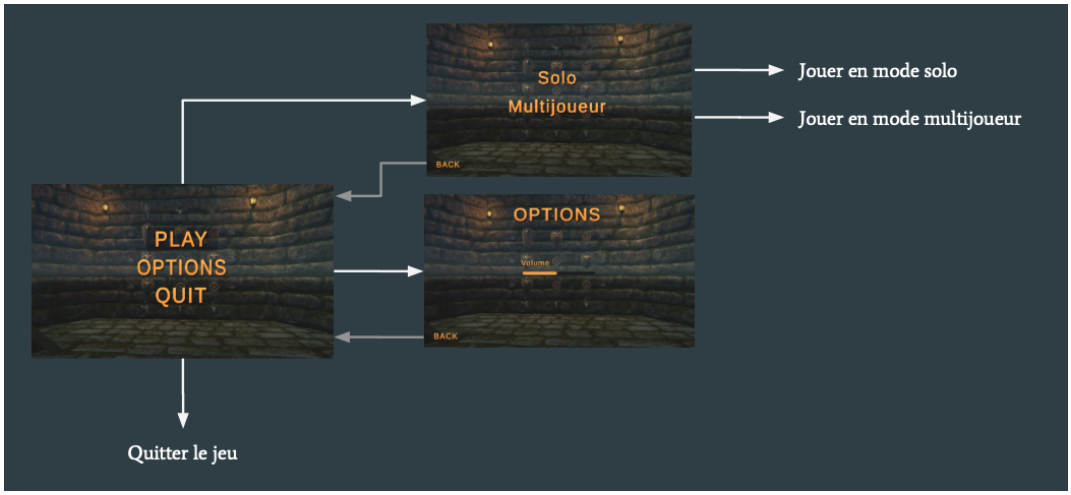


FIGURE 6 – Graphique représentant la navigation dans les menus

3.3 Audio

L'audio est une partie que nous avons prévu d'approfondir lorsque notre avancement dans la création du jeu sera plus ample. Nous avons donc actuellement une musique libre de droits pour notre jeu, ainsi qu'une sélection d'autres pour nos différents menus et niveaux. Le thème retenu pour nos différentes musiques est donjon ou héroïc fantasie, avec un florilège de musiques calmes et épiques pour les niveaux à énigmes et de combats, ainsi que pour les menus.

3.4 IA

L'IA est actuellement capable de :

— 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.

— d'arrêter de se déplacer lorsque le joueur quitte son périmètre de détection, et le lancement d'une animation spéciale informant le joueur.

Le jeu étant à son état embryonnaire, nous avons décidé d'implémenter une IA minimaliste qui vérifie à chaque frame la distance entre le joueur et elle-même. Cela est possible lorsque le joueur est placé en paramètre dans son script en tant que "cible". Les déplacements de L'IA sont faits à travers une fonction qui va calculer 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 le "NavMesh agent" où notre IA peut se déplacer (zones bleues ci-dessous). Ainsi, celle-ci aura une nouvelle position en fonction de sa cible et du "NavMesh" établis.

FIGURE 7 – Zone accessible par l'IA (NavMesh)

3.5 Réseau

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

Nous avons choisi d'utiliser le package Unity, Photon Unity Networking (PUN 2), la version gratuite étant tout à fait suffisante pour ce que nous voulons faire dans Dungeon BattleRush. 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 un matchmaking simple, puisqu'il n'y a pas besoin de rentrer un code, mais simplement de cliquer sur le bouton permettant de lancer une partie.

3.5.1 Lobby

Dans le lobby, nous pouvons :

— Rejoindre une room aléatoirement

— Créer une room aléatoirement (nom aléatoire et options de la room)

Dans un Lobby, si le joueur est connecté aux serveurs Photon, il peut rejoindre une room aléatoire. S'il ne parvient pas à rejoindre une room, c'est parce qu'aucune n'a encore été créée.

Alors une nouvelle room aléatoire est créée. Si la création de la room échoue, la tentative est réitérée.

Lors du chargement d'une room, 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.

Dans notre Lobby, nous utilisons la valeur booléenne PhotonNetwork.AutoSyncScene qui est instanciée à "true" dès le début et qui nous sera très utile pour rejoindre une room.

3.5.2 Room

Si le joueur rejoint une room et qu'il s'agit d'un master client (le créateur de la room), alors le joueur se retrouve dans la scène multijoueur. Si le master client quitte la partie en cours, alors ce rôle sera redistribué à un joueur présent dans la room. Puisque, nous utilisons la valeur booléenne PhotonNetwork.AutoSyncScene, tout autre joueur qui essaye de rejoindre cette room, après l'arrivée du master client dans la scène multijoueur, d'arriver dans cette même scène.

3.5.3 Création 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 prefabs des joueurs. Pour créer un joueur, celui-ci est instancié directement à partir du dossier ressource Photon contenant les prefabs. A l'heure actuelle, nous pouvons créer jusqu'à 4 personnages, mais nous rencontrons des difficultés pour les déplacements de ceux-ci. En effet, il semble que le mouvement d'un joueur entraîne celui des autres joueurs.

FIGURE 8 – Connexion multijoueur avec une personne

FIGURE 9 – Connexion multijoueur avec deux personne

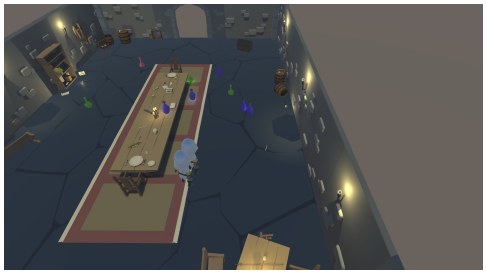


FIGURE 10 – Multijoueur problème

3.6 Mécaniques de jeu

3.6.1 Déplacements

Le personnage contrôlé par le joueur se déplace à l'aide de vecteurs de translation selon les trois dimensions que sont les axes x, y et z. Celui-ci se dirige en rotation suivant le curseur de la souris : en effet, 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 cela relatif à l'orientation de la souris. Le joueur subit la gravité, ce qui est une propriété essentielle à prendre en compte lors de l'instanciation du saut.

L'orientation du personnage se fait donc grâce à la souris. Si l'on considérait le plan du jeu comme un plan 2D, le joueur regarderait le pointeur. Cela est possible par un ingénieux système fonctionnant comme suit :

— Une droite est tirée suivant les deux points formés par l'objet caméra et le pointeur de la souris.

— Cette droite intersecte obligatoirement le plan horizontal de hauteur (sur l'axe y) celle du joueur,

— Un point physique en est déduit à cette intersection,

— Enfin, le joueur oriente son regard vers cedit point.

La sensation de déplacement dans le monde de notre jeu est donc des plus fluides et agréable.

3.6.2 Interactions et Combats

Le responsable des interactions et combats est Florent Sagot, et la suppléante est Léa Margery. Tout d'abord, commençons par dire que le combat n'est pas encore implémenté. En effet, seules les animations de combat sont présentes dans le jeu. Cependant, et cela se voit sur notre planning d'avancement, nous avons bien avancé dans les interactions. En effet, beaucoup d'actions avec les objets interactifs primaires, que sont les potions et l'équipement, sont déjà possibles, comme ramasser des pièces d'équipement, ou encore obtenir des potions qui viennent vers vous dès que vous êtes suffisamment proche, à la manière d'un aimant.

Mais l'implémentation factuelle dans le développement sur Unity s'est vue plus longue que prévu.

Afin d'expliquer factuellement la technique de développement adoptée, voici tout d'abord le fonctionnement récurrent de recherche d'objet d'interaction :

— Le script va créer une liste des objets étiquetés selon leur nature.
— Puis, le script va itérer dans cette liste en trouvant l'objet le plus proche, réunissant les conditions de distance minimale d'interaction, et autres conditions.
— Enfin, deux possibilités :
  — Si le joueur est l'acteur principal de l'action, et si le joueur appuie sur une touche d'interaction (ramasser, lâcher, ou autres), l'action adéquate se produit.
  — Si c'est un autre objet qui est l'acteur principal de l'action, par exemple les potions, l'objet adopte un comportement adéquat.

De plus, pour donner l'illusion que le joueur porte dans la main une pièce d'équipement, il suffit de faire en sorte que l'objet que l'on veut porter soit fils de la main du personnage. Il suivra donc parfaitement les mouvements de cette main, et donnera ainsi l'illusion voulue. Pour l'anecdote, nous eûmes rencontré un problème, celui-ci étant que l'objet bougeait dans la main du personnage. Cela était dû à la gravité, et nous avons résolu ce problème en désactivant la gravité de l'objet quand il est en main, et de la remettre quand il est lâché.

3.6.3 Randomisation

Le responsable de la randomisation est Florent Sagot, et le suppléant est Oriane Margelisch. Pour rappel, la randomisation est un aspect aléatoirité dans le gameplay d'un jeu. C'est pour cela qu'implémenter cet aspect-ci nous a semblé si important dans notre jeu. Pour le moment, cette aléatoirité s'applique lors de la création d'une potion, leur puissance d'influence sur le joueur, et enfin lorsqu'un coffre fait apparaître une arme ou un bouclier.

Pour rappel, dans notre jeu, il y a trois types de potions, représentés par des bouteilles de couleurs différentes. Lors de l'instanciation d'une potion sur la scène de jeu, une potion est tirée aléatoirement pour apparaître. Cela est rendu factuellement possible dans le développement sur Unity grâce à un GameObjet ayant pour fils les trois potions. Lorsque ce dernier est instancié, il va, grâce à la classe aléatoire, choisir une potion à garder, supprimer les deux autres, se défaire parent de la potion, et enfin s'autosupprimer, et ça dans un but d'optimisation.

Le deuxième point à aborder est la création des armes ou boucliers lorsque le coffre est ouvert. Pour cela, deux dictionnaires (C#) ont été créés, respectivement pour les armes et les boucliers, ainsi que leurs caractéristiques et chances d'apparition respectives. Lorsque le coffre va vouloir faire apparaître un de ces objets, il appelle une fonction qui va tirer aléatoirement, dans ces dictionnaires, une arme en fonction de leur probabilité de tirage.

Dans le futur, ce domaine affectera bien plus d'aspects du jeu, comme par exemple la disposition des salles dans le donjon, du nombre d'ennemis dans une salle, ou encore la répartition des objets dans cesdites salles.

3.6.4 Effets des statistiques

Le responsable de la randomisation est Florent Sagot, et la suppléante est Léa Margery. L'implémentation des effets des statistiques est, lors de la rédaction de ce rapport, déjà développée. Cependant, cela n'est pas encore d'actualité, car l'entièreté du gameplay du joueur n'est pas encore implémentée.

Pour rappel, seuls les objets interactifs que sont les potions, les armes et les boucliers ont un effet sur les statistiques du joueur. De plus, il est important de préciser que les potions affectent la santé du joueur, sa vitesse et son endurance. Les armes font-elles varier la statistique de puissance, et enfin les boucliers celle d'armure. Lorsque le joueur interagit avec ces objets, le script de celui-ci va récupérer dans les scripts de chaque objet récupéré les informations propres à ces derniers, et va alors modifier les caractéristiques du joueur en conséquence.

3.7 Git repository

La responsable de la gestion du repository git est Oriane Margelisch, et le suppléant est Florent Sagot. Nous avons choisi d'utiliser GitLab pour la gestion du partage de notre jeu, car la proportion avantages/inconvénients nous paraissait être la mieux répartie face à ses concurrents (entre autres GitHub et UnityCollab).

Au commencement du projet, nous avons eu quelques difficultés, notamment vis-à-vis des branches, le temps de trouver une façon de travailler qui nous convenait au mieux. Nous avons donc retenu comme méthode de fonctionnement ce qui suit : notre repository est composé de six branches, 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 tous les soirs en période d'activité intense.

— *wip* (initialement *work*) : branche secondaire du projet. Nous avions au début une branche ayant la même utilité nommée *work*, sur laquelle nous avons eu une corruption de certains de nos fichiers la semaine du 22 février. Grâce à nos branches individuelles (voir ci-dessous) nous avons pu aisément pallier ce problème. Nous l'avons donc refaite, en la renommant *wip* pour "work in progress".

— 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.

Le repository git est accessible depuis le site web du jeu. Voici un petit tutoriel que Oriane a écrit sur notre serveur Discord indiquant la démarche pour un travail en sécurité.

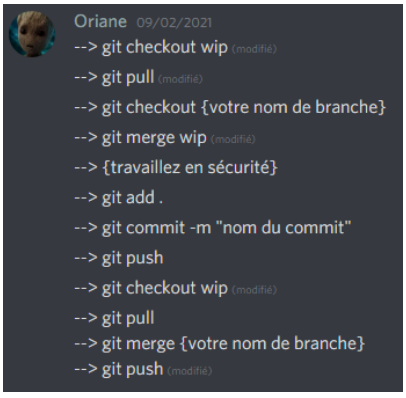


FIGURE 11 – Consignes à suivre Git

3.8 Web

Le responsable des interactions et combats est Florent Sagot, et la suppléante est Léa Margery. Le site web, comme énoncé dans le cahier des charges, a été fait sans outils interactifs de création de site web. Il a donc été créé avec SublimeText 3, et hébergé en localhost 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.

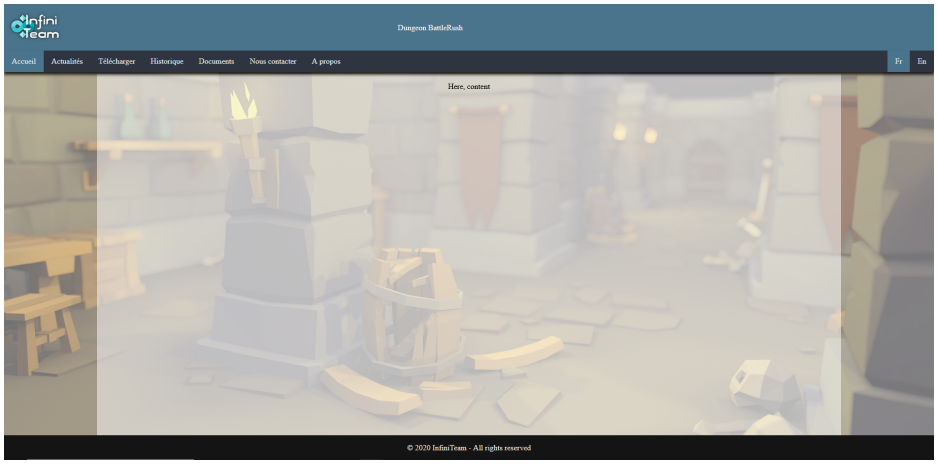


FIGURE 12 – Page web d'accueil

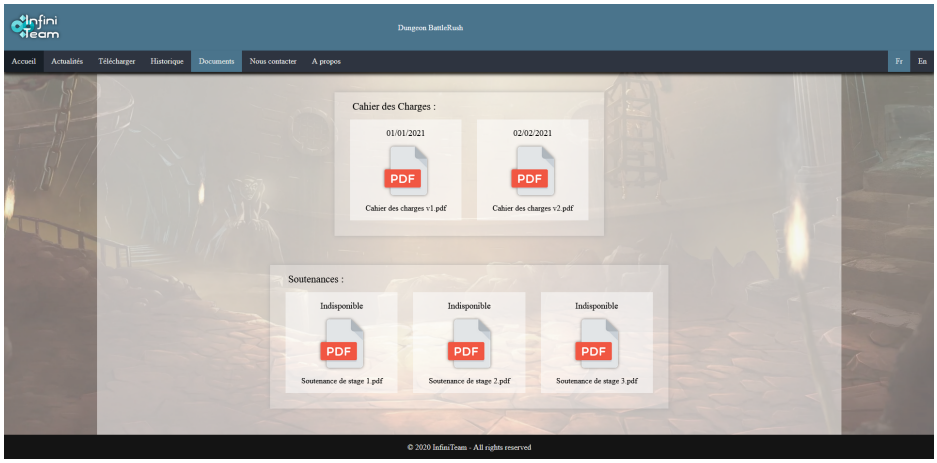


FIGURE 13 – Page web documents

4 Avances et retards

Au cours de notre avancement dans le projet, nous sommes parvenus à développer une organisation qui nous a permis de progresser efficacement. De notre point de vue, les objectifs fixés pour la première soutenance ont été atteints, voire dépassés pour certains.

	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 3 – Planning

	Prévu	Achevé
Graphismes	20%	40%
Interfaces	30%	30%
Audio	10%	10%
IA	20%	20%
Réseau	50%	50%
Mécaniques de jeu	30%	35%

Table 4 – Avancement

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.

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. Les modes solo et multijoueur sont fonctionnels, c'est-à-dire que les premiers niveaux de batailles et d'énigmes n'ont pas encore été implémentés, mais le joueur a pour l'instant la possibilité de se déplacer dans la salle (à plusieurs en mode multijoueur), de récupérer des potions (lui augmentant ses statistiques de jeu), de récupérer des armes dans des coffres et de fuir une IA qui le suit quand il se trouve dans un périmètre de détection donné. D'autres éléments du jeu ont été implémentés mais ne sont pas directement visibles en tant que tels lors du lancement du jeu, comme des scripts de randomisation de certains objets, 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), la gestion des objets récupérés et les scripts d'intelligences artificielles.

Nos objectifs pour la prochaine soutenance sont répertoriés dans le planning, nous n'avons pas eu besoin de les revoir à la hausse ou à la baisse. Nous projetons de résoudre les actuels bugs et soucis que nous rencontrons, ainsi que de continuer le développement du jeu en vue de la deuxième soutenance. De plus, nous devrons continuer de développer le site web, qui sera régulièrement mis à jour à jour au fil de l'avancement du projet.

Table des figures

Liste des tableaux