Le projet que nous avons réalisé en équipe consiste en la conception et le développement d'un jeu de tower defense en Java avec la bibliothèque Swing. Un tower defense est un genre de jeu stratégique où le joueur doit placer des tours de défense pour arrêter des vagues successives d'ennemis avant qu'ils n'atteignent un point vulnérable. Dans notre cas, nous avons fait le choix de développer un jeu de tower defense à deux dimensions avec des graphismes simples, mettant l'accent sur la stratégie et la gestion des ressources.

Structure du Code et Communication entre les Composants

L'architecture du code de notre jeu de tower defense est soigneusement conçue pour assurer une organisation claire, une maintenance efficace, et une communication fluide entre les différents éléments du jeu.

Arborescence du Projet

Le projet est segmenté en deux dossiers principaux : `bin` et `src`. Le dossier `bin` héberge les fichiers binaires, y compris toutes les ressources graphiques nécessaires au jeu, telles que les images au format PNG, GIF, etc. D'un autre côté, le dossier `src` contient l'ensemble du code source.

À l'intérieur du dossier `src`, nous avons organisé le code en sous-dossiers pour mieux structurer les différents composants du jeu. Le dossier `data` renferme les ressources utilisées par le jeu, tandis que les dossiers `models` et `views` contiennent respectivement les classes définissant les modèles du jeu (tours, ennemis, vagues) et celles qui gèrent l'interface utilisateur.

Communication entre les Composants

Dossier `models`

Dans le dossier `models`, chaque classe a un rôle spécifique dans la représentation des éléments du jeu. Par exemple, la classe `Tower` définit les caractéristiques et le comportement des différentes tours disponibles dans le jeu. De même, les classes `EnnemisType1` et `EnnemisType2` définissent les propriétés spécifiques des deux types d'ennemis présents.

Les classes dans le dossier `models` interagissent entre elles pour créer un écosystème de jeu cohérent. Par exemple, la classe `Wave` orchestre le déclenchement des vagues d'ennemis, interagissant avec les classes `Ennemis` pour générer des ennemis de types variés.

Dossier `views`

Le dossier `views` contient les classes responsables de l'interface utilisateur. La classe `Main` agit comme point d'entrée du programme, instanciant les différentes vues et lançant le jeu. Les classes telles que `hautMenu` et `PauseMenue` gèrent les menus du jeu, tandis que `Spiel` est responsable de l'affichage et de la gestion de la carte principale du jeu.

Les vues interagissent également avec les classes du dossier `models`. Par exemple, la classe `Game` doit connaître l'état des vagues d'ennemis pour afficher correctement la progression du jeu à l'utilisateur. Cette interaction est réalisée en utilisant des méthodes et des mécanismes de communication définis dans le code.

Collaboration et Maintenance

La séparation nette entre les dossiers `models` et `views` favorise une collaboration efficace entre les membres de l'équipe de développement. Chaque composant est isolé dans son domaine de responsabilité, facilitant la compréhension du code et permettant des modifications spécifiques sans perturber l'ensemble du système.

Caractéristiques du Jeu

Map et Niveaux de Difficulté

L'une des caractéristiques essentielles de notre jeu réside dans la diversité offerte aux joueurs à travers deux cartes distinctes : une au style **CyberPunk** et une autre inspirée de l'univers de **Mario**. Chacune de ces cartes propose un environnement visuel unique, créant une expérience immersive et captivante pour les joueurs. Le choix entre ces deux mondes offre non seulement une variété esthétique mais introduit également des dynamiques de jeu différentes, incitant les joueurs à adapter leurs stratégies en fonction de l'environnement choisi.

Pour renforcer l'aspect progressif et adaptatif du jeu, nous avons mis en place trois niveaux de difficulté croissants : Débutant, Intermédiaire, et Expert. Ces niveaux influent sur divers paramètres, offrant ainsi une expérience de jeu adaptée à différents niveaux de compétence des joueurs.

- **Débutant :** Un point d'entrée doux dans le jeu, idéal pour les nouveaux joueurs. Ce niveau offre une monnaie de départ généreuse, un nombre d'ennemis limité, des dégâts initiaux faibles, et une vitesse de déplacement des ennemis réduite. Les paramètres sont ajustés pour permettre aux joueurs de se familiariser avec les mécaniques du jeu et de développer leurs compétences stratégiques.

- **Intermédiaire :** Un niveau intermédiaire qui constitue une transition vers des défis plus complexes. Avec une monnaie de départ moins importante, un nombre d'ennemis accru, des dégâts nécessaires pour éliminer un ennemi augmentés, et une vitesse de déplacement

des ennemis plus rapide, ce niveau exige une stratégie plus élaborée. Les joueurs doivent affiner leurs compétences pour survivre aux vagues successives d'ennemis.

- **Expert :** Le niveau de difficulté ultime pour les joueurs chevronnés. Avec une monnaie de départ limitée, un nombre d'ennemis substantiel, des dégâts considérables nécessaires pour éliminer un ennemi, et une vitesse de déplacement des ennemis maximale, ce niveau pousse les joueurs à la limite de leurs compétences. Seuls les stratèges les plus habiles pourront surmonter les défis complexes que propose ce niveau.

Types d'Ennemis, Tours et Building

Inspiré du jeu Bloons TD2, notre jeu met en avant la diversité à travers deux types d'ennemis distincts, chacun apportant son propre style visuel et ses caractéristiques uniques, ajoutant une profondeur stratégique au gameplay.

- **EnnemisType1** : Ces ennemis arborent un style visuel particulier et possèdent des attributs spécifiques qui les rendent uniques. Leur comportement imprévisible et leurs caractéristiques distinctes obligent les joueurs à adapter constamment leur tactique en fonction de la variété des ennemis rencontrés.

- **EnnemisType2 :** Un deuxième type d'ennemi avec un style visuel distinct et des traits spécifiques. L'introduction de ces ennemis enrichit davantage le gameplay en présentant de nouveaux défis et en exigeant des ajustements tactiques pour les contrer efficacement.

Pour contrer ces ennemis, les joueurs ont à leur disposition trois types de tours, chacune avec une fonctionnalité unique, et l'ensemble du système est alimenté par une économie monétaire ingénieuse.

- **Tour Neutralisante** : Une tour capable de neutraliser les ennemis pendant une période courte, offrant une pause stratégique précieuse aux joueurs. L'achat de cette tour nécessite une dépense monétaire, encourageant ainsi une gestion prudente des ressources.

- **Tour Laser :** Une tour puissante qui inflige des dégâts continus aux ennemis traversant son rayon, idéale pour éliminer les ennemis robustes. L'amélioration de la portée et des dégâts de cette tour est possible moyennant une dépense monétaire supplémentaire.

- **Tour à Rafale :** Une tour qui tire des projectiles un à un, fournissant une stratégie de tir plus précise pour éliminer sélectivement les ennemis. Comme pour les autres tours, l'amélioration de la portée et des dégâts est possible en échange de monnaie, incitant les joueurs à investir judicieusement pour maximiser leur efficacité.

Les joueurs ont également à leur disposition un **Building** qui aide à la régénération d’Energy.

La variété d'ennemis et de tours crée une expérience de jeu riche et stimulante, mais la gestion monétaire devient également une composante stratégique essentielle. Les tours ne sont pas seulement des moyens de défense, mais aussi des investissements nécessitant

une planification financière.
Les joueurs doivent peser les avantages de chaque amélioration par rapport à la nécessité de construire de nouvelles tours pour faire face à des vagues d'ennemis de plus en plus redoutables.

Ainsi, la monnaie devient un élément clé du jeu, ajoutant une couche de complexité à la stratégie globale des joueurs. L'équilibre entre l'achat de nouvelles tours, l'amélioration des tours existantes, et la gestion des ressources monétaires devient une danse tactique où chaque décision financière influence directement la capacité du joueur à relever les défis croissants du jeu.

Gestion des Ressources et Stratégie

Au cœur de notre jeu de tower defense réside un système de gestion de ressources essentiel, où la Lifeline et l'Energy jouent des rôles cruciaux dans la survie du joueur et l'efficacité de sa défense.

Lifeline : La Vie en Jeu

La Lifeline représente la vie du joueur en temps réel. Chaque fois qu'un ennemi réussit à traverser le parcours prédéfini, la Lifeline diminue proportionnellement. La Lifeline est, en quelque sorte, la frontière entre la victoire et la défaite. Elle impose une pression constante sur le joueur pour maintenir une défense solide et éviter que trop d'ennemis n'atteignent le point vulnérable. La gestion prudente de la Lifeline devient donc une composante stratégique fondamentale pour assurer la survie à long terme dans le jeu.

Energy : La Ressource Tactique

Contrairement à la Lifeline, l'Energy représente une ressource dépensée à chaque tir de tour. Chaque fois qu'une tour est activée, de l'Energy est consommée pour effectuer le tir. Cependant, l'aspect distinctif de l'Energy réside dans sa capacité à se régénérer à chaque ennemi tué. Cette caractéristique crée une dynamique tactique unique, obligeant les joueurs à prendre des décisions éclairées sur l'utilisation de leurs tours et à considérer la régénération de l'Energy comme un facteur clé dans leur stratégie.

Stratégie et Optimisation

La stratégie du joueur repose sur l'optimisation de ces deux ressources interconnectées. Il doit prendre des décisions cruciales, telles que choisir le moment idéal pour déployer une tour en fonction du niveau d'Energy disponible, ou décider de sacrifier temporairement des points de Lifeline pour investir dans des améliorations stratégiques.

La régénération de l'Energy offre des opportunités pour des manœuvres tactiques, incitant les joueurs à définir des priorités parmi les cibles potentielles pour maximiser l'efficacité de leur défense tout en assurant la régénération continue de l'Energy.

En résumé, la gestion adroite de la Lifeline et de l'Energy constitue un élément clé de la stratégie globale du joueur. Cela ajoute une couche de complexité au gameplay, obligeant les joueurs à anticiper, à s'adapter et à optimiser constamment leurs ressources pour relever les défis croissants proposés par les différents niveaux de difficulté. La recherche d'un équilibre entre la défense immédiate et l'investissement à long terme devient ainsi le défi central du joueur dans la quête de la victoire.

Interface Utilisateur

L'interface utilisateur, gérée par les classes du dossier `views`, joue un rôle crucial dans l'expérience de jeu. Les différentes interfaces, telles que les Maps principale, les menus, et les éléments interactifs, sont conçues de manière à être intuitives et ergonomiques.

Défis Rencontrés et Solutions Apportées

Le développement du jeu n'a pas été sans défis. Parmi ceux-ci, la gestion des ressources, l'équilibrage des niveaux de difficulté, et la coordination entre les différentes classes ont nécessité une réflexion approfondie. Cependant, grâce à une communication efficace au sein de l'équipe, nous avons pu trouver des solutions adaptées à chaque défi rencontré.

Une communication qui s’est faite virtuellement grâce à des plateformes de développement collaboratif en ligne, avec Git en tant que système de contrôle de version. Cette approche a permis à chaque membre de l'équipe de travailler sur le code de manière indépendante, tout en assurant une synchronisation efficace des modifications apportées.

Résultats et Tests

Des captures d'écran du jeu en action, accompagnées de retours sur les tests effectués, illustrent les résultats obtenus. Ces tests ont permis de vérifier la stabilité du jeu, l'efficacité des mécaniques de jeu, et l'expérience globale du joueur. Des ajustements ont été apportés en fonction des retours obtenus, contribuant ainsi à l'amélioration continue du jeu.

Conclusion

En conclusion, ce projet de jeu de tower defense en Java avec Swing a été une expérience enrichissante. La collaboration entre binôme a permis de concrétiser un jeu captivant, offrant des défis stratégiques et une variété visuelle. Les enseignements tirés de cette expérience ouvrent la voie à des améliorations futures et à de nouvelles opportunités de développement.

Perspectives d'Avenir

Pour l'avenir, des améliorations telles que l'ajout de nouvelles cartes, de types d'ennemis, et de fonctionnalités de jeu pourraient être envisagées. Les retours des joueurs seront également essentiels pour affiner et étendre le jeu, contribuant ainsi à sa croissance continue, car oui nous voulons le mettre en ligne.

Remerciements

Nous tenons à exprimer notre gratitude envers nos enseignants qui ont contribué, de près ou de loin, à la réussite de ce projet. La collaboration et le soutien ont été des éléments clés de cette expérience de développement de jeu.

Cette réalisation démontre notre engagement envers la création de jeux interactifs et stimulants, et nous sommes impatients de poursuivre cette passion dans de futurs projets.