

Titre du projet : Développement d'une version personnalisée du jeu "Crossy Road"

Objectifs du projet :

Maîtrise des fondamentaux de la programmation : Les étudiants auront l'opportunité de renforcer leurs compétences en programmation en utilisant le langage de programmation C# et Unity3D pour développer les fonctionnalités du jeu.

Conception orientée objet : Le projet mettra l'accent sur la conception orientée objet, encourageant les étudiants à créer des classes et des objets bien structurés pour représenter les entités du jeu, tels que les personnages, les obstacles, et l'environnement.

Utilisation de l'analyse UML : Les étudiants apprendront à utiliser l'UML pour modéliser le système, en créant des diagrammes de classes, diagrammes d'activités, d'états et des cas d'utilisation. Cette phase d'analyse préliminaire permettra d'assurer une compréhension approfondie des interactions entre les composants du jeu.

Contrôle de code source et planification : Les étudiants apprendront à mettre en place et à utiliser un contrôle de code source (Exemple : git) et à planifier le développement avec un gestionnaire de tâche (exemple Trello) et d'outils comme le diagramme de GANTT

Gestion des événements et des animations : Les participants exploreront la gestion des événements pour contrôler les mouvements des personnages et des objets du jeu. De plus, ils intégreront des animations pour améliorer l'expérience utilisateur.

Personnalisation du jeu : Les étudiants auront l'opportunité d'apporter des modifications créatives au jeu, tels que l'ajout de fonctionnalités uniques, de personnages spéciaux, ou de défis supplémentaires, pour développer leur créativité et leurs compétences en résolution de problèmes.

Ce projet offre une combinaison idéale entre le développement de compétences techniques en programmation et la compréhension approfondie de l'analyse UML, préparant ainsi les étudiants à relever des défis concrets dans le domaine du développement logiciel.

Description du jeu :

📺 J'ai codé un robot pour ATOMISER Crossy Road.

Le but est d'avancer autant que possible à travers des routes, des rivières, de l'herbe et des voies ferrées, sans mourir.

Le joueur joue avec une mascotte (comme un poulet, koala, ou lapin). Il existe de nombreux obstacles qui causent immédiatement la mort, tels que les rivières, les voitures et les trains.

Pour surmonter ces obstacles, le joueur doit avancer lorsque le chemin est libre ou doit utiliser les plateformes mouvantes sur la rivière pour la traverser.

Pour décourager la marche au ralenti ou l'arrêt total, un aigle apparaît et s'empare de la mascotte s'il reste au même endroit pendant trop longtemps ou se déplace de trois pas en arrière, entraînant la perte de la partie.

Chaque déplacement d'un carré vers l'avant fera gagner un point. Tous les 50 points gagnés, un effet sonore se fait entendre. De plus, les pièces d'or utilisées pour déverrouiller les mascottes sont disséminées dans l'environnement

Développement attenduGénération automatique de la carte (2D ou 3D isométrique)

Il existe plusieurs lignes de terrain possibles :

- Herbe : certaines cases sont occupées par des buissons ou des arbres, ces derniers ne sont pas en mouvement,
- Route : Circulation de voitures ou de camions qui ne changent pas de voie de circulation et avec un espacement raisonnable entre chaque obstacle,

- Rail : Parcouru par des trains,
- Rivière : On retrouve sur les rivières des troncs d'arbres et des nénuphars qui permettent de traverser l'eau

Le jeu devra donc reprendre un minimum de :

- 1 terrain fixe
- 1 terrain avec objet mouvant
- 1 terrain non franchissable

Bien sûr vous pouvez adapter la thématique des terrain

Plusieurs éléments sont à prendre en compte

- Les différents types de lignes doivent se succéder de manière aléatoire.
- Les vitesses de déplacement des différents éléments sont fixes (toutes les voitures se déplacent à la même vitesse). En revanche, deux objets différents peuvent se déplacer à des vitesses différentes (un camion et un train n'iront pas à la même vitesse).
- Les objets peuvent se déplacer de droite à gauche ou de gauche à droite, mais une ligne ne changera pas de sens une fois affichée à l'écran.
- Les objets devront se succéder à l'infini sur une ligne visible.
- Toute collision entre le joueur et une voiture, un camion ou un train met fin à la partie. Le score est alors sauvegardé.
- Une rivière ne peut pas être franchie, le joueur devra obligatoirement utiliser un tronc ou un nénuphar pour passer.
- Le score se calcul en fonction du nombre de ligne que le joueur arrive à franchir sans mourir.

Interface

- Menu de démarrage (Jouer, Choix de difficulté, Classement par difficulté, Quitter ..)
- Menu de pause (Continuer, Recommencer, Quitter ..)
- Interface permettant de visualiser le score au cours du jeu, l'écoulement du temps de la partie, son meilleur score à battre
- Menu option (Activer/Désactiver le son)

Algorithme de pathfinding

- Création d'un robot capable de jouer au jeu.
- Le robot utilisera un algorithme de pathfinding et une prévision des collisions pour jouer

Evaluation

Le rendu sera à faire sous la forme d'un exécutable windows disponible dans le gestionnaire de source (branche release).

Le code sources devra également être accessible.

La date du rendu 17/05/2024 23h59 au plus tard. Tout sujet rendu après cette date ne sera pas évalué.

L'évaluation POO sera découpée en trois parties :

- 11 points attribués au projet (les fonctionnalités basiques du jeu)
- 4 points pour le robot
- 5 points pour la présentation : 20 minutes (orale + powerpoint + jeu) et 20 minutes de questions

L'évaluation MET aura pour objectif d'évaluer :

- La planification et la répartition des tâches,
- La gestion du contrôle de code source,
- L'analyse UML