

Titre du projet : Reproduction du jeu « Geometry Dash »

Contexte et objectifs du projet

[Vidéo présentation du jeu](#)

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.

- Programmation C# : Application des concepts de programmation orientée objet en C#.
- Conception Unity : Utilisation de Unity pour la gestion de physique, d'animation et d'interface utilisateur.

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

- Contrôle de Version (Git) : Suivi des modifications et gestion collaborative du code. Utilisation de branches pour les différentes fonctionnalités, gestion des versions et suivi des contributions.
- Planification (Trello & Gantt) : Organisation des tâches en sprints avec Trello et un suivi temporel via un diagramme de Gantt.
 - [Trello](#)
 - [Explications du 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.

Gestion de Projet

Modélisation UML : Analyse des besoins et structure du projet à travers des diagrammes de classes, d'activités, d'états et des cas d'utilisation. Éléments clés du projet Organisation et Planification de Projet.

- Diagramme de Classes : Définir les classes principales (ex. Joueur, Obstacles, Gestionnaire de Niveau ect..) et leurs relations.
- Diagramme d'Activités : Représenter les différentes étapes du gameplay (déplacement automatique, sauts, collisions ect..).
- Diagramme d'États : Visualiser les différents états du joueur (en déplacement, saut, collision, game over).
- Cas d'Utilisation : Identifier les fonctionnalités principales du jeu (ex. sauter, redémarrer, générer des obstacles ect..), Les différents menus du jeu, pour organiser le code en modules fonctionnels.

Développement Technique & Définition du besoin

Plusieurs éléments sont à prendre en compte

- Contrôles du Joueur : Script pour le déplacement automatique et le saut.
- Environnement et Obstacles : Génération d'obstacles dynamiques, gestion des collisions, type d'obstacles différents
- Accès à différents bonus affectant le joueur (supérieur à 2 bonus),
- UI et Feedback Visuel, Audio: Effets audios et visuels durant la partie, effets visuels de fin de partie avec animation de mort ou de réussite de niveau.

Interface

- Menu de démarrage (Jouer, Quitter, Editeur, Scores ..),
- 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, Optionnel : Édition des touches)

Générateur de niveau

- Import/Export des cartes au format fichiers (ex. json, xml) ★ Checkpoint 9 point sur 15
- Editeur de carte ★ Checkpoint 12 point sur 15

IA

- Format d'import export standard selon fichier d'annexe
- Création d'une IA capable de jouer au jeu en s'inspirant de la vidéo suivante :

 J'ai fait une IA qui apprend à jouer à Geometry Dash.

★ Checkpoint 15 point sur 15

Processus de Réalisation et Livrables Phases de DéveloppementPhase 1 : Planification et Modélisation

Créer un diagramme de Gantt pour organiser les étapes du projet. Utiliser Trello pour attribuer les tâches et planifier les sprints. Rédiger les diagrammes UML pour définir la structure du code et les interactions principales.

Phase 2 : Développement et Tests

Implémenter les fonctionnalités de base du gameplay. Utiliser Git pour le contrôle de version, avec des commits réguliers.

Phase 3 : Finalisation et Documentation

Rédiger une documentation décrivant la structure du projet, les diagrammes UML, et le processus de développement.

Livrables Finaux Code Source : Bien organisé et versionné, Un exécutable du jeu doit être disponible et fonctionnel. Utilisation de Git, modularité et lisibilité du code.

Trello et Gantt : Capture d'écran de Trello avec toutes les tâches, diagramme de Gantt final.

Diagrammes UML : Diagrammes de classes, d'activités, d'états et cas d'utilisation, sous forme de fichier PDF ou image.

Critères d'Évaluation

- Qualité de la Planification : Utilisation effective de Trello, cohérence du diagramme de Gantt.
- Respect de l'Analyse UML : Diagrammes corrects et cohérents avec le code final.
- Gameplay et UI : Fonctionnalités stables et expérience utilisateur satisfaisante.
- Présentation Orale : Récapitulatif du travail, des défis rencontrés, et des solutions.
Présentation du jeu.

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 17/05/2025 23h59 au plus tard. Tout sujet rendu après cette date ne sera pas évalué.
L'évaluation POO sera découpée en trois parties :

- 15 points attribués au projet
- 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

```

Y5      ###
Y4      vv
Y3
Y2      <#
Y1 S   ^   <#>           E
Y0 #####
   XXXXXXXXXXXXXXXXXXXXXXXX
   01234567891111111111222
   0123456789012

```

Il faut voir l'exemple ci-dessus comme un tableau en 2 dimensions (XY). Les indications de ligne (X) et colonne (Y) ne font pas partie du fichier d'échange. Vous trouverez un fichier utilisable pour vos tests ici :

- caractère espace : un espace vide
- S exemple position (X0;Y1) : le point de start
- E exemple position (X22;Y1) : la fin de la map
- # exemple position (X0;Y0) : un cube infranchissable
- ^ exemple position (X4;Y1) : un pique vers le haut
- v exemple position (X6;Y4) : un pique vers le bas
- < exemple position (X10;Y1) : un pique vers la gauche
- > exemple position (X12;Y1) : un pique vers la droite