Jeux et Multimédia - Projet de session

# Énoncé

Dans le cadre de ce travail, vous allez devoir développer un projet multimédia[[1]](#footnote-1) interactif qui peut être un jeu, une simulation multimédia ou une démonstration multimédia.

Le projet devra avoir les éléments ci-contre. Chacun de ces points sera développé plus bas.

* Menu initial
  + Démarrage
  + Instruction
  + Configuration
    - Ajustements indépendants des niveaux sonores de la musique et des effets sonores
* Cœur : Jeu, simulation ou démonstration
  + Interaction avec l’utilisateur
  + Graphisme et animation
  + Son
    - Touches raccourcies pour mettre en sourdine (*mute*)
    - Son d’ambiance pour la musique
    - Sons réactifs (effets sonores)
  + Algorithmes ou mécanismes de jeu
  + Changement de niveau
  + Données pour les geeks
    - Touche F12 pour afficher la ram et les images par seconde à l’écran en surimpression
* Scène de fin
  + Une scène indiquant la fin du projet
    - Il doit y avoir une fin perdante et une fin gagnante
  + Menu pour quitter ou revenir au menu principal ou redémarrer
* Mise en pause

# Descriptions des points

## Structure du projet

Votre dossier de projet sur GitHub devra avoir les éléments suivants :

* Collaboration avec mon compte « nbourre »
* Dossier « src » pour le projet Godot
* Fichier « readme.md » à la racine pour la présentation et documentation de votre projet

## Le menu initial

Le menu initial constitue la porte d’entrée de votre projet. Il doit permettre à l’utilisateur de **démarrer le jeu** ainsi que d’accéder à un **menu de paramètres** pour configurer divers éléments avant de jouer.

Parmi les options attendues dans ce menu, on retrouve notamment :

* **Le réglage du volume** de la musique et des effets sonores.
* **La configuration des touches** pour interagir avec le jeu.
* **Le choix du niveau de difficulté**, si applicable.
* **Les instructions** expliquant les mécaniques du jeu.

Ce menu joue un rôle essentiel dans l’introduction du projet. Il doit être clair, fonctionnel et offrir une première impression engageante à l’utilisateur.

## Le Cœur du jeu, de la simulation ou de la démonstration

Cette section représente l’élément central de votre projet, l’équivalent du plat principal dans un repas. Tout le reste du projet gravite autour de cette composante. C’est ici que vous devez démontrer votre créativité, vos compétences techniques et votre capacité à concevoir une expérience interactive cohérente.

Pour réussir cette partie, votre projet devra inclure les éléments suivants :

### Interaction

Le projet doit offrir une **interaction minimale avec l’utilisateur**, que ce soit par le clavier, la souris ou une manette de jeu. Il doit également être **compatible avec l’arcade sous Linux**.

Il faut prendre en considération les limitations de l’environnement.

### Changement de niveau ou de scène

Votre projet doit inclure un mécanisme permettant le **changement de niveau** ou de scène, selon la progression du joueur ou les événements du jeu. Ce changement peut être déclenché par :

* L’atteinte d’un objectif.
* L’interaction avec un élément du jeu.
* Une condition particulière (ex. : temps écoulé, score atteint, etc.).

Chaque niveau doit proposer une **variation significative** en termes de contenu, de difficulté ou d’environnement.

### Graphisme et animation

Un projet multimédia sans graphisme ni animation perd son caractère immersif. Vous devez intégrer des **éléments visuels animés** et prévoir des **réactions visibles** aux actions de l’utilisateur. Par exemple :

* Un personnage qui réagit lorsqu’il est touché par un ennemi.
* Une voiture qui entre en collision avec un mur.
* Une pièce de casse-tête qui se *snappe* à sa position.

Un **overlay** doit également être présent, affichant des informations comme le pointage, le nombre de vies, les forces ou les contrôles.

### Son

Le projet doit inclure un **minimum de son**, comprenant :

* Une musique de fond.
* Des effets sonores réactifs lors d’événements.
* Une option pour mettre le son en sourdine, accessible via le menu de configuration et une touche raccourcie pendant le jeu.

### Algorithmes et mécanismes de jeu

Vous devez intégrer **au moins deux algorithmes distincts**, développés **à partir des principes de base**, sans recourir à des bibliothèques externes. Par exemple :

* Génération de terrain ou de labyrinthe.
* Cinématique inverse.
* Intelligence artificielle.

Alternativement, vous pouvez utiliser des **mécanismes intégrés** à la plateforme de création (ex. : Godot), comme le *raycasting*, le *path following* ou le parallax. Dans ce cas, **le double des points** sera requis pour compenser l’absence de développement algorithmique.

### Données pour les geeks

Le projet doit inclure une **touche spéciale** permettant d’afficher des **données de débogage** utiles, telles que :

* Des boîtes de collision.
* Des vecteurs sous forme de flèches.
* Le taux de rafraîchissement (FPS).
* L’utilisation de la mémoire (RAM).

## Scène de fin

Votre projet doit inclure une scène de fin qui s’affiche lorsque le jeu ou la simulation se termine. Cette scène doit indiquer clairement à l’utilisateur s’il a atteint une **fin gagnante** ou une **fin perdante**, selon les objectifs du projet.

La scène de fin doit également offrir les options suivantes :

* Quitter le jeu.
* Revenir au menu principal.
* Redémarrer le jeu.

Pour faciliter la correction, une **touche spéciale** doit permettre d’accéder directement à cette scène, ainsi qu’aux éléments clés du jeu.

### Mise en pause

Votre projet devra permettre à l’utilisateur de mettre l’exécution sur pause via une touche que vous déciderez.

# Les algorithmes et ensemble de mécanismes de jeu

Comme indiqué dans l’énoncé, le projet devra inclure au moins deux algorithmes distincts et développés à partir de la base, i.e. sans l’utilisation d’une librairie.

Voici une liste non-exhaustive d’algorithmes que des étudiants ont développé dans le passé :

|  |  |
| --- | --- |
| Algorithme génétique | Générateur de labyrinthe/donjon |
| A\* | Réseau de neurones |
| Essaimage (Comportement d’agrégation) | Automate cellulaire |
| Générateur de piste de course | Jeu multi-joueur |
| Système L-Tree | Cinématique inverse |
| Génération procédurale | Agent autonome |
| Fog of war | Champ de vision |
| Algorithme de Prim | Système de Voxel |
| Patron de conception – State | Path following (Recherche de chemin) |
| Goal Oriented Action Planning (GOAP) | Patron de conception – Object pool |
|  |  |

Vous pouvez aussi utiliser vos propres idées d’algorithme, il suffira de me les faire approuver.

# Remise

La date de remise du projet sera pour la première semaine de décembre. Les cours subséquents seront gardés pour la finalisation, le montage de la présentation et le débogage.

La remise doit se faire via le formulaire [Git](https://forms.office.com/r/XSt0EmXz9D).

# Grille

|  |  |  |
| --- | --- | --- |
| **Critère** | **Points** | **Remarque** |
| Menu initial | 5 |  |
| Configuration d’élément dans le jeu | 5 |  |
| Instructions | 5 |  |
| Interaction dans le jeu | 5 |  |
| Élément d’animation | 5 |  |
| Réaction à la suite d’événements | 5 |  |
| Overlay | 5 |  |
| Son | 10 |  |
| Algorithme 1 | 10 | Mécanisme 1 et 2 |
| Algorithme 2 | 10 | Mécanisme 3 et 4 |
| Données de débogage | 10 | FPS, Mémoire, etc. |
| Scène de fin | 5 |  |
| Quitter, Retour au menu et redémarrer le jeu | 5 |  |
| Touche rapide | 5 |  |
| Ajustement de qualité | 10 |  |
| **Total** | **100** |  |

Handicap Processing : Si vous décidez de réaliser le projet en Processing, l’implémentation des nœuds de mécanisme de Godot peuvent être considérés comme les algorithmes 1 et 2. Par exemples, la collision, les animations, les tuiles, etc.

Une valeur de 10% du total est gardée pour la qualité générale du projet.

1. Pour alléger le texte, j’utiliserai le terme « jeu », mais celui-ci inclura tout projet multimédia. [↑](#footnote-ref-1)