



**Le projet**

Le projet consiste à créer des IA capables de conduire des véhicules.

Pour se faire, nous allons tout d’abord créer un simple jeu de course.

Ensuite, nous allons entraîner des IA sur ce jeu.

Environnement de développement : Unity.

Technologie ciblée : ml-agent (intelligence artificielle)

Notre jeu est une course de voiture 2D en point de vue “top down”.

Tous les joueurs débutent sur la ligne de départ au même moment.

La course se déroule sur un parcours aléatoire en fonction de la configuration fournie par l’utilisateur. (semi-procédural, full-procédural, nombre de tours, etc...)

Le gagnant est le premier qui atteint la fin de la course.

La course se termine lorsque tous les joueurs ont atteint la ligne d’arrivée.

Un classement avec le temps des joueurs est affiché à la fin de la course.

Dans le cas d’une course “infini”, il n’y a pas de gagnant.

La partie doit être stoppée manuellement par l’utilisateur.

**Règles du jeu**

Voici les règles :

Il ne faut pas toucher de mur.

Il est interdit de conduire en sens inverse.

Il est obligatoire de passer tous les checkpoints pour terminer la course.

Aucun sabotage.

**Fonctionnalités**

Conduite autonome : Les participants de la course sont tous des IA.

Dérapage : les véhicules sont difficiles à contrôler et requiert de la précision.

Course procédurale : la course est générée aléatoirement.

Checkpoints : chaque course est composé de points de sauvegarde qui permettent aux joueurs  
de réapparaître sur la piste en cas de collision avec un mur. (sauvegarder leur progression)

Ils servent aussi de condition pour terminer la course.

**Objectifs**

Objectifs nécessaires à la complétion du projet :

#1 1 tour sans toucher le mur (map test)(modifié)

#2 5 tour consécutifs sans toucher le mur (map test)

#3 1 tour sur deux map choisies aléatoirement

#4 1 tour sur une liste de 5 maps (ordre choisie aléatoirement)

Objectif optionnel :

Chaque participant entraîne un AI pour une confrontation sur la piste.

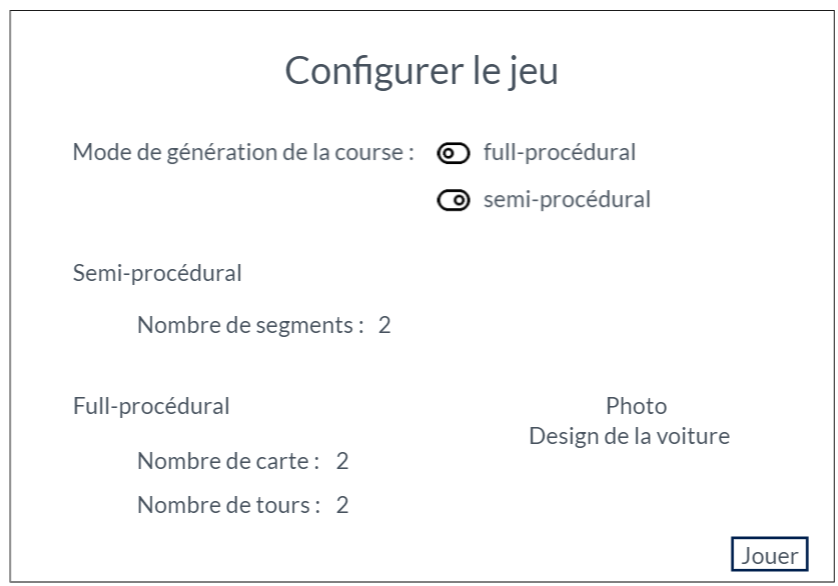
**Interfaces**

Ici vous insérez les dessins des fenêtres de votre logiciel.

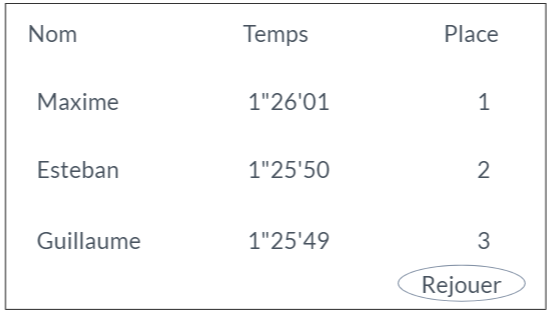
**Écran 1**

****

**Écran 2**



**Écran 3**

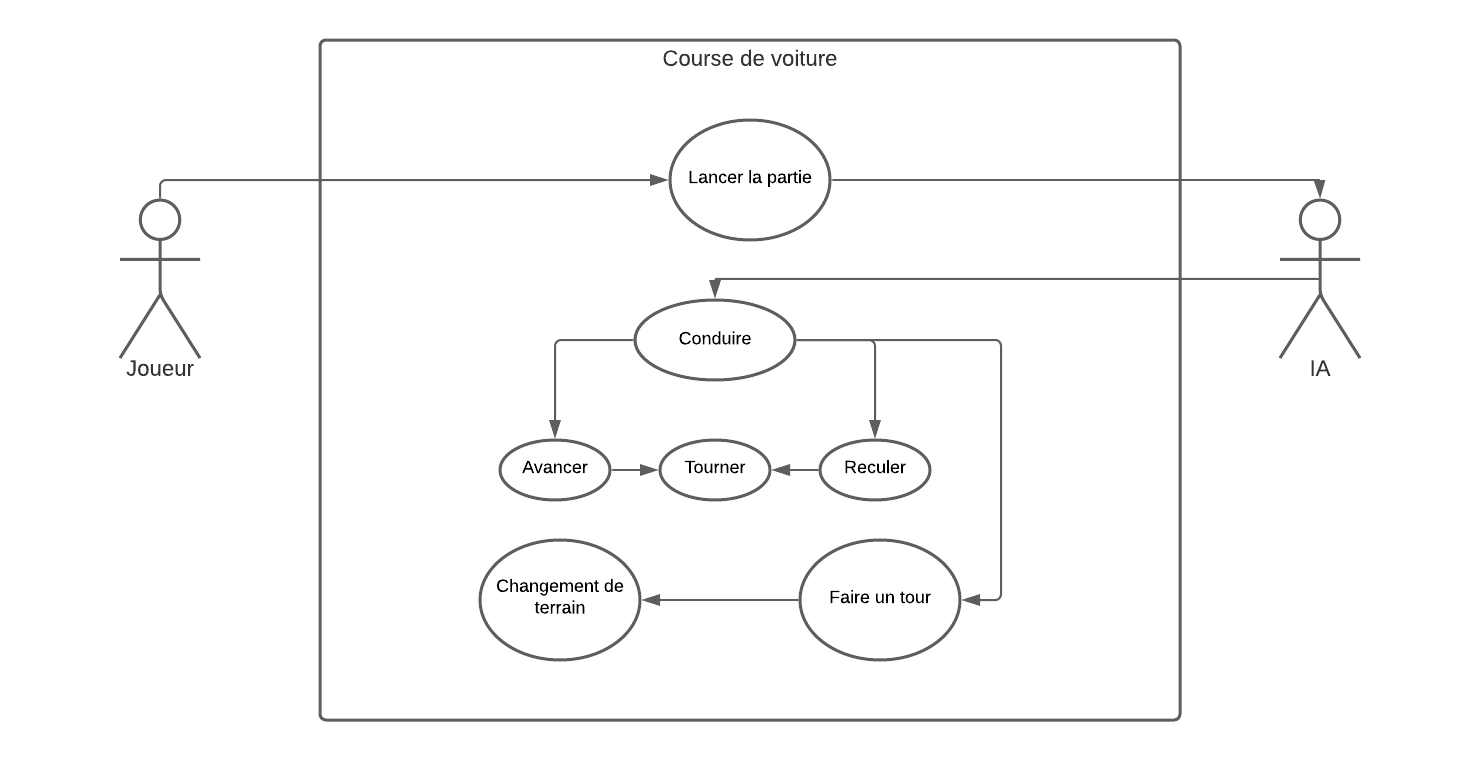


**Cas d’utilisation**

**Diagramme des cas d’utilisation**

Insérez une copie de votre diagramme UML de cas d’utilisation

Le diagramme de cas d’utilisation est une table des matières des fiches de cas d’utilisation présentées, vous avez un ovale pour une fiche. Si une fiche mentionne une autre fiche, c’est alors qu’il y a un lien entre les ovales.



**Fiche de cas d’utilisation A**

Ici vous résumez dans un langage fonctionnel les étapes de l’interaction avec l’utilisateur. D’une manière robotique on liste ce que l’utilisateur fait et ce que le système répond en terme d’interface utilisateur (bouton cliqué, texte affiché, écran affiché.

|  |  |  |
| --- | --- | --- |
|  |  | |
| **Résumé** | Lancement partie | |
| **Acteur principal** | Joueur/utilisateur | |
| **Intervenants & Rôles** | **Intervenant** | **Rôle** |
|  |  |
|  |  |
|  |  |
| **Préconditions** | Aucune | |
| **Postconditions** | Aucune | |
| **Déclencheur** | Bouton clique | |

|  |  |  |
| --- | --- | --- |
| **Scénario nominal** | **Étape** | **Action** |
|  | 1 | Cliquer sur le bouton jouer. |
| 2 | Configurer les options de la carte. |
| 3 | Une fois les options choisies, lancer le jeu avec le bouton “Jouer”. |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Scénarios alternatifs** | **Étape** | **Actions** |
|  | 1 | Cliquer sur le bouton jouer. |
| 2 | Laisser les configurations de bases. |
| 3 | Lancer le jeu directement (deuxième bouton jouer). |
|  |  |
|  |  |

**Fiche de cas d’utilisation B**

Ici vous résumez dans un langage fonctionnel les étapes de l’interaction avec l’utilisateur. D’une manière robotique on liste ce que l’utilisateur fait et ce que le système répond en terme d’interface utilisateur (bouton cliqué, texte affiché, écran affiché.

|  |  |  |
| --- | --- | --- |
|  |  | |
| **Résumé** | Changement de camera | |
| **Acteur principal** | Joueur/utilisateur | |
| **Intervenants & Rôles** | **Intervenant** | **Rôle** |
|  |  |
|  |  |
|  |  |
| **Préconditions** | Avoir lancé la partie. | |
| **Postconditions** | Aucune. | |
| **Déclencheur** | Appuie d’un bouton | |

|  |  |  |
| --- | --- | --- |
| **Scénario nominal** | **Étape** | **Action** |
|  | 1 | Appuyer sur un bouton |
| 2 | Changement de caméra |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Scénarios alternatifs** | **Étape** | **Actions** |
|  |  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Fiche de cas d’utilisation C**

Ici vous résumez dans un langage fonctionnel les étapes de l’interaction avec l’utilisateur. D’une manière robotique on liste ce que l’utilisateur fait et ce que le système répond en terme d’interface utilisateur (bouton cliqué, texte affiché, écran affiché.

|  |  |  |
| --- | --- | --- |
|  |  | |
| **Résumé** | Configuration de la map | |
| **Acteur principal** | Joueur/utilisateur | |
| **Intervenants & Rôles** | **Intervenant** | **Rôle** |
|  |  |
|  |  |
|  |  |
| **Préconditions** | Avoir lancé la partie | |
| **Postconditions** |  | |
| **Déclencheur** |  | |

|  |  |  |
| --- | --- | --- |
| **Scénario nominal** | **Étape** | **Action** |
|  | 1 | Choisir l’option full-procédurale |
| 2 | Choisir les options suivantes |
| 3 | Lancer la partie |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Scénarios alternatifs** | **Étape** | **Actions** |
|  | 1 | Choisir l’option semi-procédurale |
| 2 | Choisir le nombre de segment |
| 3 | Lancer la partie |
|  |  |
|  |  |

**Grille de test fonctionnel**

Ici vous documentez au moins 3 tests fonctionnels sous forme de grille à signer par le testeur

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Instructions | Résultat | Réussi | Échoué | Signature |
| Partie en mode semi-procédural. | 1. l’utilisateur appuie sur Jouer. 2. l’utilisateur choisit l’option semi-procédurale. 3. L’utilisateur clique sur lancer pour démarrer la partie. | La fenêtre Écran de jeu apparaît avec trois IA qui coursent sur **une série de circuits pré-créés**.  La partie se termine lorsque tous les IA ont atteint la ligne d’arrivée. |  |  |  |
| Partie en mode full-procédural. | 1. l’utilisateur appuie sur Jouer. 2. l’utilisateur choisit l’option full-procédurale. 3. L’utilisateur clique sur lancer pour démarrer la partie. | La fenêtre Écran de jeu apparaît avec trois IA qui coursent sur **une piste entièrement générée**.  La partie se termine lorsque tous les IA ont atteint la ligne d’arrivée. |  |  |  |
| Partie en mode infini | 1. l’utilisateur appuie sur Jouer. 2. l’utilisateur choisit l’option infini. 3. L’utilisateur clique sur lancer pour démarrer la partie. | La fenêtre Écran de jeu apparaît avec trois IA qui coursent.  **La partie ne se termine jamais**. Le joueur doit quitter manuellement à partir du menu pause. |  |  |  |

**Produits similaires**

Les jeux de voiture possédant une intelligence artificielle similaire à notre jeux sont les suivantes :

1. Need for Speed
2. Gran Turismo
3. Dirt
4. Wreckfest
5. Burnout
6. Crash Bandicoot : Racing
7. Mario Kart
8. Forza
9. Euro Truck Simulator

**Analyse comparative**

Nous avons opté pour Unity plutôt que Unreal Engine car nous voulons faire un jeu 2D. Unreal engine est plus utile pour un environnement 3D. Nous avons également plus d'expérience avec ce logiciel, il est donc plus intéressant pour l'équipe de pouvoir programmer le jeu le plus rapidement possible étant donné que la technologie evaluer n’a aucun lien avec notre création.

Nous avons décidé d’utiliser ML-agent car il est compatible avec Unity et possède énormément d'exemples afin de nous aider à la compréhension. De plus, ce paquet a été créé pour le développement de l’intelligence artificielle dans Unity. Le choix de cet outil était inévitable.

**Défi de l’équipe**

Le défi de l'équipe est de réussir à maîtriser l’intelligence artificielle par rapport à un jeu Unity. Etant donné qu'aucun de nous avons une expérience dans ce domaine. Celui-ci représente une réelle épreuve.  
Cependant Esteban va devoir apprendre à utiliser Unity. Ce qui lui donne un défi supplémentaire.

**Ressources**

Ici vous placez les liens vers les outils importants de la conception du projet : outils d’analyse fonctionnelle, dessin des écrans, prototypage, inspirations, jeux existants, etc.

**Liens vers les outils**

**Outil de prototypage d’écran**

https://marvelapp.com/

**Outil de diagrammes**

<https://lucid.app/>

**Liens vers les sources d’inspiration**

**Lien pour les idées d’interfaces**

<https://fr.wikipedia.org/wiki/Crash_Bash> (tres vieux jeu)

**Lien pour les ressources graphiques utilisée**

Voiture faite par Marie-Emilie Amsallem