

# Jeu

Collège de Bois-de-Boulogne

Élève 1, 1234567

Élève 2, 1234567

Élève 3, 1234567

Élève 4, 1234567

Élève 5, 1234567

**[MOTEUR DE JEU]**

# Trafic

## Résumé du jeu :

Le jeu est un jeu de plateforme à défilement latéral intégrant des éléments de collecte et de course contre la montre. Dans ce jeu, le joueur incarne Jean Richard, un petit employé d'Onion Corp, qui a pour objectif de se rendre à temps à son travail et d'ainsi éviter de se faire licencier.

Dans le jeu, le joueur aura donc pour objectif de se rendre à destination le plus rapidement possible dans le but de marquer le plus haut score. Pour se faire, il devra parcourir la ville et ses environs, sautant de plateformes en plateformes tout en évitant les obstacles sur son chemin.

## Caractéristiques techniques du jeu :

### Menus :

Trafic est un «platformer» aux allures cartoon. Les graphismes généraux du jeu se doivent d'être simples et colorés. L'esthétique du jeu ne tend pas vers un rendu réaliste. Les menus, quant à eux, doivent être clairs et ergonomiques. Il nous faut donc un moteur de jeu qui offre la possibilité de gérer une interface graphique dynamique avec des icônes 2D et 3D interactives.

### Rendu des objets 3D :

De nombreux éléments 3D du jeu utiliseront le «cel-shading» pour amplifier l'aspect cartoon. Il nous faudra donc un moteur de jeu capable de rendre en temps réel ce type de matériel. À tout le moins, le moteur devra être apte à rendre des «shaders» simples (lambert, blinn, phong), ainsi d'afficher des textures. Bref, notre jeu est principalement composé d'éléments de basse résolution; nos éléments 3D, ayant un nombre limité de faces et étant texturés avec des images, ne dépasseront pas les 512 pixels carré.

### Particules :

Le moteur que nous utiliserons devra avoir une option permettant le déclenchement de particules ou de «sprites». En effet, ceux-ci seront utilisés lors de diverses interactions entre le personnage et le décor.

### Animations :

Pour réaliser notre jeu, il nous faudra un engin capable de lire en temps réel des animations importés d'un logiciel 3D (Maya, MotionBuilder, etc.). Les animations ne comporteront que la géométrie reliée à des joints pré-animés. De plus, l'animation de chaque action devra être jouée par le moteur lors de son déclenchement et ce

dernier devra être en mesure de gérer des interpolations et des boucles d'animation en temps réel (course, marche, etc.)

### **Lumières et rendu :**

Le jeu utilisera un éclairage dynamique pour illuminer les niveaux du jeu. Ces lumières devront réagir dynamiquement par rapport au cel-shading appliqué sur le personnage et émettre des ombrages par rapport au joueur et son environnement.

### **Programmation :**

Le jeu ne comportera pas d'interactions qui requerront des algorithmes trop complexe. Le moteur devra nous permettre d'élaborer des actions simples en évitant de coder le plus possible.

### **Interaction avec les réseaux sociaux :**

Le moteur devra permettre de laisser libre-cours aux partages de certaines données, notamment, des scores.

### **Caractéristiques de l'engin :**

#### **Besoins :**

- L'engin supporte le toon shading.
- L'engin comporte un module qui interagit avec les réseaux sociaux.
- La communauté par rapport au moteur de jeu est assez développée et sérieuse.
- L'aide et le support du moteur de jeu est facile d'accès et pertinent.
- L'engin supporte les formats de la suite Autodesk (FBX,OBJ,MAX,MA,MB,etc.)
- L'interface est facile à comprendre.
- Le moteur est gratuit ou est disponible à moindre coût.
- L'exportation du jeu peut se faire sur PC, Android et IOS.
- L'engin supporte un éclairage dynamique.
- L'engin supporte différents types de langages de programmation.
- Le moteur de jeu peut être amélioré grâce à des plug-ins.
- L'engin supporte les lumières dynamiques.
- Création de GUI aisé.
- Un système de particule est intégré dans l'engin ou peut être rajouté.

#### **Engins :**

##### **Unity 3D :**

Plusieurs engins de jeu supportent les caractéristiques nécessaire à notre jeu. Par contre certains ont plusieurs avantages. Unity 3D a une communauté très développée et sérieuse. Le support est aussi facilement accessible grâce à un manuel d'utilisation disponible sur la page du support d'Unity 3D. Il y a aussi un

module complémentaire qui permet le partage de données sur Facebook. Son interface est ergonomique et facile à comprendre et les outils ressemblent beaucoup à ceux d'Autodesk Maya et il supporte les fichiers Maya. Unity offre une version gratuite de son logiciel, mais certaines fonctions ne sont pas disponibles. Toutefois, les fonctions verrouillées ne sont pas essentielles à notre jeu. En outre, la programmation peut être réalisée en différents types de langage de programmation, nous offrant plus de choix lorsque viendra le moment d'en choisir un avec lequel nous sommes confortables. De plus, Le système de particules est déjà intégré dans l'engin de jeu. Unity 3D ne permet pas la lecture de vidéos pour les cinématiques. En somme, un engin bien adapté à nos besoins.

### **Torque 3D :**

Torque est un logiciel qui semble solide au niveau 3D, mais le support entourant cet engin est moins présent et plus difficile à trouver. Par contre, le logiciel est fait de plusieurs modules qui peuvent être substitués au besoin, lui donnant une grande versatilité. La multitude de modules permet d'obtenir un engin parfaitement adapté aux besoins de chaque production. Torque intègre également une fonction qui permet de mettre à jour les textures de façon interactive et permet la publication du jeu sur la plateforme Steam. Bref, si Torque est totalement gratuit et modulable, la difficulté de trouver des tutoriels ou des informations pertinentes sur certains aspects techniques de cet engin nous ont fait préférer Unity 3D à Torque.

### **Conclusion :**

Notre choix s'est donc arrêté sur Unity 3D comme engin de jeu. Sa flexibilité, sa versatilité et sa facilité d'approche nous ayant convaincu que ce moteur est le mieux adapté, dans le cadre du cours d'interactivité dans le jeu, à nos besoins pour le type de jeu que nous souhaitons produire. En effet, ce dernier répond à nos exigences et la vaste quantité d'informations disponibles sur ce logiciel, que ce soit par le biais du manuel d'instruction officiel ou par biais de pairs au travers de forums, nous a poussé vers Unity.