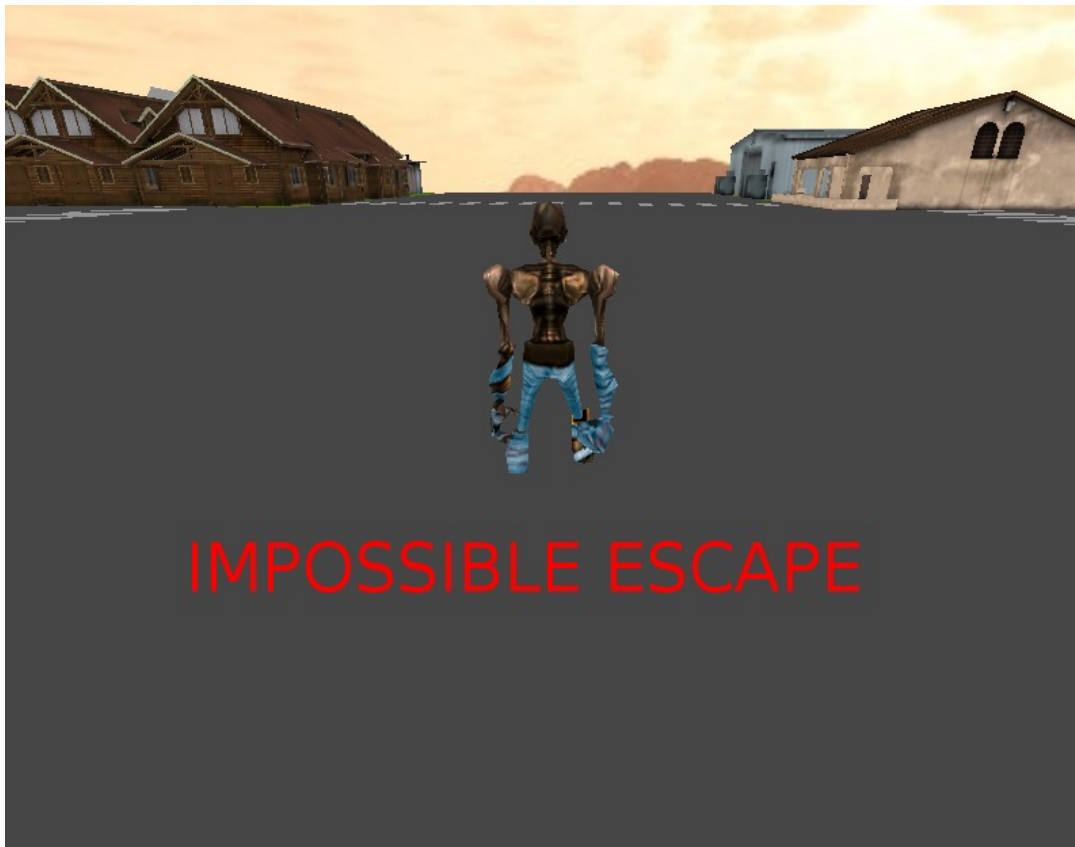


DI FOLCO Maxime
GIROT Charly

Rapport Jeux vidéo



1 - Principe du Jeu :

Notre jeu suit le principe d'un « escape game ». Le but du jeu est de réussir à s'échapper d'un monde magique. Pour cela, notre personnage, une momie, doit résoudre différentes énigmes afin d'avancer de monde en monde pour trouver la sortie.

Pour l'aider des *nains de jardins* lui permettront de voyager de décor en décor. Mais ces nains apparaîtront seulement lorsqu'une énigme a été résolue. A vous de résoudre les énigmes pour avancer dans l'histoire.

Le jeu commence dans une mystérieuse ville et transporte le joueur successivement sur une île pirate et à l'intérieur d'un club

1.1 - Commandes.

Vous pouvez déplacer le personnage grâce aux 4 flèches directionnelles. La souris vous permettra d'interagir avec les différentes fenêtres qui s'afficheront pendant le déroulement du jeu et de

résoudre une énigme. Les commandes sont rappelées si nécessaires.

Bonne chance

ATTENTION

La dernière partie de ce rapport comporte la solution aux énigmes.

2 – Gestion des Événements

2.1 - Déplacement du personnage

Le personnage est un Mesh3D représenté par un Animated Mesh. Deux caméras (CameraSceneNode) lui sont liées : une caméra principale qui suit ses déplacements et une caméra dédiée GUI pour un affichage différent. Pour le déplacement du personnage, une caméra est liée au personnage. Cette caméra principale reste fixe par rapport au personnage et le suit ainsi dans tous ses déplacements.

La gestion multi-touches (clavier) associée notamment aux déplacements du personnage est gérée dans le main et dans un Event Receiver.

L'Event Receiver contient un tableau de booléen « KeyIsDown » de la taille du nombre de touches qui stocke les touches enfoncées à tout instant. Les valeurs de ces booléens sont ensuite utilisées dans le main pour gérer les déplacements du personnage (avance, recule, tourne, ...). De plus une série de booléens permet également de gérer les animations du personnages lorsque celui ci est en mouvement ou non (Run, Stand, ...).

2.2 – Le GUI

L'interface utilisateur de ce jeu est composée de différentes fenêtres. Elles fournissent des informations nécessaires au déroulement du jeu et à son histoire. La classe myWindows permet la création et la gestion de ces fenêtres.

L'affichage des fenêtres s'effectue grâce à la fonction myWindows ::create-windows(enum nom_de_la_fenêtre) qui crée alors la fenêtre correspondante à l'enum passée en paramètre.

Les fonctions de créations de fenêtre spécifique (comme myWindows::create_window_begin()) gèrent des booléens qui nous permettent de situer notre personnage dans l'histoire. Une gestion correcte des booléens permet un affichage des fenêtres suivant l'avancée dans le jeu.

De plus l'affichage des fenêtres est très fortement lié à la classe EventReceiver, chargée de la gestion des événements claviers, gui et souris. C'est cette classe qui offre la possibilité de manipuler le contenu des fenêtres gui (SpinBox, Textes, Boutons, ...).

Finalement, une dernière fenêtre GUI indépendante de l'histoire est accessible par la touche ESC. Elle permet à l'utilisateur de mettre pause (le chrono cependant ne s'arrête pas à ce stade du développement), d'afficher à nouveau la dernière fenêtre d'informations utiles ou enfin de quitter le jeu sans l'avoir terminé.

3 – La scène et les éléments de décor

Lors du lancement du jeu, nous chargeons tous les décors et les autres éléments de notre jeu : les nains de jardins, les objets, les billboards, les cubes, les textures.

Le décor s'actualise et s'affiche en fonction de celui dans lequel le personnage doit se trouver. De plus les collisions sont gérées en actualisant le sélecteur de triangles et l'animateur de collision en fonction du décor affiché.

La procédure est la même pour tous les autres éléments de la scène, affichés lorsqu'ils sont nécessaires au déroulement de l'histoire.

3.1 - Les arches / nains de jardins et leur lumière (pour un repérage plus facile)

Les arches ont été créées en utilisant la combinaison de Mesh 3D (un nain de jardin) et des billboards. Les billboards liés au node de chaque Mesh3D sont utilisés pour afficher des sources de lumière autour du Mesh. Cela permet de rendre notre arche plus visible.

Un simple calcul de distance est utilisé pour gérer les collisions entre le personnage et les nains de jardins. Cette collision déclenche le changement de décor.

3.2 Les cubes

Grâce à la fonction addCubeSceneNode, nous avons pu créer des cubes. Nous avons défini ces cubes afin qu'il paraisse plat dans le jeu et utilisé dans le décor club pour afficher les chiffres aux murs et les peintures avec les énigmes.

3.3 – Le coffre

Le coffre est un Mesh 3D pour lequel la collision est gérée par un calcul de distance. Néanmoins, cette collision ne déclenche pas un changement de décor. Lors de la fermeture de la fenêtre la position du personnage est modifiée (décalage par rapport au coffre) pour sortir de la zone de collision et ne plus afficher la fenêtre.

4 – Solution aux énigmes

Énigme du coffre dans la jungle : La date est 11/06/1995

Énigme du club : Les numéros sont à l'étage sur des peintures. La solution est 907.

Trouver le dernier nain : Le dernier nain est au milieu d'un bloc de 4 bâtiments qui se situe derrière des bâtiments gris.

Énigme finale : La suite proposée est la suite de tous les nombres qui commencent par la lettre D (2,10,12,17,19), le prochain est donc 200.

5 – Améliorations possibles

- Gestion de la pause et du Chrono
- Enregistrement/Lecture du temps et d'un pseudo dans un fichier texte tiers
- Animation des nains de jardins
- Améliorer la gestion mémoire en ne chargeant les décors que lorsqu'ils sont nécessaires.
- Fluidité de la caméra lors des déplacements (interpolation du déplacement de la caméra vis à vis du déplacement du personnage)
- Fenêtre de chargement du jeu ? Aucune idée de comment mettre ça en place pour l'instant

- Faire en sorte que la caméra ne puisse pas traverser les murs (pour l'instant fixée brutalement à l'arrière du personnage et sans gestion des collisions, exemple : si le personnage est dos à un mur, éventualité que la caméra le traverse)
- Rendre les déplacements du personnage plus fluide
- Affichage du chrono en temps réel
- Ajout d'un multi écran pour jouer à plusieurs, avec éventuellement une histoire légèrement différente