LOST TOWN

PROJET D'OPENGL – C++

Kaddouri Chamseddine

Rousset Julien

Rozen Sudry Léa

11- 01-16

Imac 2 – OpenGL / C++

BIRI Venceslas

PICHARD Cyril

SOMMAIRE

PARTIE 1 : COTE DEVELOPPEUR

« CE PROJET VOUS PERMETTRA D'ABORDER LES NOTIONS CLES D'UNE PART DE LA PROGRAMMATION OBJET (VIA LE C++) ET D'AUTRE PART DU RENDU 3D (VIA OPENGL). L'ENJEU EST ICI DE CONSTRUIRE UNE ARCHITECTURE LOGICIELLE COHERENTE, EFFICACE ET QUI REPONDE AUX ENJEUX DE VISUALISATION DEMANDEE. LE THEME EST CELUI D'UNE VISITE VIRTUELLE - UN PARCOURS INTERACTIF - DANS UN UNIVERS 3D, DONT LE THEME EST LAISSE A VOTRE APPRECIATION. »

I. CONTRAINTES

a. CE QUI EST ATTENDU EST :

- √ De **pouvoir naviguer** (automatiquement) d'un "site" à l'autre
- √ Chaque site doit permettre de **naviguer vers d'autres sites**
- √ Les "sites" et les choix sont définis dans des **fichiers de configurations** chargées à l'initialisation
- √ Chaque "site" doit représenter un objet ou une **scène précise**. Cet objet ou cette scène doit avoir une apparence à chaque fois particulière, différente.
- √ L'utilisateur doit pouvoir **choisir la suite de son parcours.**

b. EN PLUS,

- √ **Navigation entre site** plus élaborée (fondu enchaînés, vrai déplacement le long de courbe spline)
- √ **Interaction** avec un objet
- ✖ **Modèle massifs** (+100 000 triangles)
- ✖ **Modélisation par points**

c. A METTRE EN VALEUR

- √ Respect de **spécifications**
- √ **Robustesse** du code (résiste aux crashs, gestion de la mémoire)
- ~ **Originalité** et diversité des rendus graphiques (shaders)
- √ **Lisibilité** du code
- √ **Intérêt** et originalité du jeu

II. DESCRIPTION DES CLASSES, INTERETS ET PROBLEMES RENCONTRES

A. MODEL

C'est une classe avec un chemin vers un objet de modélisation (fichier.obj). Le model contient aussi un vecteur de texture, celle de l'objet. Cet objet est représenté par ses meshs. (voir classe Mesh). Il se modélise grâce à ASSIMP. ASSIMP est une librairie qui possède la classe *IMPORTER*. (Voir class Importer)

Il contient aussi un vecteur de matrice 4 (mat4). Chacun mat4 donne une position, échelle et rotation ce qui permet d'afficher le modèle à plusieurs endroits pour un seul chargement.

Grâce à cette logique, on diminue l'espace de stockage dans la mémoire utilisé. On optimise donc le chargement de la scène.

Pour un modèle, on diminue de 'n' fois le temps de chargement (avec n : le nombre d'apparition du modèle dans la scène).

PROBLEMES RENCONTRES :

Les models étant nombreux, il était ennuyeux d'allouer et de désallouer les espaces utilisés par les models. Ayant plusieurs scènes qui peuvent s'afficher plusieurs fois, un modèle chargé pour la 1ere scène doit pouvoir être désalloué lorsqu'on passe à une autre scène, et être réalloué lorsqu'on revient à la 1ere scène.

C'est pourquoi nous avons créé une map<int, unique_ptr<Model>> pour chaque scène. Les unique_ptr sont des pointeurs uniques qui, lorsqu'on les supprime, désallouent automatiquement les objets pointés, ici, les models.

Ainsi, lorsqu'on passe d'une scène à une autre, on désalloue les models de la 1ere scène et on alloue les models de la seconde. Ceci, afin d'alléger la consommation mémoire utilisée par les objets 3D (très lourds).

B. IMPORTER

On donne à importer le chemin de l'objet, ses meshs et ses textures (le vector de textures). Il va se charger de dessiner les model à partir des données textuelles du fichier de modélisation.

Pour l'importation de modèle 3D avec Assimp, nous avons suivi le tutoriel de learnopengl.org. Nous avons dû réorganiser notre arborescence de manière à ce que cela puisse être compatible avec le résultat du tutoriel.

PROBLEMES RENCONTRES :

L' « *Importer* » d'ASSIMP peut importer des fichiers de modélisation avec différentes extensions (normalement). Cependant, on n'arrive pas à importer des fichiers autres que « .obj ». On n'arrive pas à faire « .fbx », « .dae ». Cette contrainte nous a limités énormément pour la création de notre scène. Nous avons dû arranger notre scénario en fonction de ce que nous pouvions trouver.

C. MESH

Lors de l'utilisation de la librairie Assimp, les modèles ne peuvent pas être gérer par OpenGL. Il est essentiel de pouvoir stocker les modèles chargés dans un dossier qu'OpenGL peut utiliser par derrière. La classe Mesh transforme les données pour rendre un objet utilisable. Cette classe comprend un ensemble de sommets où chacun d'eux contiennent un vecteur position, un vecteur normal et un vecteur de coordonnées de texture. Il contient aussi des indices pour les données (de dessin et textures)

D. SHADER

Créer des shaders (avec vertex et fragment).
Compiler et envoyer les variables uniformes.

E. FREEFLY

Création de la caméra et de sa position par rapport à la souris (angle) et au touche du clavier (position dans l'espace)

PROBLEMES RENCONTRES :

- Difficulté à reprendre les matrices de transformations vues en cours, surtout avec la caméra freefly.

➔ On a suivi un tutoriel sur internet (toujours sur learnopengl.com) pour créer une caméra freefly correcte. Puis, on y a jouté les déplacements. Ensuite on a modifié la caméra du tutoriel pour qu'elle respecte notre GamePlay (camera au sol, saut, se déplace dans l'axe de la caméra mais reste sur le plan (x ; y).

F. MOUSE

Gestion de la souris avec les coordonnées. Elle permet de garder en mémoire ses coordonnées. On pourra ainsi s'en servir pour l'utilisation de la caméra ou de la navigation du menu externe et interne au jeu.

G. MUSIC

Permet de charger une musique et la lancer que ce soit un son d'ambiance ou un « chunck » à un instant t suivant une action réalisée.
Il a fallu installer l'extension *SDL_mixer*.

PROBLEMES RENCONTRES :

- Téléchargement de l'extension. Les fonctions ne fonctionnaient pas. Il a fallu désinstaller puis réinstaller la *SDL* vu que plusieurs extension ont été installé sans succès et ont perturbé le fonctionnement de la *SDL*.
- Difficulté à prendre en main la bibliothèque (gestion du son via les canaux).
- (Pour la musique) SDL_mixer / (Pour les tests textures) SDL_image : Nous avions des références indéfinies et nous avons mis énormément de temps à comprendre ce que cela impliquait au niveau du cmake (ajout des définitions –lSDL_mixer, -lSDL_image, ainsi que les ajouts de librairies, etc…)

H. CHARGEMENT D'UNE SCENE DEPUIS UN FICHIER, MODEL

On peut, à partir du fichier (models.txt), créer le modèle et donc la scène. Chaque modèle est intégré à une map (en tant que pointeur unique). On utilise cette map pour dessiner tous les modèles (drawModele). Cette fonction appelle pour chaque modèle la fonction draw.

Exemple :

```
# Template :
# first line : Model filepath
# other lines : tx ty tz sx sy sz rx ry rz rdegré
Model ./../../../files/assets/models/stormtrooper/Stormtrooper.obj
10.0f -3.0f -20.0f 0.9f 0.9f 0.9f 0.f 0.f 0.f
12.0f -3.0f -17.0f 0.9f 0.9f 0.9f 0.f 0.f 0.f
Fin
```

Ligne commençant par # = commentaire
Ligne commençant par Model = chemin du modèle
Ligne commençant par chiffre = vector de float

PROBLEMES RENCONTRES :

- Problème de compatibilité
- Créer un fichier qui charge le model et ses transformations

I. IHM

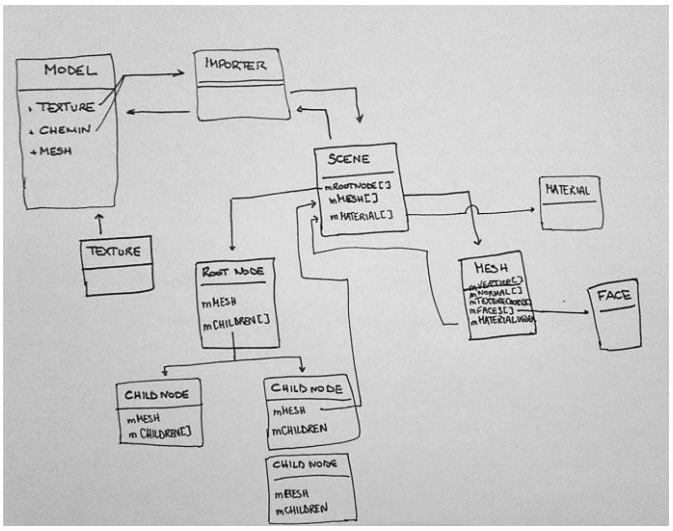


Diagramme de classe approximatif (ASSIMP)

Nous avons choisi un design pattern de type expert où le modèle est l'expert car c'est lui qui a le plus d'information pour les méthodes de notre programme comme « creeModele » « afficheModele » « trouverTexture ».

PROBLEMES RENCONTRES :

- Comprendre la librairie ASSIMP et les méthodes pour récupérer les éléments souhaités.

J. ARCHITECTURE DES DOSSIERS

Pour «installer et lancer le projet » :

- Installer et fait fonctionner le programme avec ./install at the directory root
- Lancer le programme avec ./launch

Pour le développeur :

- Actualiser le « build » en faisant ./update
- Actualiser le “make” avec ./cmake_update

Comment le système fonctionne :

- Toutes les commandes sont à la racine et le dossier “files” aussi.
- Dans ce dossier, il y a le dossier *BUILD* qui ne doit pas être “touché” car il est créé par la fonction d’installation et fonctionne avec “launch”.
- L’ensemble des « hpp » et « cpp » sont créés par nous (développeur) dans le dossier *ENGINE*. Il est possible d’y ajouter des fichiers. Afin de respecter la structure, il faut inclure dans le dossier *ENGINE* “engine/nom_du_fichier.hpp ». Le cmake fait le reste.
- Le main.cpp est dans le dossier *GAME*.
- Les shaders sont dans le dossier *SHADERS*.
- Tout les models 3D sont ranges dans le dossier *ASSETS* ainsi que les textures. L’ensemble des “images” peuvent y être stockées.
- Glimac et Third-party sont des dossiers composes de librairies que nous utilisons (Assimp, glm, gtest and Glimac).

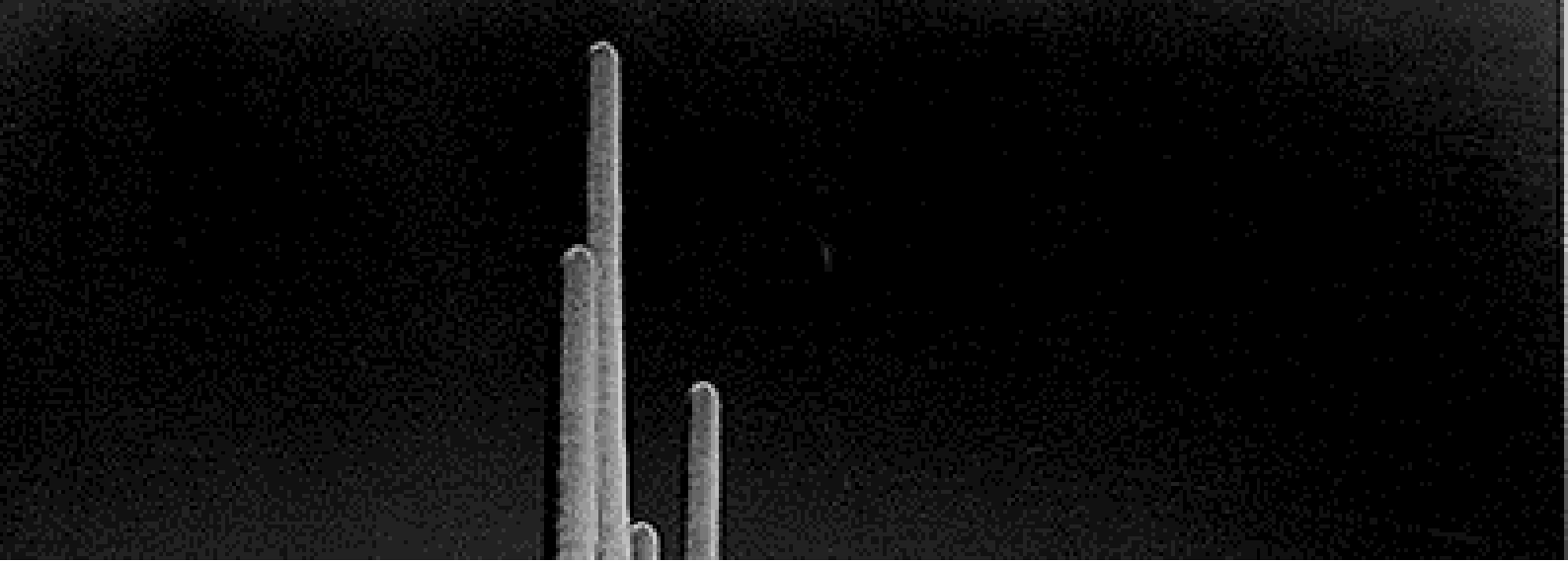
PROBLEMES RENCONTRES :

Nous ne savions pas réellement comment se structure un projet de cette envergure. Aucun de nous trois avait une expérience en création de jeu d’image de synthèse.

L’aide de certains imac3 et imac2 nous a permis de mieux nous rendre compte de l’organisation des dossiers et fichiers. Nous avons pu avoir une idée plus claire de comment nous devions organiser notre projet.

III. QUESTION EN SUSPEND

- Faire une source de lumière (l'intégrer)
- On aurait bien aimé pouvoir mettre en place les Deisgn Pattern de Stratégie dans le menu (effet au click en fonction de là où est le curseur).
- Gérer la gestion du fichier audio avec n'importe quel format. Pour le moment, seul le « .wav » peut être lu.



IV. APPORTS DU GROUPE

- GIT (appréhension, gestion, approfondissement des connaissances)
- Travailler en groupe, déléguer
- Gérer les Unique_ptr
- Gérer les Map
- Comprendre et utiliser des modèles/bibliothèque prédéfinis (Mesh, Importer, ASSIMP)
- Trouver des solutions aux problèmes rencontrés
- Gérer le chargement depuis un fichier (moteur)
- Trouver et implémenter une architecture pour un projet de synthèse d'image
- Cerner l'organisation et les étapes d'un projet d'image de synthèse
- Appréhension, utilisation, approfondissement des connaissances du C++ (location, gestion de mémoire)

PARTIE 2 : COTE GRAPHIQUE

I. HISTOIRE

a. MENU EXTERNE

Le jeu est exécuté et on arrive sur le menu principal.

On décide de :

- Jouer
- Mettre ou enlever le son
- Voir les crédits
- Quitter

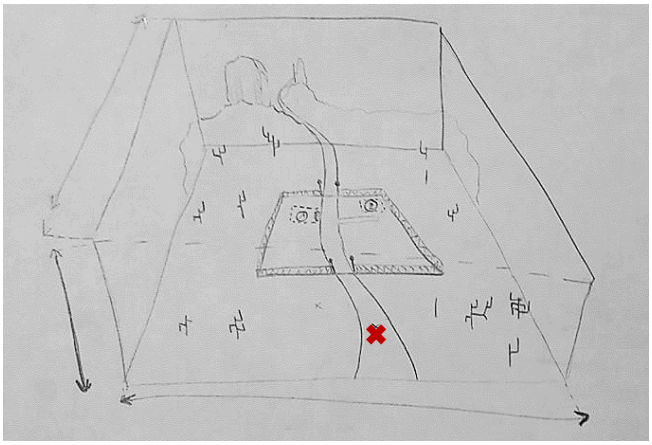
Si on décide de jouer, il est possible de voir une aide avec l'utilisation du clavier et de la souris dans le jeu.

b. LE JEU

Avant d'arriver sur le jeu, une image de présentation de la mission est présentée.

> *« Bonjour à la personne qui retrouvera ce message. Je suis Colin Winston et j'espère que vous réussirez à savoir pourquoi cette ville n'est plus habitable ! Je serais heureux de pouvoir revenir vivre ici. Le maire de LOST TOWN est vraiment tordu ! Faites attention à vous. »*

Puis, vous vous retrouvez à l'extérieur du village et vous allez commencer à rentrer dedans pour le visiter.



Modèle du village (skybox)

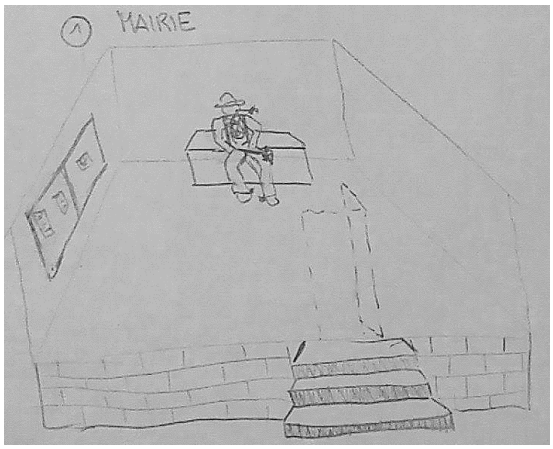
Après être rentré dans le village, un message s'affiche :

> *« Mmh, Hi ! Bienvenu à LostTown mon ami ! Je suis le maire du village. … Je suis intrigué « Que fais-tu là ? » Ici, il n'y a rien et personne. L'ensemble des commerces sont fermés d'ailleurs … navré. Vient me voir dans mon bureau, il y a un peu d'eau si tu le souhaites. »*

On se dirige vers la mairie et un panneau avec des informations sont affichés. On remarque un papier qui saute aux yeux où y est inscrit :

> *« ON A SOIF ! ON A SOIF ! ON NE VEUT PAS MOURIR ICI ! RENDEZ-NOUS NOTRE EAU ! ».*

Une fois rentrée dans la mairie, on remarque le shérif dans la pièce devant son bureau nous attendant.



Modèle de la marie + maire (2e scène)

Un message va s'afficher

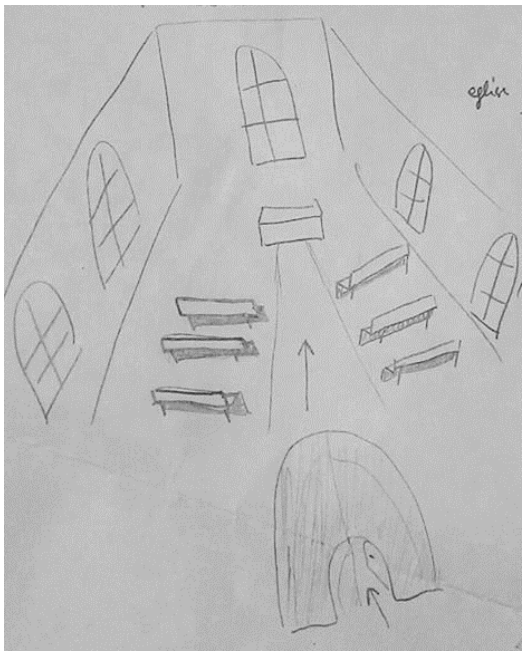
> *"Il n'y a plus d'eau dans la ville, les gens ont paniquer.. Je suis désolé... Je ne voulais pas ça, je voulais juste un peu d'argent, et il m'en avait promis... Je ne peux plus vivre avec ça sur la conscience... L'argent est désormais entre les mains de Dieu. Et la clé repose dans la 7e ressource d'eau."*

Le joueur doit alors comprendre que :

- « les mains de Dieu » fait référence à l'église
- « la clé repose dans la 7e ressource d'eau » fait référence à la cachette de la clé qui se trouve dans le 7e cactus éloigné de la ville.

En effet, le maire ne voulait plus donner son argent à la communauté et a donc arrêté de payer l'eau. Il l'a caché dans un coffre fermé à clé mais ne se souvient plus où. Le seul indice étant son énigme. Les villageois ont dû quitter la ville par force et le maire se retrouve sans rien (ni villageois, ni argent … et bientôt ni eau). Ayant peur que le joueur vol son argent en le retrouvant avant lui, il décide de le droguer.

Avant que le joueur parte de la mairie à la quête de cette clé, il se verra offrir un verre d'eau … empoisonné.
Il se retrouvera alors à la position du début du jeu (la transition se fait par un noir). La suite du jeu aura une vision brouillée par moment par des « noirs » ou des « flou ».
Un compte à rebours sera alors enclenché pour qu'il réussisse à trouver l'argent caché pour obtenir de l'eau … le simple remède qui le sauvera.
Une fois qu'il aura retrouvé la clef dans la forêt de cactus, il devra rejoindre l'église et retrouver le coffre.



Modèle de l'église + hôtel + banc (3e scène)

Le joueur devra aller à l'hôtel où il pourra interagir et déplacer involontairement sa position. En dessous apparaitra alors le coffre rempli d'argent.

FIN. (Texte indiquant que vous vous êtes sauvé la vie en plus d'avoir ramené la vie dans la ville et donc ses habitants. La nouvelle banque portera désormais le nom de votre personnage en honneur à votre geste envers Lost Town).

c. Menu interne

Si on presse la touche « échappe », le menu interne au jeu s'affiche.

On a la possibilité de :

- Revoir l'utilisation des touches du clavier.
- Mettre ou enlever le son
- Quitter et sauvegarder
- Retourner au jeu qui s'est mis en pause

II. CAPTURE D'ECRAN ET IMAGES INTEGRES

a. MENU EXTERNE

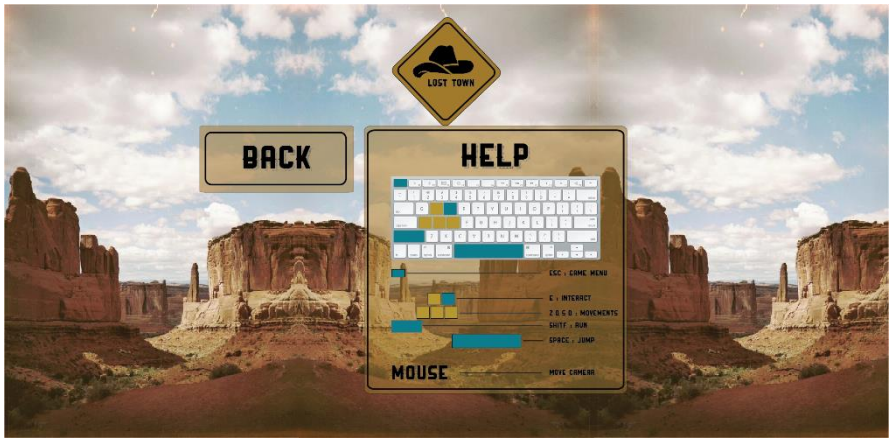
MAIN MENU >

MAIN MENU >EXIT

MAIN MENU > CREDITS

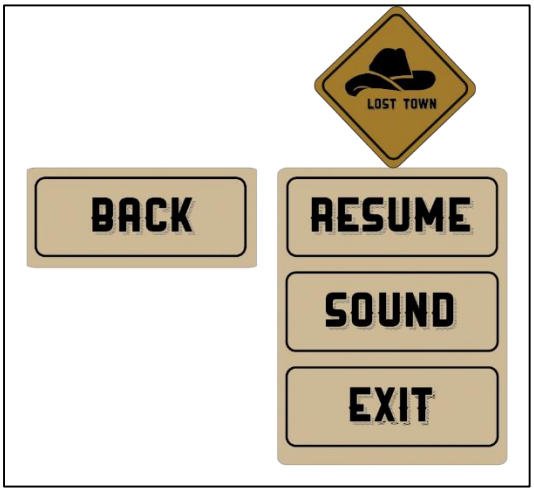
Main Menu >Sound

Main Menu > Play

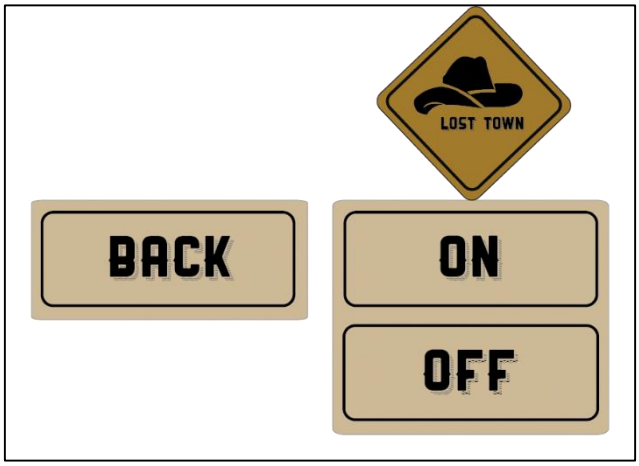


Main Menu > Play > Resume

b. MENU INTERNE

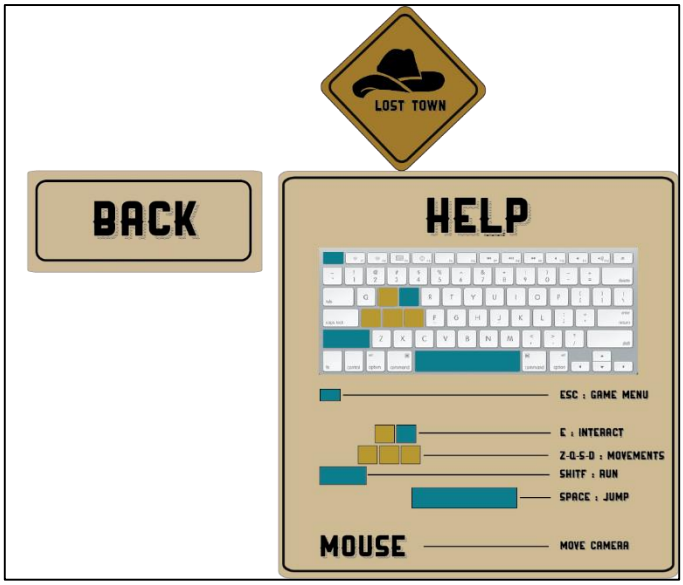


INTERN GAME MENU >



MAIN MENU > SOUND >

MAIN MENU > EXIT >



MAIN MENU > RESUME >

c. MODELE A CHARGER

- Une ville
  - Skybox (ciel + montagne grand canyon dans le fond)
  - Des montagnes pour délimiter le champ d'action

- Une église (extérieur, intérieur)
  - Des bancs
  - Un hôtel
  - Un lustre
  - Des placards

- Une mairie (intérieur, extérieur)
  - Un bureau
  - Un panneau d'affichage de prospectus
  - Un Cowboy, maire de la ville
  - Un verre d'eau

- Cactus (plusieurs modèles)
- Une clef
- Une porte pour entrer dans les deux maisons

d. MUSIQUE

https://vimeo.com/128877443 : Dry Light de Xavier Chassaing

PARTIE 3 : ANNEXE

1. GESTION CLAVIER