

**Dossier d’analyse**

*DIELMAN Geoffrey*

Epreuve intégrée réalisée pour l’

****

Présentation du projet

Le but de ce projet est de reproduire en VR une démo de base du jeu, si célèbre en son temps : **Day Of The Tentacule**. Basé naguère sur le système du ***point and click****,* le gameplay proposé tirerait énormément de bénéfice des progrès technologiques de notre époque et particulièrement de ceux réalisés dans le domaine de la réalité virtuelle.

Un petit bout d’histoire

Alors que ***Bernard***, ***Huagi*** et ***Laverne***, les trois protagonistes de l’histoire, sont chez eux et vaquent à des occupations aussi absurdes qu’inutiles, un télégramme arrive de la part d’un vieil ami de Bernard: le ***Docteur Fred***. Ce dernier sollicite leur aide afin d’échapper à une catastrophe provoquée par l’un des animaux de compagnie originaux qu’il a génétiquement créé: la ***Tentacule Pourpre***. Cette dernière serait devenue folle suite à l’absorption d’un produit chimique et aurait le projet de dominer le monde.

Une fois arrivés au manoir du Docteur Fred, Bernard, Huagui et Laverne le trouvent ligoté. Ce dernier leur apprend qu’il est désormais trop tard pour empêcher la Tentacule Pourpre de mettre en œuvre son terrible projet. Si on veut l’arrêter, il faudra donc intervenir HIER!

Notre jeu commence au moment où nos trois Héros se réveillent après un accident provoqué par les ***Chrono-WC***, une machine à voyager dans le temps inventée par le savant fou qui les a envoyés chacun dans une époque différente.

But du jeu

Dans une version complète du jeu, le but serait bien évidemment de trouver le moyen d’empêcher la Tentacule Pourpre de dominer le monde. Dans le cadre de cette démo, il sera simplement demandé au joueur de parcourir les deux premières pièces du manoir auxquelles il a accès pour le moment et de découvrir l’entrée du laboratoire secret du Docteur Fred.

Mécanique de Jeu

En explorant le manoir, le joueur trouvera une quantité d’objets qui lui seront utiles durant sa quête. Il devra se servir judicieusement de chacun d’entre eux afin d’évoluer dans le jeu pour atteindre son objectif. Trouver la clé correspondant à la porte close, utiliser les allumettes sur la cheminée afin de réchauffer George Washington ou encore coiffer une momie de spaghettis pour remporter le prix de la plus belle coiffure du concours de beauté: autant d’actions qui promettent d’être aussi intéressantes à implémenter qu’à vivre en Réalité Virtuelle.

En effet, chacune de ces actions pourra être réalisée de façon plus interactive que dans l’ancien système via les implémentations d’interactions que je décrirai plus loin dans le cadre de l’analyse des cas d’utilisations.

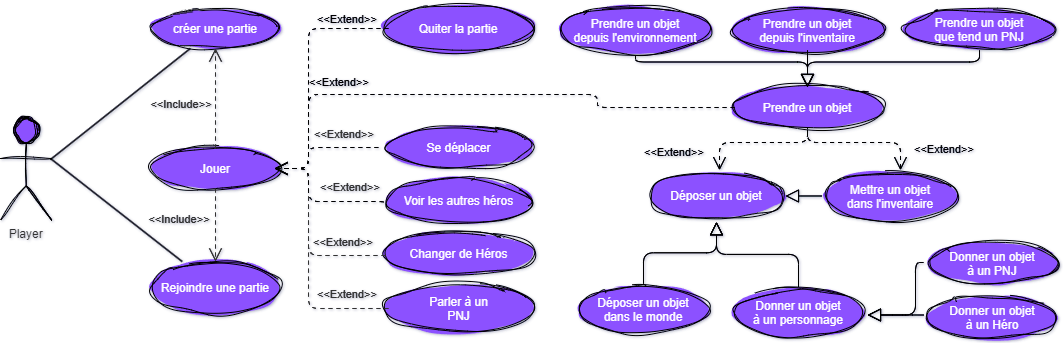
Fonctionnalités

Contrairement au jeu de base, cette version sera solo et multijoueur. En arrivant, les joueurs auront la possibilité de créer une partie ou d’en rejoindre une existante.

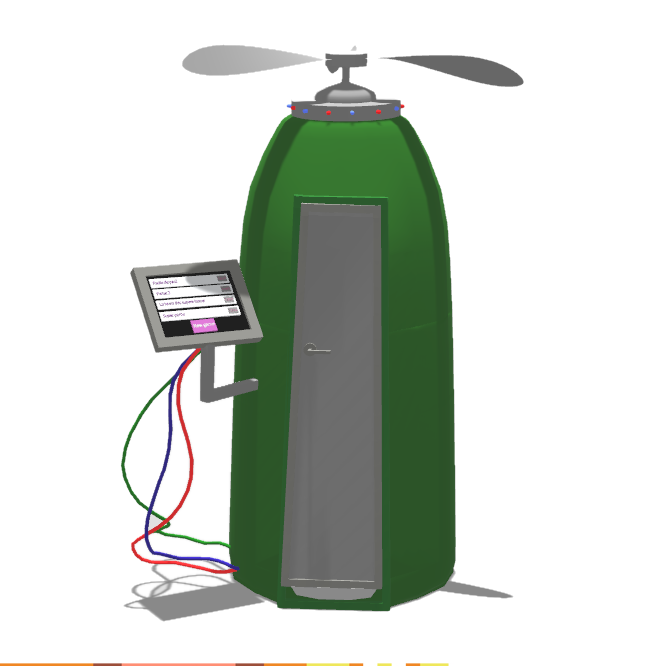
Lorsqu’un premier joueur entre dans une partie, il pourra incarner l’un des trois Héros disponibles: Bernard, Huagui ou Laverne. Il aura la possibilité de passer d’un Héros à l’autre à tout moment. Si un deuxième joueur rejoint la partie, il pourra choisir l’un des deux Héros encore disponibles lors de sa connexion et ainsi de suite. Les Héros vivront simultanément l’aventure dans leur époque respective mais pourront néanmoins s’entraider en s’envoyant des objets utiles à travers le temps via les Chrono-WC.

Use Cases

Voici une analyse détaillée des actions possibles au cours d’une partie de ***Dott-VR***.



* Créer une partie

Lorsqu’il lance le jeu, le joueur arrive dans le menu principal. Ce dernier est représenté par les Chrono-WC et leur tableau de commande. Sur l’écran de commande s’affiche une liste de toutes les parties disponibles sur le serveur et un bouton permettant de générer une nouvelle partie. Si tel est son choix, elle sera créée sur le serveur avec un nom auto-généré.

* Rejoindre une partie

Si le joueur sélectionne une partie dans la liste et appuie sur le bouton *Load*, il lance cette partie. Une fenêtre apparaît pour lui demander lequel des Héros encore disponibles il souhaite incarner.

* Quitter une partie

Quand il est en jeu, le joueur pourra revenir à tout moment au menu principal à l’aide de la touche *Menu* de la manette de gauche. Il pourra alors se servir de la poignée de la porte des Chrono-WC à l’aide de la touche *Grip* afin de sortir du jeu.

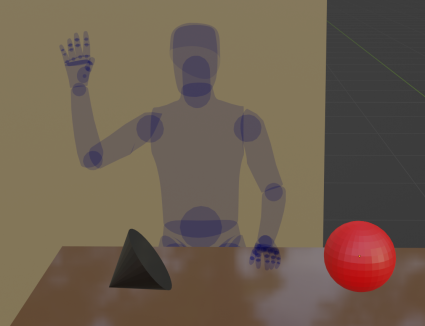
* Jouer

Une fois dans la partie, le joueur se retrouve dans une pièce du manoir dont le décor variera selon l’époque dans laquelle il se trouve.

Il y aura dans chaque époque divers personnages non-joueurs (PNJ) et des objets avec lesquels il pourra interagir, comme nous le verrons plus loin.

* Se déplacer

Le joueur peut se déplacer en utilisant le *Stick Analogique* de la manette de gauche ou en se déplaçant physiquement.

* Voir les autres Héros

Comme les Héros sont dans des époques différentes, ils ne devraient théoriquement pas pouvoir se voir. Cependant, j’ai jugé utile de faire une petite adaptation scénaristique afin de leur donner cette possibilité. Le passage dans les Chrono-WC ayant modifié leurs capacités sensorielles, les Héros peuvent voir leurs camarades à travers le temps sous forme de fantôme.

Étant donné que la maison reste la même à travers les époques, ceci ne pose aucun problème logique. Cela dit, un joueur verra ses amis mais pas les objets qui les entourent ni même ceux qu’ils tiennent en main.

* Prendre un objet

**- Depuis l’environnement**

Quand un objet est préhensile dans l’environnement, il est mis en surbrillance quand le joueur tend la main vers lui. S’il appuie sur le bouton de *Grip* de la main qu’il tend, le joueur prend alors l’objet en main. Le bouton de préhension sera à contact fugitif, c’est-à-dire activé uniquement tant que l’on appuie dessus.

**- Depuis l’inventaire**

Quand le joueur appuie sur un bracelet virtuel situé sur son bras gauche, il ouvre un écran holographique où figure son inventaire. Il peut y sélectionner un objet qui se matérialise alors au-dessus de l’écran. Si le joueur s’en saisit, l’objet prend place dans sa main. Il peut alors exécuter toutes les actions réalisables avec un objet.

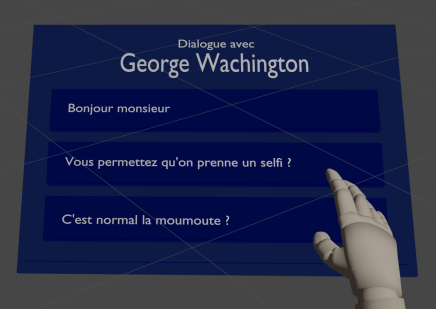
On remarque également les icônes de ses camarades qui peuvent servir pour leur envoyer des objets ou pour changer de Héros (voir la définition des *Use Cases* “Changer de Héros”**, p.X,** et “Donner un objet à un Héros”**, p.Y**).

**- Que tend un PNJ**

Suite à certains dialogues ou certaines interactions, les PNJ peuvent tendre la main pour donner un objet au joueur. Cet objet est préhensile et se comporte exactement comme s’il était dans l’environnement.

* Changer de Héros

**Blablabla**

* Parler à un PNJ

Comme pour la prise d’objet, le joueur peut tendre la main vers l’un des PNJ qui sera alors mis en surbrillance. Il peut alors appuyer sur le bouton d’interaction *A* pour afficher le menu de dialogue à coté de sa main droite.

Il peut choisir l’une des possibilités de dialogue en appuyant avec son doigt virtuel. La phrase qu’il a choisie sera alors dite par son personnage qui recevra une réponse (vocale elle aussi) de son interlocuteur.

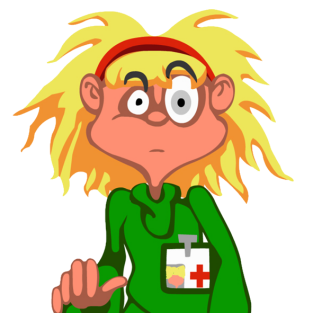
* Déposer un objet dans le monde

Le bouton de préhension étant à contact fugitif, le joueur tient l’objet tant que ce bouton est enfoncé. Il peut donc le déposer dans le monde en lâchant simplement le bouton de *Grip.*

* Mettre un objet dans l’inventaire

Quand il a un objet en main, deux possibilité s’offrent au joueur afin de le mettre dans l’inventaire.

**- Stockage rapide**

Si le joueur appuie sur le bouton *B*, il stocke immédiatement l’objet qui disparaît de sa main et prend place dans l’inventaire.

**- Déposer dans l’inventaire**

Si l’inventaire du joueur est ouvert alors qu’il a l’objet en main, il peut le déposer sur l’une des cases vides. Lorsqu’il approche l’objet d’une case, via la technologie des *Sockets*, la case de réception se met en surbrillance. Cela indique au joueur que l’objet peut être placé à cet endroit. Lorsque le bouton de *Grip* est lâché, l’objet disparaît directement de la main et est placé dans l’inventaire.

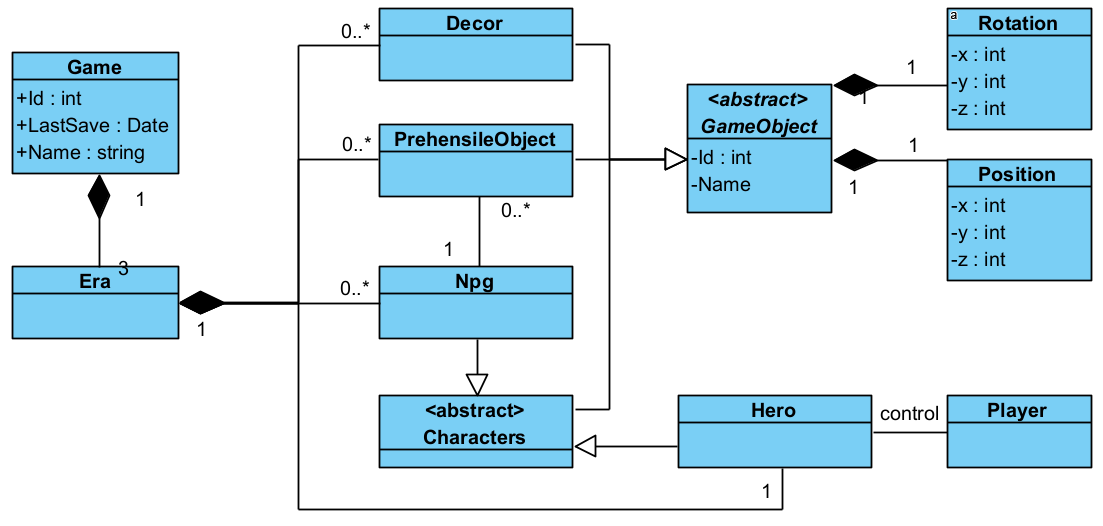
* Donner un objet à un PNJ

De la même façon que pour l’inventaire, quand un joueur tient un objet en main et l’approche d’un PNJ, ce dernier se met en surbrillance. Cela indique au joueur qu’il peut le lui donner. S’il lâche le bouton de *Grip*et que le PNJ accepte l’objet, ce dernier disparaît de la main et est donné au PNJ. Si la réaction du PNJ indique un refus, l’objet retourne alors dans l’inventaire du joueur.

* Donner un objet à un Héros

Comme nous l’avons vu plus haut, l’inventaire comprend deux icônes représentant les autres Héros de l’aventure. Comme pour l’inventaire, le *Drag and Drop* d’un objet sur le portrait d’un des Héros transfert cet objet dans l’inventaire de ce Héros.

Diagramme de classe



* Game

Il est instancié lorsqu’un joueur charge l’une des sauvegardes ou quand il crée une nouvelle partie. Les attributs d’une partie (game) sont:

**- Id**

Il est un Integer unique et sert à identifier la partie. Il est auto-incrémenté par la système.

**- name**

un String qui est uniquement accessible en get() et est une concaténation du mot “partie “ et de l’Id de la partie.

**- lastSave**

De Type date, il indique le moment de la dernière sauvegarde.

**- era**

Une partie contient toujours trois époques (Era): Le présent, le passé et le futur. Ils sont liés par un lien de constitution, la suppression d’une partie entraînant, en cascade, la destruction de ces trois époques. La collection d’époque a toujours une longueur de 3 et est initialisée à l’instanciation de chaque partie. On ne peut ni ajouter, ni retirer d’époque d’une partie.

* Era

Les époques (era) sont des parties du jeu qui sont des mondes dans lesquels les trois joueurs évoluent en parallèle.

Chaque époque contient:

**- decors**

Collection d’objets Decor. Chaque époque a des décors différents et leur nombre est variable.

- **PrehensileObjects**

Collection d’objets préhensiles qui sont présent à la base dans chaque époque mais peuvent passer de l’une à l’autre.

* Gameobject

Représentent tout les objets pouvant être présent dans le monde. Il s’agit d’une abstraction, qui possède les arguments suivants:

**- Id**

**- Name**

**- Rotation**

Composé de trois integer: x, y et z qui sont compris entre 0 et 360. Ils présentent, en effet le nombre de degrés de la rotation qu’a subit l’objet sur chacun des trois axes.

**- Position**

Composé de trois integer: x, y et z qui servent à localiser un objet dans un environnement tridimensionnel.

* Decor

Il s’agit des éléments statique du jeu tels que les murs, les portes ou les meubles. Ils ne peuvent être déplacés par le joueur.

* PrehensileObject

A l’inverse ces derniers peuvent être déplace et tenu par le joueur.

* Npc

Ils sont les personnage non-joueurs (Non Player Charecter) qui sont dans chaque époque.

Npc hérite de la classe GameObject ett a les attributs suivants:

**- Inventary**

liste de PrehensileObject en possession du NPC. Il peut donner l’un de ces objets au joueur ou en recevoir un.

**- AcceptedObjects**

liste de PrehensileObject que le NPC peut accepter dans son inventaire.

**- Dialogues**

Listes de strings qui sont les phrases que peut dire le Pnj.

Héros

Les Héros sont aussi des personnages mais ils sont liés à un joueur qui les contrôle.

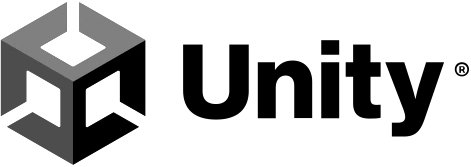
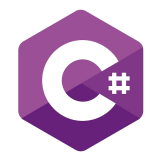
Architecture

Le choix du multijoueur a été fait afin de pouvoir mettre ne place une architecture backend - frontend. Le sauvetage des données nécessitera l’utilisation d’une Db.

Choix des technologies

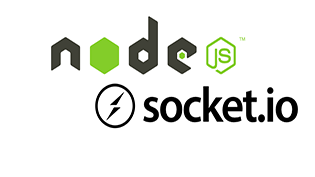
* Frontend : le jeu

Ce jeux sera développé sur Unity qui utilise le language C# pour ces écritures de script.

* Backend: le serveur de jeu

Je développerai le backend en Node Js, cette technologie étant utilisée par Socket IO qui me servira à assurer une communication constante entre les backend et le frontend en cours de jeu.



* Base de donnée

