

# Cahier des charges RV01

## Goldberg's Dream

### Introduction

#### Contexte de réalisation

Ce projet est développé dans le cadre de l'enseignement RV01 (réalité virtuelle) durant le semestre d'automne 2016. Le projet est réalisé par Alexandre Guth et Baptiste Montange, tous deux GI05, et encadré par Indira Thouvenin et Florian Jeanne.

#### Description du projet

Le projet propose une expérience de réalité virtuelle permettant de créer et de manipuler des machines de Goldberg. Il s'agit d'une machine automatique permettant de réaliser une tâche simple de manière très compliquée.

<https://www.youtube.com/watch?v=GOMIBdM6N7Q>

Une grande partie de l'attrait de ces machines est l'effet domino autour duquel elles sont construites, c'est à dire qu'une action en entraîne une autre jusqu'à ce que le but de la machine soit rempli.

Ainsi, nous proposons de placer le joueur dans la position de créateur de machine pour résoudre des puzzles. Le projet consistera ainsi à créer un world builder permettant au joueur de réaliser ses propres machines.

#### Matériel mis en œuvre

##### *Dispositifs de visualisation :*

Afin d'offrir la meilleure expérience pour l'utilisateur, nous nous sommes tournés vers le dispositif CAVE. Il a comme avantage de pouvoir fonctionner sans casque intégral et de permettre un déplacement relativement libre de l'utilisateur. Comme les machines de Goldberg sont volumineuses et souvent repliées sur elles même, l'utilisateur pourra se placer en plein milieu des différents composants et ressentir une expérience de visualisation originale.

L'inconvénient principal du CAVE est qu'il sera moins immersif lors de la mise en mouvement des composants de la machine : en effet, avec un casque de type Oculus, l'utilisateur pourrait être plongé au sein de la machine, suivant le même parcours que les

différents composants de cette dernière. Avec le CAVE, l'expérience sera moindre étant donné la plus grande liberté de mouvement de l'utilisateur.

*Dispositifs d'interaction :*

Nous avons choisi d'utiliser le dispositif Razer Hydra pour gérer les interactions de l'utilisateur avec l'environnement. En effet, nous avons besoin d'un dispositif avant tout réactif et précis, quitte à sacrifier un peu d'immersion pour l'utilisateur.

*Plateforme de développement :*

Nous utiliserons l'éditeur Unity 5.4 afin de réaliser notre projet.

# Description du projet

## Sélection des niveaux

Afin d'accentuer le thème du jeu, nous pensons créer une machine de Goldberg qui se déroulera au fur et à mesure que l'utilisateur complète des niveaux.

Ainsi, au début du jeu, l'utilisateur voit la machine se mettre en branle puis s'arrêter, et le premier niveau se lance. Après complétion, la machine se remet en marche jusqu'au second niveau, et ainsi de suite.

## Description des puzzles

Le cœur de notre projet est la réalisation des puzzles.

Ces derniers seront centrés autour d'un thème : la machine de l'utilisateur doit permettre de réaliser une action simple, par exemple ouvrir une porte ou tourner la page d'une livre. Le thème sera mis en avant par un style visuel simple et des couleurs vives, mettant en avant l'objectif.

Afin d'orienter l'utilisateur dans la construction de sa machine, un système de checkpoint sera présents dans les niveaux : la machine créée devra impérativement passer par ces différents points afin de résoudre l'énigme.

Pour valider un niveau, l'utilisateur doit lancer sa machine et vérifier qu'elle fonctionne bien : dans le cas contraire, il devra la modifier afin qu'elle remplisse la tâche demandée.

## Construction des puzzles

Nous avons identifiés trois phases constituant la boucle de construction des puzzles :

Phase d'observation et de déplacement :

La première action de l'utilisateur sera de se déplacer dans l'espace.

Afin de gérer la caméra, l'utilisateur dispose de deux options. D'une part, il peut tourner la tête et se déplacer librement dans le CAVE, ce qui lui permet d'avoir une vue partielle de son environnement. D'autre part, l'un des sticks analogiques de la manette sera utilisé pour gérer les mouvements de la caméra, dans toutes les directions.

Afin de se déplacer, l'utilisateur dispose de deux options. La première est d'utiliser le second stick de la manette, pour des déplacements courts et précis. La seconde est de pointer vers un élément dans la scène, de la sélectionner (interaction décrite après), puis d'appuyer sur un bouton qui interpolera la position courante de l'utilisateur et celle de l'élément pointé, afin de déplacer le joueur au niveau de ce dernier.

Nous aimerions expérimenter une technique de déplacement spéciale, où l'utilisateur ferait défiler l'écran en « s'accrochant » à un endroit et en bougeant la manette dans le sens où il

souhaite se déplacer. Cependant, nous ne sommes pas sûrs de la pertinence de cette interactions, et nous testerons donc comment l'implémenter.

Phase de placements d'objets :

Afin de construire sa machine, l'utilisateur aura accès à plusieurs composants qu'il pourra agencer comme il le souhaite.

Afin de sélectionner le composant qu'il désire, l'utilisateur pointe l'une des manettes vers l'écran et appui sur le bouton correspondant. Une roue de sélection s'affiche alors à l'endroit pointé par l'utilisateur. Ce dernier navigue dans les différents menus, puis sélectionner le composant de son choix, soit en bougeant la manette déclencheuse de manière circulaire, soit en pointant une partie de la roue avec la manette.



Nous avons identifiés deux types principaux de composants, qui supposent deux méthodes de sélection différentes :

- Composants préfabriqués : l'utilisateur positionne son autre manette sur le camembert correspondant, et réalise un glisser déposer de l'objet dans la scène. Afin de faciliter le placement de l'objet dans la scène, un aperçu de l'élément sera placé,
- Composants procéduraux : certains éléments, comme les rampes, seront utilisables d'une manière particulière : en effet, l'utilisateur sélectionne le type de composant procédural qu'il souhaite, puis trace une forme dans l'espace. Le composant sera alors instancié en suivant cette forme, permettant des rampes customisées répondants au besoin du puzzle.

Une fois les éléments placés dans la scène, on peut les manipuler.

Phase de manipulation d'objets :

L'utilisateur dispose de plusieurs options pour manipuler les objets :

- Déplacement d'un élément : il suffit à l'utilisateur de pointer l'objet désiré et de le verrouiller à l'aide d'un bouton. L'élément suivra alors les déplacements de la manette de l'utilisateur. Pour des déplacements précis, le joueur pourra également utiliser le stick analogique d'une manette.
- Rotation d'un élément : l'utilisateur pourra utiliser un système de pivot pour les grandes rotations. Il sélectionne d'abord l'objet comme pour un déplacement, puis utilise la seconde manette pour faire tourner l'objet dans la direction qu'il souhaite, en respectant le pivot. L'autre système sera de réaliser des rotations en fonction du centre de l'objet, avec le stick analogique. Le joueur passera d'un axe de rotation à l'autre en appuyant sur un bouton.
- Mise à l'échelle d'un élément : cette interaction fonctionne comme la rotation, mais pour les axes de scaling.

Interface utilisateur

Afin de faciliter l'expérience utilisateur, nous pensons implémenter les interfaces suivantes :

En premier lieu, nous prévoyons d'afficher en permanence deux pointeurs sur l'écran, permettant à l'utilisateur de repérer l'emplacement de ses manettes par rapport à l'environnement. Ces pointeurs permettront de mettre en valeur les possibilités d'interactions avec les éléments des décors, par exemple en surlignant les lignes de l'objet ou bien en changeant de couleur.

Autour de ces manettes, nous ajouterons des dialogues d'aide contextuelle, rappelant à l'utilisateur les interactions disponibles en fonction de la situation. Par exemple, si l'utilisateur sélectionne un élément, il lui sera rappelé qu'il lui faut appuyer sur un bouton pour changer la taille de l'objet sélectionné.

Pour finir, nous implémenterons les feedbacks nécessaires, sonores pour un retour sur les différentes actions, et visuels notamment pour la différenciation des actions lors de la sélection. Par exemple, nous pensons surligner de couleurs différentes les objets pointés et sélectionnés pour signifier la possibilité d'une interaction :

Rouge pour statique, Vert pour sélectionnable, Jaune pour sélectionné, et ainsi de suite.

## Risques et contraintes

La principale difficulté de ce projet sera le manque de temps. En effet, nous ne maîtrisons pas vraiment les outils que nous utilisons, et nous ne disposons que de deux mois et demi pour réaliser ce projet. Ainsi, la prise en main des différents éléments et de l'éditeur Unity devra être rapide sous peine de ne pas finir le projet dans son intégralité.

Trouver des assets originaux pour créer des machines visuellement attrayantes est également possiblement compliqué, mais il ne s'agit pas d'une priorité pour le projet, et l'asset store est remplis de produits gratuits de qualité.

## Niveaux de réalisation

Réalisation minimale.

L'objectif minimal est la création d'un puzzle complet, avec les interactions décrites précédemment à l'exception de la rotation par pivot. Implémentation de composants préfabriqués mais pas des composants procéduraux. Interface simple.

Réalisation intermédiaire.

Ajout de la rotation par pivot et des composants procéduraux. Interface élaborée et feedbacks complets.

Réalisation maximale.

Création de plusieurs niveaux, et de l'écran titre sous forme de machine de Goldberg.

## Planning

Début septembre - 14/10 : Définition du projet et réalisation du cahier des charges.

14/10 - Toussaint : Exploration des interactions, création des premiers composants, interface principale.

Toussaint-30/11 : Finition des interactions, réalisation du premier niveau.

02/12 : Revue de projet

02/12-Vacances de Noël : Implémentation des feedbacks et de l'interface complémentaire.

Création de niveaux supplémentaires et de l'écran titre.

Fin des Vacances : rendu du projet.