

# GARCIA KEVIN

## DÉVELOPPEUR LOGICIEL / WEB / JEUX / VR

VISUAL STUDIO, UNITY (C#, C++, JAVA, ETC...)

WEB BACKEND (PHP, MYSQL, SYMFONY2, APACHE, ETC...)



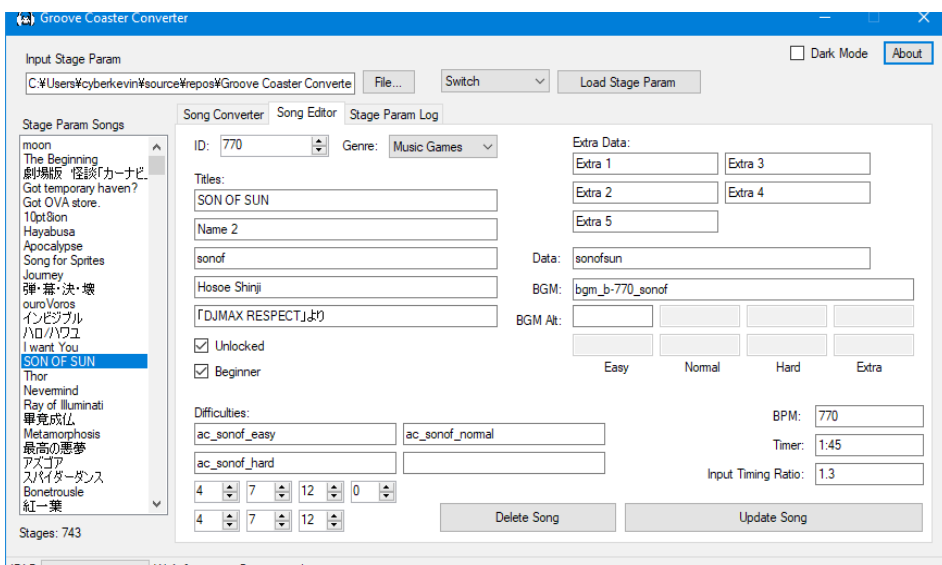
DU LOGICIEL DE GESTION DE DONNÉES BINAIRES, AU JEUX EN RÉALITÉ VIRTUELLE.

# DÉVELOPPEUR WEB, LOGICIEL, DE JEUX 2D/3D ET APPLICATIONS VR

## APPLICATIONS LOGICIEL

Création d'applications simples pour la gestion de données sous toutes ses formes, réalisation d'interfaces graphiques sans fioritures pour une utilisation complète et simplifié le plus possible la production de contenu via ces applications.

## GROOVE COASTER CONVERTER

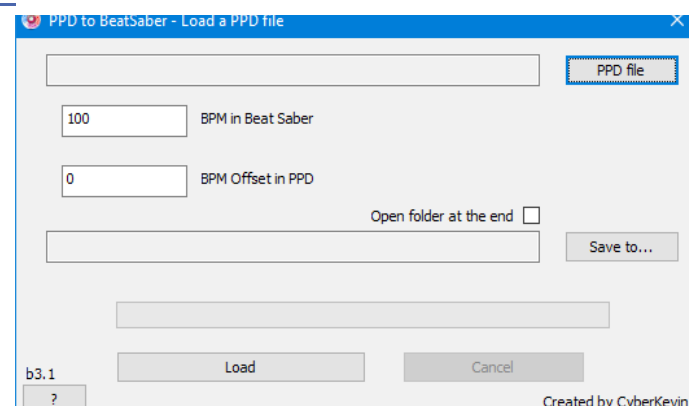


Application Open Source permettant le décodage de fichiers en format propriétaire, afin d'en modifier le contenu de manière simple.

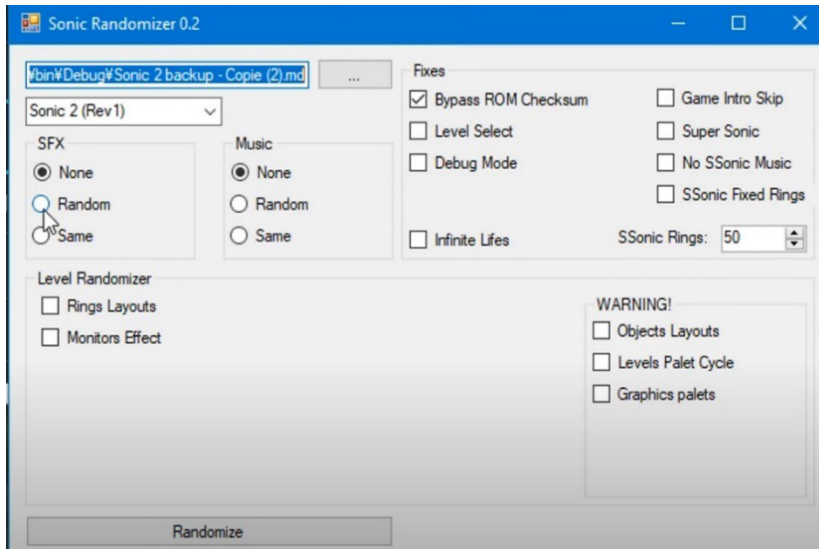
Il permet aussi la conversion de données d'un format propriétaire à un autre, ainsi que la sortie de fichiers fonctionnels pour l'application propriétaire.

Lien : <https://github.com/oocyberkevinoo/Groove-Coaster-Converter>

## PPD TO BEATSABER



Application basée sur la même fonctionnalité que Groove Coaster Converter, mais orientée sur la conversion de chansons jouables dans un jeu de rythme à un autre jeu de rythme.



## SONIC RANDOMIZER

Application Open Source permettant d'altérer les données d'un fichier ROM de jeux SEGA MegaDrive, afin d'y apporter de l'aléatoire dans le fonctionnement du jeu.

Se basant sur une base de données connue de jeux, il détermine à quelle adresse se trouve chaque fonctionnalité à altérer, puis créer une nouvelle structure de données selon les réglages demandés par l'utilisateur.

Lien : <https://github.com/oocyberkevinoo/Sonic-Randomizer>

Version non altéré



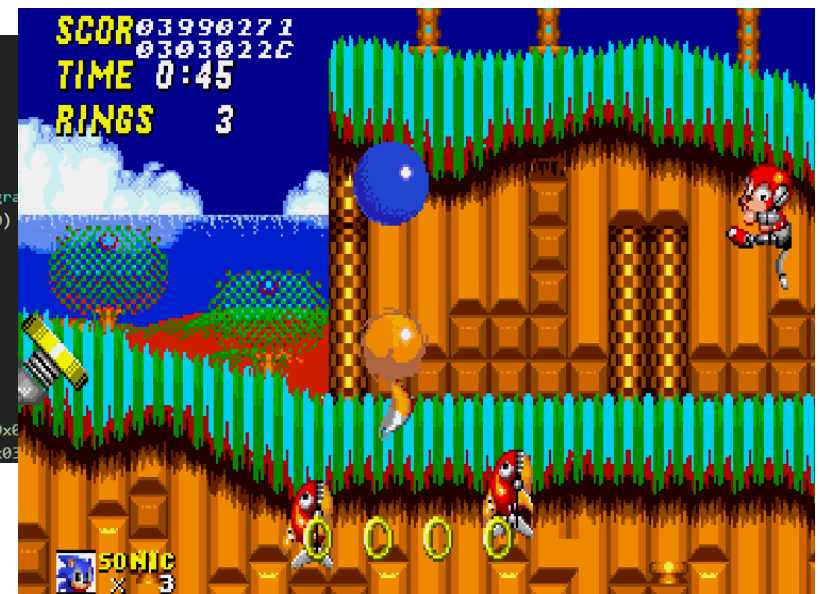
```
int gameSelection = -1;
int revSelection = -1;
int lockon = 0;
int offset = Datas.Games.offset;

Program.reader = new BinaryReader(File.Open(Program.
if(Program.reader.BaseStream.Length > 0x00200000)
{
    lockon = 1;
    offset += 0x00200000;
    Functions.MessageHandler.ErrorMessage(7);

Program.reader.BaseStream.Position = offset;

Byte[] gameID = Functions.DataManager.readData(0x0
Byte[] revID = Functions.DataManager.readData(0x03
```

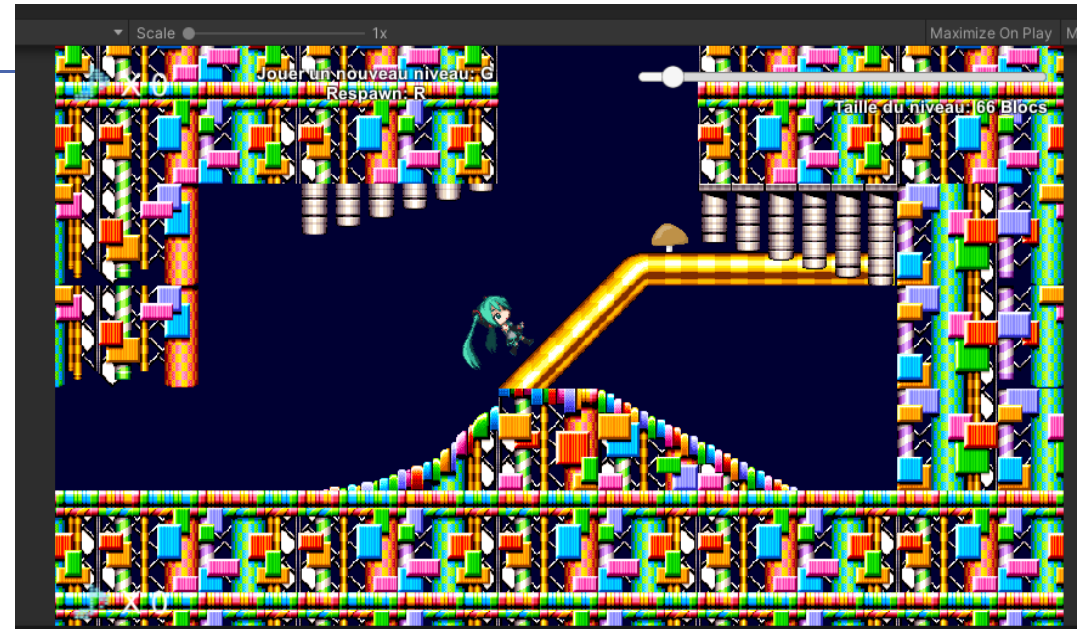
Version altéré par l'application



## PROTOTYPE : JEU DE PLATEFORME PROCÉDURAL

Prototype d'un jeu de plateforme générant ses niveaux de manière procédurale.

Il vérifie toujours l'accessibilité de chaque salle pour créer un niveau cohérent et jouable de bout en bout.



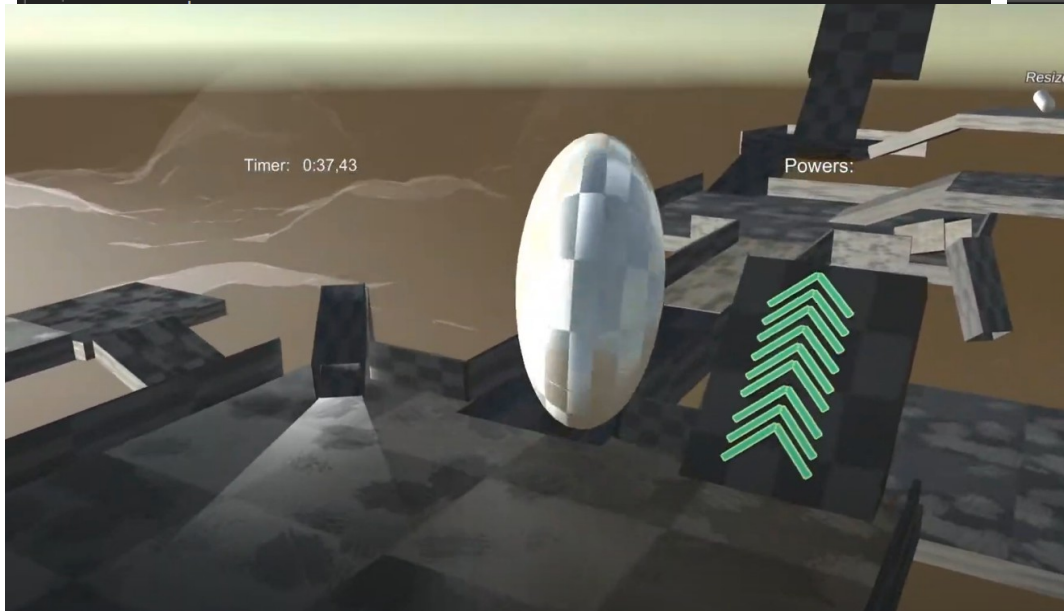
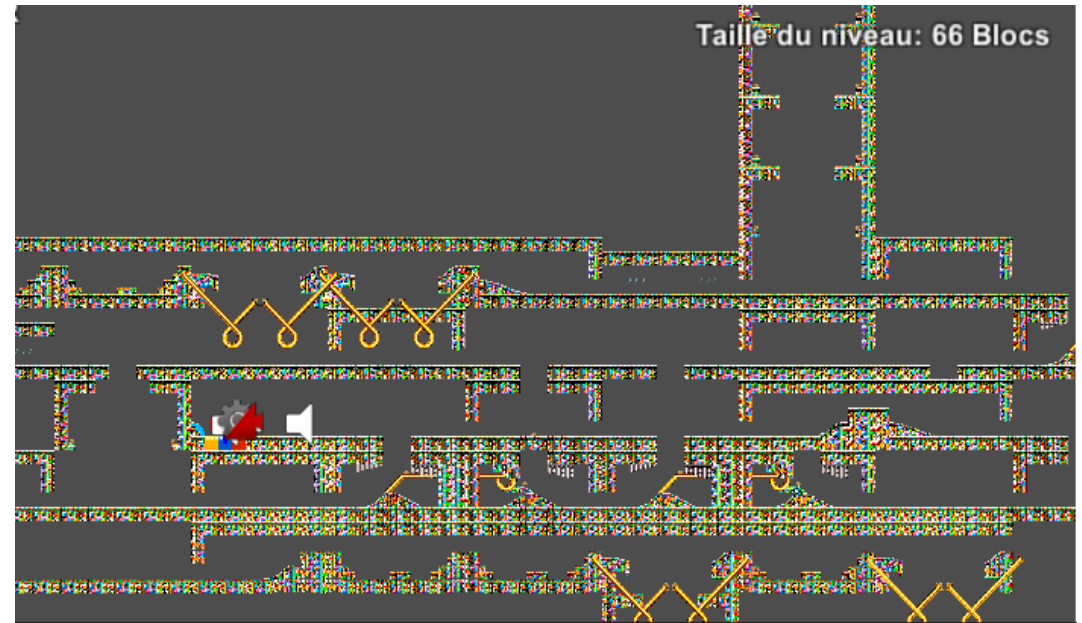


```

while (!flag) // Tant que la salle n'a pas été créée.
{
    int x;
    counter++;
    x = Random.Range(0, tilemapList.transform.childCount); // On choisit un type de salle au hasard.
    GameObject tilemap_selected;
    GameObject new_tilemap = null;
    int chance;
    tilemap_selected = tilemapList.transform.GetChild(x).gameObject;
    chance = tilemap_selected.GetComponent<tilemap_prefab>().chance;
    x = Random.Range(0, 100); // et on calcul la probabilité...
    if(x <= chance)
    {
        temp_cursor = cursor;
        cursor = CursorPlacement(mode, tilemap);
        new_tilemap = Instantiate(tilemap_selected, cursor, Quaternion.identity);
        new_tilemap.SetActive(true);
        new_tilemap.transform.SetParent(grid.transform);
        flag = TilemapAccessible(mode, tilemap_selected);
    }

    if (flag)
    {
        roomQuantity--; // Et on compte la nouvelle salle.
        rooms++;
        new_tilemap.GetComponent<tilemap_prefab>().previous_tilemap = tilemap;
        new_tilemap.GetComponent<tilemap_prefab>().previous_mode = mode;
        new_tilemap.GetComponent<tilemap_prefab>().done = true;
    }
    else
    {

```



## PROTOTYPE : GEOPOTE

Prototype simpliste de jeu de plateforme/énigme coopérative basée sur la possibilité de jouer des formes géométriques simples, mais pouvant se déformer via des capacités spécifiques.

La coopération réside dans la déformation d'un des deux joueurs afin de passer les énigmes de manière toujours différentes.

```

public void Resize()
{
    if (pl_action.resize)
    {
        if (Input.GetButton(player.getKey("Action1")) && (sizeX < normalMaxSize || sizeY < normalMaxSize || sizeZ < normalMaxSize) && !pl_move.crushed)
        {
            if (pl_action.larger && (sizeX < normalMaxSize || sizeZ < normalMaxSize))
            {
                sizeX += normalResizeForce;
                sizeZ += normalResizeForce;
            }
            else if (pl_action.bigger && sizeY < normalMaxSize)
            {
                sizeY += normalResizeForce;
            }
            else if (!pl_action.bigger && !pl_action.larger)
            {
                if (sizeX < normalMaxSize)
                    sizeX += normalResizeForce;
                if (sizeY < normalMaxSize)
                    sizeY += normalResizeForce;
                if (sizeZ < normalMaxSize)
                    sizeZ += normalResizeForce;
            }
        }
    }
}

```

DÉVELOPPEUR WEB, LOGICIEL, DE JEUX 2D/3D ET APPLICATIONS VR

JEUX ET APPLICATIONS VR SOUS UNITY

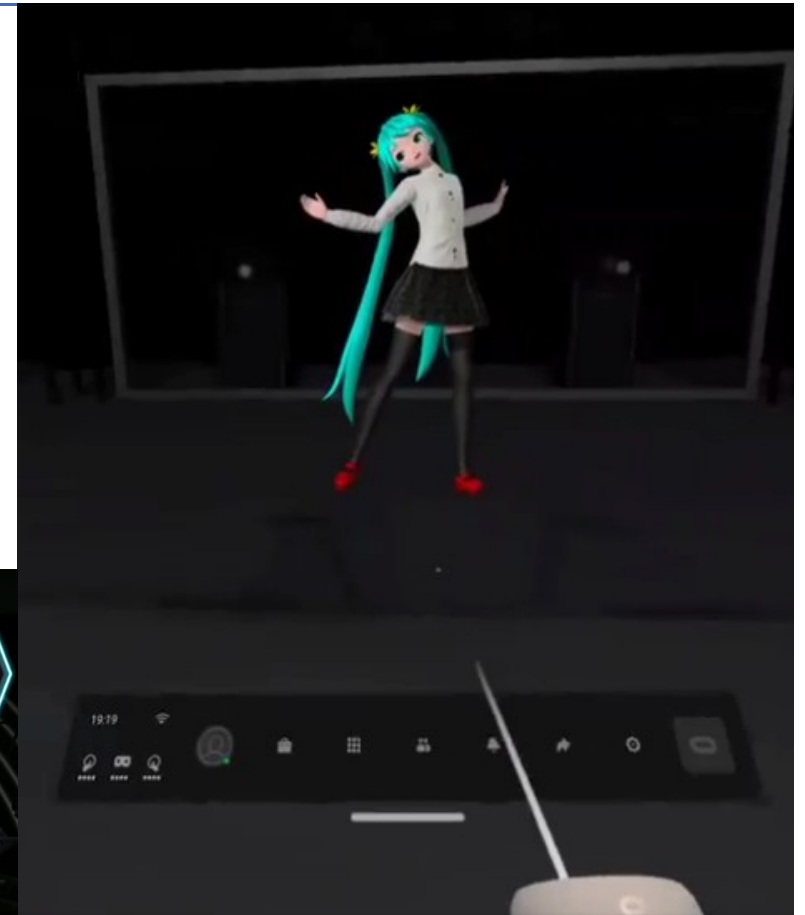
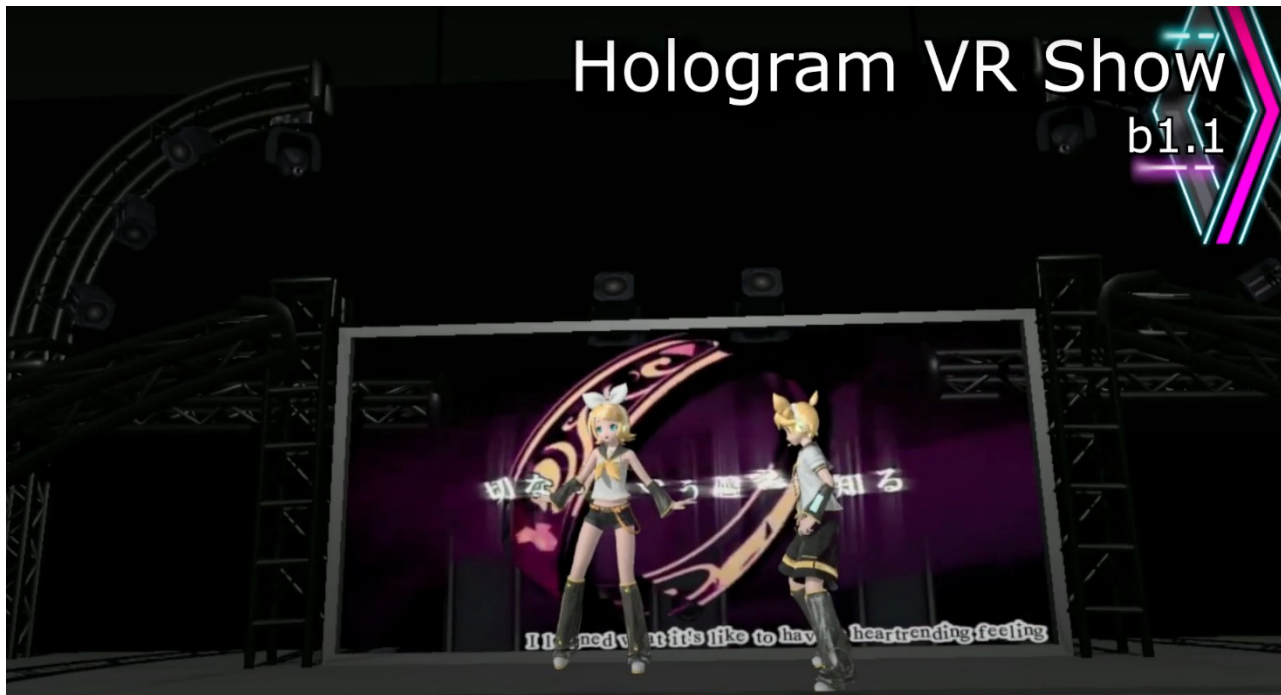
## HOLOGRAM VR SHOW (PC VR / OCULUS QUEST)

Une application en réalité virtuelle qui permet de lire des fichiers vidéo sur une scène de concert holographique virtualisé.

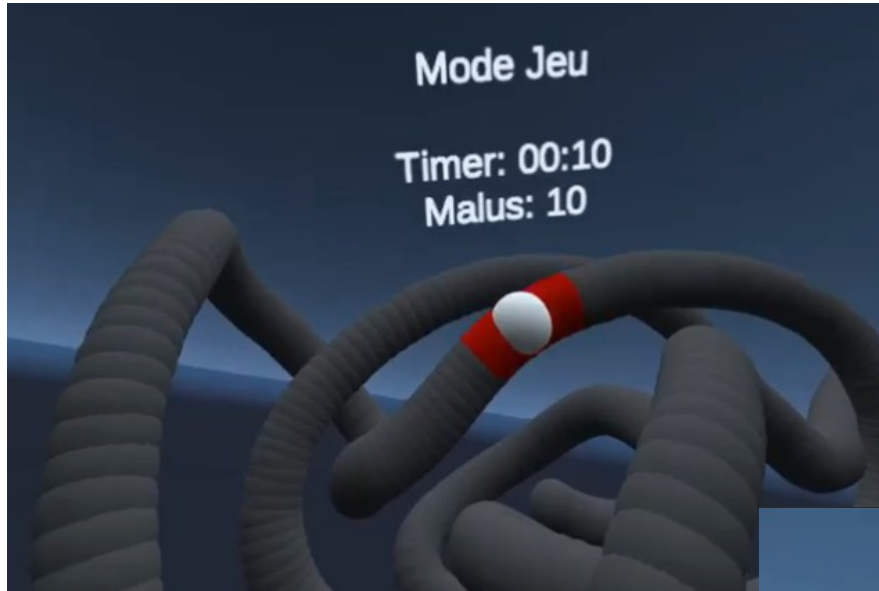
En y diffusant des vidéos dites « Hologram Ready », l'application se charge de retirer la couleur de fond de la vidéo afin de la faire apparaître sur la scène comme un hologramme.

On peut y ajouter des vidéos de fond, des vidéos sur le décor, et autres effets de mises en scène.

Version PC en vidéo et téléchargement : <https://www.youtube.com/watch?v=eaRNChswiAE>



## CIRCUIT VR (OCULUS QUEST)

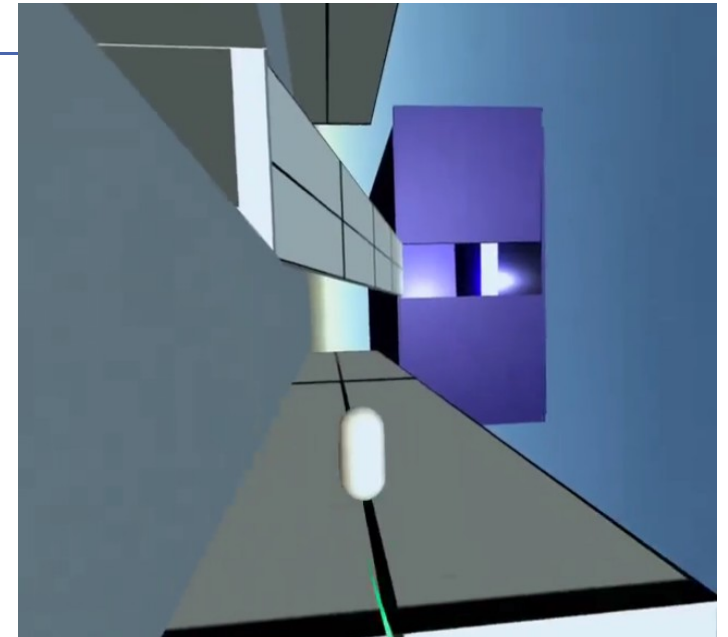
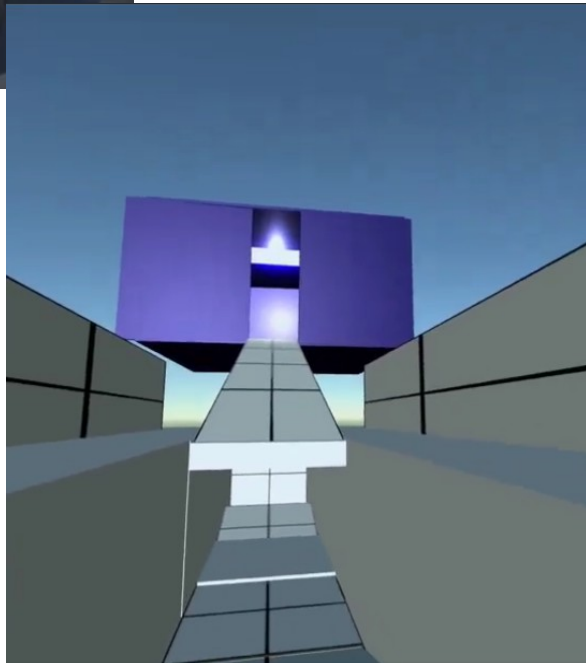


Projet de jeu en réalité virtuel multijoueur utilisant un casque VR.

Le joueur portant le casque VR doit dessiner un circuit. Le 2<sup>e</sup> joueur devra compléter à l'aveugle le circuit, sans en dévier et uniquement via les indications du joueur portant le casque.

Chaque sortie de circuit provoquera une vibration intense de la manette pour prévenir le joueur de se remettre en place, et le score diminuera au fil du temps que le joueur restera hors zone.

Le jeu n'est pas encore disponible publiquement par manque de finition, mais sa partie jeu est fonctionnelle.



## TRANSPPOSITION VR (OCULUS QUEST)

Un projet de jeu en réalité virtuel demandant au joueur de traverser des niveaux en se téléportant de surface en surface, changeant la gravité selon l'orientation de la surface.

Le simple fait de se retrouver sur un mur créé une sensation de perte, et le joueur devra se forcer à comprendre où il se trouve afin d'avancer en utilisant la physique et sa logique.