Dossier de conception et de développement de Laser Room

Bureau d’étude 2018

L3 Informatique

Sidahmed ARROUI

Quentin BOONE

Lucas CLEMENT

Index

[**I - Conception**](#_fgaj5sdb4y4d) **2**

[1 - Prototype et IHM](#_4tgzu9bq6qey) 2

[2 - Level design](#_aj031ybgvsf2) 10

[3 - Progression](#_migpxd3k9zd7) 13

[**II - Développement**](#_i882u7o4hmin) **14**

[Menu principal](#_9nt7m4oza5ug) 14

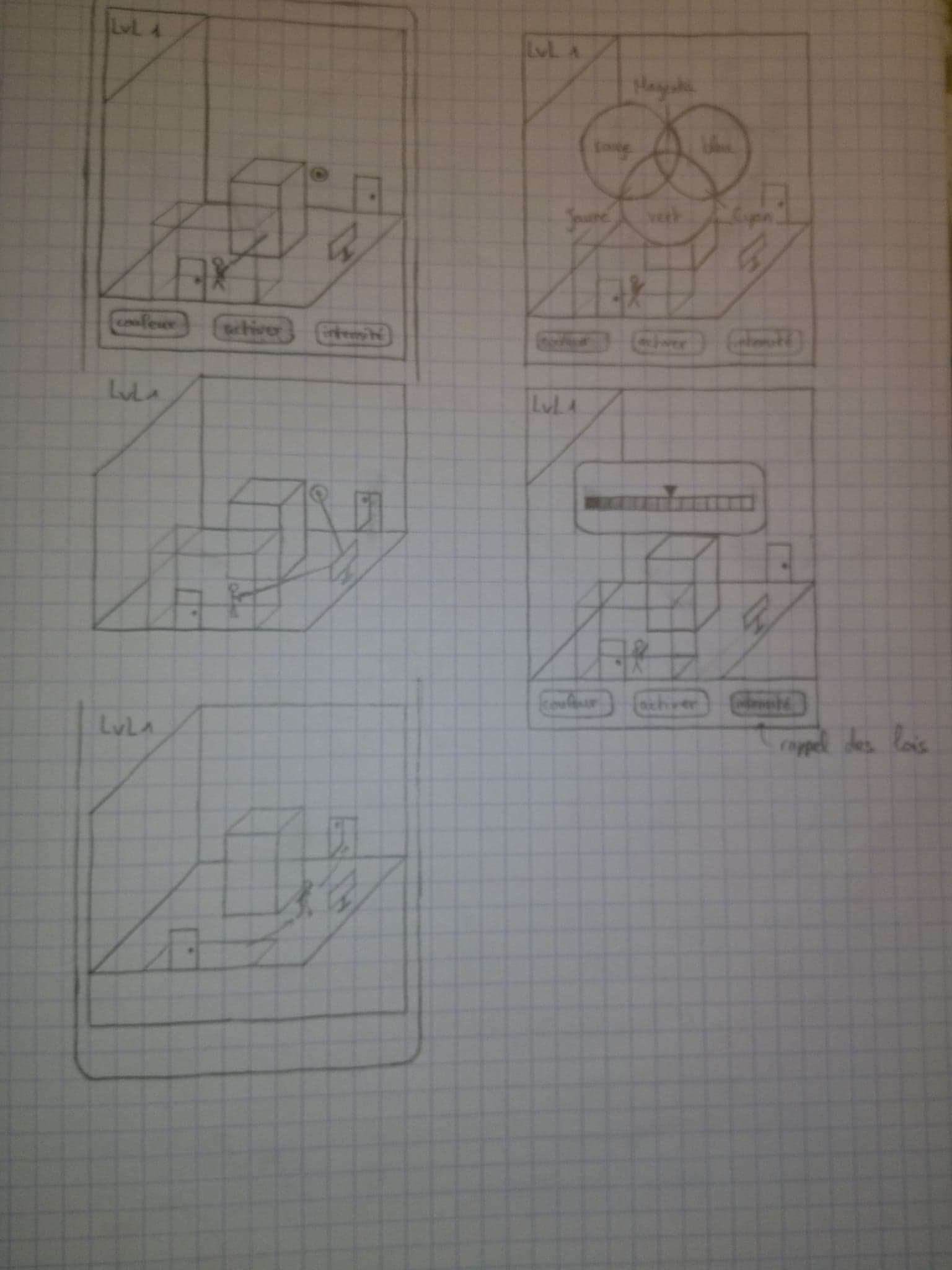
[Menu in game](#_4er87if58jh) 15

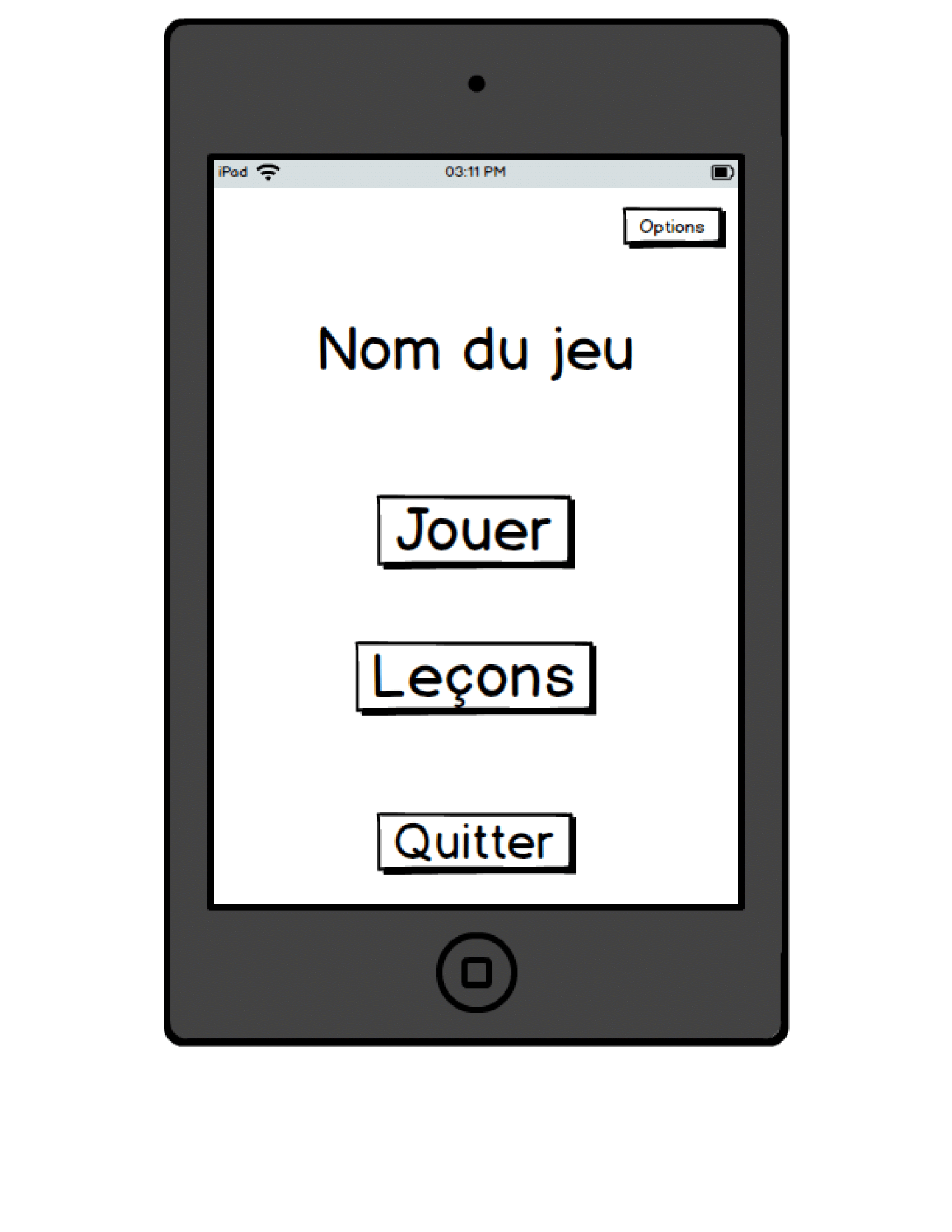
[Miroirs et canon](#_ttsu86i724ik) 16

[Laser](#_i0wnj4p0wdhm) 16

# I - Conception

## 1 - Prototype et IHM





## 

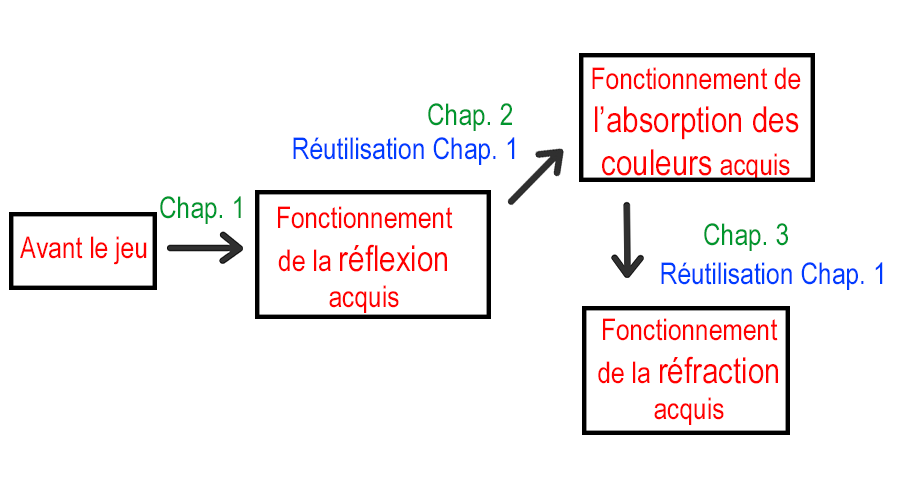
## 

## 

## 

## 2 - Level design

## 3 - Progression

Le premier chapitre sur la réflexion est essentiel pour le reste du jeu puisque c’est le principe de base. Le chapitre sur la réfraction se débloque après avoir fait celui sur les couleurs, mais ils sont indépendants entre eux. Le chapitre réfraction pourrait donc être le chapitre 2 à la place.

# II - Développement

## Menu principal

Le menu principal est implémenté en 2D sur un canvas. Il regroupe 3 menus à savoir le menu principal qui permet d’accéder aux autres, le menu d’options, le menu des niveaux et le menu des leçons. On passe d’un menu à l’autre avec des set actifs, chaque menu étant un objet, ils partagent donc tous le même canvas.

Le menu de niveaux permet de charger différentes scènes où chacune correspond à un niveau. Un script permet de stocker en mémoire le plus haut niveau atteint afin de ne rendre interactif que les boutons allant vers des niveaux accessibles. Un bouton retour permet de revenir sur le menu.

Le menu des leçons propose 3 différents cours très basique (d’une page) qui marchent eux aussi avec des sets actifs lorsque l’on sélectionne le bouton associé. Un bouton retour permet de revenir sur le menu.

Le menu d’option permet de gérer le volume à l’aide d’un slider, sa valeur est sauvegardée de façon à ce que le volume ne se remette pas à sa valeur initial à chaque lancement du jeu. Un bouton reset permet de remettre à zéro l’avancement des niveaux. Il réinitialise simplement la valeur qui correspond au niveau le plus haut fermant l’accès à tous les niveaux à part le premier.

backToMenu : Est appelé sur le menu principal pour savoir si on doit afficher le menu principal ou celui des niveaux à l’aide d’un int stocké en mémoire. A 1 le menu des niveaux est chargé, à 0 c’est le menu principal qui est chargé.

MainMenu : Charger le menu et permet de quitter le jeu.

SliderMemo : permet de fixer le curseur du slider sur la valeur du son stocké en mémoire.

SoundManager : Permet de sauvegarder l’intensité du son en mémoire pour éviter de le régler à chaque connexion ou changement de scène.

TopNiveau : Permet de récupérer la valeur du niveau (le plus haut niveau réalisé), empêcher l'interaction avec les boutons qui amènent au niveau supérieur au menu max atteint et incrémenter le niveau max atteint.

## Menu in game

Le menu in game est composé de multiples menus, un menu principal et un menu pour chaque objet de la scène qui vient remplacer le menu principal lorsque l’objet qui lui est associé est sélectionné. Pour gérer l’ensemble des menus il a fallu mettre en place un menu manager qui permet de gérer le passage de l’un à l’autre sans quoi ils se superposent et deviennent inaccessibles. Il est possible d’accéder à des leçons depuis n’importe lequel de ces menus. Un bouton d’options fait apparaître une fenêtre qui permet de revenir soit au menu principale soit directement au menu des niveaux, ce changement est réalisé à l’aide d’une valeure stockée en mémoire qui permet de savoir quel menu doit être affiché.

ColorButton : Contient une fonction qui permet de voir si oui ou non l’objet à une couleur et si non désactive le bouton de couleur.

GetHelp : Contient une fonction qui désactive toutes les fenêtres d’aide en jeu pour ensuite pouvoir n’en afficher qu’une seul et ne pas avoir de problème d’interactivité avec les fenêtres sous-jacentes.

InGameBackToMenu : a le même comportement que back to menu mais ne prend pas les menus en paramètres ce qui permet de supprimer des erreurs dues à des variables sans valeurs.

MenuManager : Permet de passer d’un menu à l’autre dans la scène à l’aide d’un tableau contenant chacun des menus.

SelectCamera : Le jeu offre 2 vues de caméra différentes, ce script permet de passer d’une caméra à l’autre.

ButtonAction : permet d’activer / désactiver un menu.

Canon : Permet de créer le laser à la pointe du canon et gérer ses modifications lorsque l’on change l’orientation du canon.

ColorMenu : permet d’afficher le pop up des couleurs.

Colors : permet de correctement calculer la taille du pop up en fonction du nombre de couleurs de l’objet (Unity ne proposait que des pop up vers le bas).

## Miroirs et canon

Les informations contenues sur le menu changent d’un objet à l’autre, pour les miroirs on peut changer les couleurs et l’orientation. Des écrans de couleurs différentes sont stockés dans l’objet et seul le miroir de la couleur demandée est affiché, on passe d’une couleur à l’autre simplement en remplaçant un écran par un autre. Pour orienter le miroir un script placé sur le bouton permet de récupérer l’orientation du miroir et l’incrémenter jusqu'à un max choisi à chaque clic. Il est aussi possible de positionner le miroir à son orientation initial.

De son côté le laser peut être activé et désactivé, il est orienté à l’aide d’un drag et les boutons sont uniquement là pour permettre de cibler avec plus de précision.

LiquidCube : Permet d’initialiser l’indice de réfraction d’un liquide.

MirrorColor : Permet de changer la couleur du miroir en fonction du bouton sélectionné par le joueur.

OrientCanon : Permet d’orienter le canon à l’aide de ses boutons.

OrientMirror : Permet d’orienter le miroir à l’aide de ses boutons.

Interaction : Permet de mettre en valeur les objets interactifs lorsque l’on passe la souris dessus.

## Laser

Le laser est un ray cast, c’est à dire qu’il est unidirectionnel, il a une source et une direction. Pour implémenter les différentes fonctions propres au jeu il a donc fallu créer un nouvel objet à chaque fois que le rayon rencontrait un miroir ou un liquide. Une fonction à donc pour but de gérer l’envoi du rayon, voir s’il rentre en contact avec un objet, si oui tester si l’objet est un miroir, un liquide ou un obstacle et tracer ou non ou le rayon suivant en conséquence.

C’est sur ce même script que l’on gère le changement de couleur du rayon, lorsqu’il rencontre un miroir il appelle une fonction qui va récupérer la couleur du miroir et celle du rayon, faire la soustraction des couleurs et crée un nouveau rayon avec la nouvelle couleur et la bonne direction.

Si le rayon rencontre l’objectif il détecte celui-ci et met fin à la partie en affichant le menu qui propose de passer au niveau suivant ou retourner au menu des niveaux, il incrémente également le niveau max atteint si le niveau réalisé est le dernier niveau disponible.

Reflective : Contient les fonctions qui gère la réflexion ainsi que le changement de couleur du laser sur un miroir coloré.

Refractive : Contient les fonctions qui gèrent la réfraction.

Utility : Contient des fonctions basiques pour gérer la réflexion et la réfraction.

Winning : Permet d’afficher le message de victoire et les actions associées lorsque le laser atteint la cible.