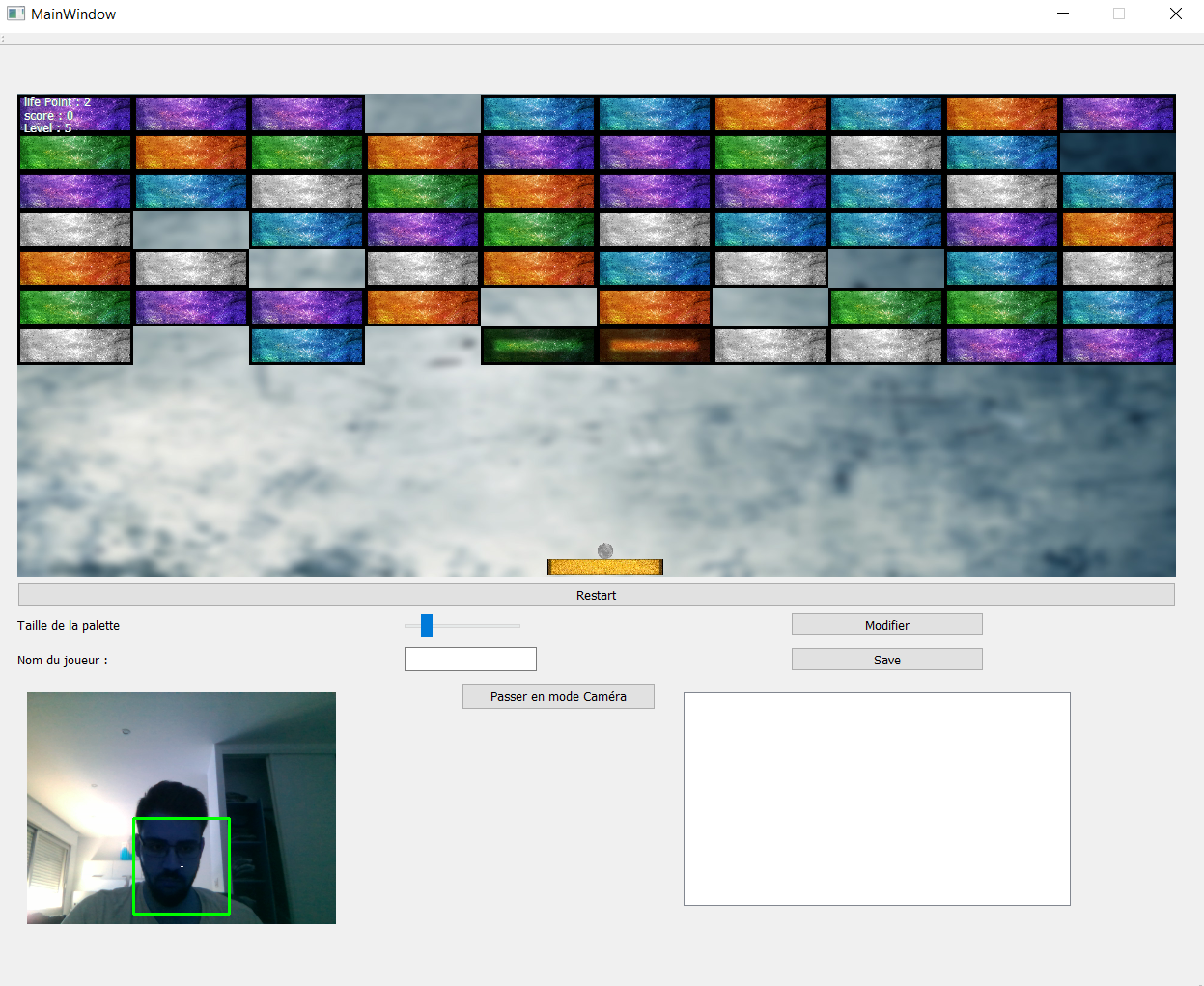


**Spécifications**

Voici une capture de l’interface de jeu :



La partie supérieure de la fenêtre correspond à l’interface de jeu. C’est dans cet interface contenant le widget OpenGL que va se dérouler la partie. Nous pouvons y voir une palette, une boule, des briques, un background, et les informations de la partie dans le coin supérieur gauche. On y trouve le nombre de points de vie restants au joueur, son score et le niveau auquel il se trouve. Le background et la difficulté des briques change jusqu’au 5ème niveau, à partir duquel la difficulté du jeu se répète (à l’exception de la bille qui continue d’augmenter de vitesse). La texture des briques change en fonction de leurs nombres de points de vie et des dommages qu’elles ont subi.

La touche ALT permet de démarrer la partie, c’est-à-dire de lancer la boule.

Il existe deux modes de jeu : le mode Caméra et le mode Souris. Le mode Souris permet de déplacer la palette avec la souris. Le mode Caméra permet de la déplacer avec les mouvements détectés dans la vidéo de la webcam affichées en bas à gauche. Un mouvement horizontal permet de déplacer la palette dans le côté correspondant, et un mouvement vertical permet de stopper le déplacement. Le mode de déplacement par défaut est le mode Caméra, cependant un bouton Passer en mode souris est disponible.

Un bouton Restart permet de réinitialiser la partie (ceci incluant le mode de jeu ainsi que la taille de la palette et la vitesse de la bille).

Un bouton Modifier permet de changer la taille de la palette en fonction de la position du slider qui est situé à sa gauche.

Un label Nom du joueur permet de nommer le joueur afin de sauvegarder le score du joueur par la suite en cliquant sur le bouton Save. Si le score du joueur fait partie du top 10 des meilleurs joueurs il apparaitre dans une liste dans le coté inférieur droit de la fenêtre sinon il ne sera pas enregistré. Le tableau de score est chargé à chaque ouverture de l’application (les données sont sauvegardées dans un fichier texte).

**Conception**

**Un diagramme de classe est disponible dans le .zip : Diagramme de classe.jpg**

MainWindow :   
Cette classe est notre main. C’est ici que nous avons géré la mise en forme de l’interface graphique. Il nous sert de gestionnaire de jeu et gère les interactions avec les boutons et les éléments du jeu.

OpenCVWidget :  
Cette classe contient notre Widget OpenCV. C’est elle qui gère les interactions avec la caméra. Le code qu’elle contient est issu de celui donné en exemple pour l’installation d’OpenCV en début de projet. Elle fait apparaître dans le MainWindow le flux de la vidéo de la caméra les détecte les mouvements qu’on y fait en son sein.

ListPlayer :  
Cette Classe hérite de list<Player> des méthodes de sauvegardes,de trie et de chargement y sont implémentés de façon à pouvoir sauvegarder les scores et les charger.

MyGLWidget :  
Cette classe contient notre Widget OpenGL. C’est dans ce widget que l’on fait afficher l’écran de jeu sur la fenêtre principale. Elle permet de visualiser notre partie. C’est également dans cette classe que l’on charge toutes les textures, pas le fond, la balle, la palette et les briques. C’est elle aussi qui dessine les éléments du jeu et qui les mets à jour en temps réel en fonction des informations renvoyées par le GameManager qui lui est directement relié.

GameManager :  
Cette classe sert d’intermédiaire entre le MyGLWidget, c’est-à-dire l’affichage de la partie, et toutes les classes de ses composants. C’est elle qui permet de gérer les éléments de la partie en fonction des informations qu’elle reçoit des autres classes. Elle va notamment avoir un rôle important pour la gestion des collisions.

Ball :  
Cette classe permet de créer les balles à chaque début de partie et à chaque mort. Elle contient toutes leurs caractéristiques

Player :  
Cette classe permet de gérer les informations relatives au joueur. On y retrouve son score, sa vie ou encore son nom.

Square :  
Cette classe permet de gérer tous les éléments avec lesquels la boule peut rentrer en collision. Elle gère les briques, avec leur type ou encore leurs points de vie restants et maximums, les murs, qui sont de longues briques avec une vie infinie, et la palette, qui possède en plus ses propres caractéristiques.

Palette :  
Cette classe permet de gérer la palette et ses caractéristiques, telles que sa vitesse ou sa taille.

**État de finalisation**

À l’heure actuelle, l’application fonctionne relativement bien.

Les collisions sont gérées correctement. Il n’y a pas d’erreur lorsque la boule doit cogner les murs, ou lorsqu’elle doit être détruite. Les collisions avec les briques se font également bien, avec une gestion des cas où la balle tape dans le coin d’une brique. Il en est de même avec la palette, où la direction dépend de la distance entre le point de collision et son centre. Un point qui pourrait être à améliorer est que l’on ne prend pas en compte l’angle d’incidence au moment de la collision avec la palette mais uniquement la position de cette collision pour déterminer l’angle de renvoi. Utiliser l’angle d’incidence rendrait le jeu plus réaliste.

La gestion des niveaux se fait correctement. Les textures se chargent bien, que ce soit à chaque fin de niveau, mais aussi à chaque collision entre la boule et les briques.

Cependant, nous n’avons pas utilisé de multithreading, donc à une vitesse de balle lente, des petits à-coups se font remarquer durant le jeu, cela reste cependant suffisamment faible pour ne pas avoir été une priorité (mise en place d’un système de compensation suffisant) .

De plus, la gestion de la palette avec la caméra est compliquée. Du traitement d’image préalable à l’utilisation des vecteurs de déplacement aurait pu améliorer cet aspect de notre jeux (observation sur plus de 2 frames par exemple).

**Fichiers d’en tête**

**Dans le .rar est fourni un dossier généré avec Doxygen contenant des fichiers html permettant d’accéder à notre doc. Tous nos fichiers d’en tête y sont accessibles avec les informations de chacune des fonctions que nous avons mis en place(ainsi que beaucoup d’autres informations).**