

# Fi-Info5.1

## Bibliothèques de Développement Multimédia - Projet 2017-2018 : « Casse briques »

### 1. CONSIGNES GENERALES

#### a. OBJECTIFS PEDAGOGIQUES DU PROJET

L'objectif est de renforcer et de mettre en pratique les compétences du bloc « Bibliothèques de développement multimédia » à travers le développement d'un petit projet. Le développement se fera en C++ sous l'environnement QtCreator en utilisant les bibliothèques Qt, OpenGL et OpenCV. A l'issue de ce projet vous devrez être capable de :

- Paramétrer l'environnement de développement pour utiliser des bibliothèques open source.
- Utiliser les principales fonctions des bibliothèques Qt, OpenGL et OpenCV.
- Gérer la création et la visualisation d'une scène 3D dynamique.
- Concevoir et réaliser une interface utilisateur conviviale.
- Mettre en place une interaction avec l'utilisateur à partir de l'acquisition et du traitement des images issues d'une WebCam.

#### b. TRAVAIL DEMANDE

Le travail demandé concerne la conception et le développement d'un jeu de casse briques.



L'application est décrite plus en détails dans la section 2 de ce document. Le développement devra être réalisé en C++. Vous devrez rendre un rapport au format PDF ainsi qu'une archive ZIP contenant le programme exécutable, les codes sources, le fichier projet et tous les fichiers nécessaires à la compilation.

Le rapport devra comporter :

- 1 page maximum de spécifications présentant l'interface utilisateur et décrivant les principales interactions possibles avec l'utilisateur,
- 2 pages maximum de conception présentant les principales classes que vous avez utilisées en précisant le rôle de ces classes et leurs relations (faire un diagramme des classes),
- 1 page maximum précisant l'état de finalisation de l'application : les fonctions qui ont été validées, celles qui ne sont pas finalisées, les bogues qui subsistent,...
- Les fichiers d'entête des classes (déclaration des classes et des fonctions) qui devra contenir des commentaires sur le rôle des champs et des méthodes utilisés (rôle de la méthode, paramètres d'entrée, de sortie, de retour et, le cas échéant, l'algorithme utilisé) ainsi que le nom de l'auteur.

Soyez clair et synthétique. Merci de respecter le nombre de pages.

#### c. EVALUATION

La note finale du projet est collective et vient s'ajouter aux autres notes individuelles du module « Bibliothèque de développement multimédia ». L'évaluation du projet est notée sur 20. Elle est calculée à partir des 3 éléments suivants :

- Evaluation du rapport (30%),
- Fonctionnement de l'application évalué pendant la démonstration (45%)
- Evaluation des fichiers sources et exécutable (25%),

Le fonctionnement de l'application sera évalué pour chaque binôme lors d'une séance de démonstration qui aura lieu le **23 mai matin** (horaire différent suivant les groupes). Lors de cette séance, il faudra présenter chacune des 5 fonctionnalités décrites dans le cahier des charges (cf. paragraphe 2b).

#### d. REMISE DU PROJET

Le projet (rapport + sources + exécutable) est à remettre au plus tard le **22 mai** à 23:59:59. Il devra être déposé sur le portail, dans l'espace de dépôt correspondant à votre groupe de TD, le nom du fichier déposé comportant les noms des auteurs du projet et le nom du groupe de TD.

Ex: Nom1-Nom2-TDA.zip

#### e. ORGANISATION

La réalisation du projet doit se faire impérativement en binômes. Lorsque le nombre d'étudiants d'un groupe de TD est impair, un étudiant doit travailler seul (on ne peut avoir qu'un seul étudiant travaillant seul). La constitution des binômes est laissée au choix des étudiants.

5 séances de TD sont prévues pour apporter une aide au développement du projet, mais il est indispensable de travailler également en dehors de ces séances. Les séances sont composées de :

- 2 séances de 3h pour la mise en place des éléments d'interface graphique 3D (OpenGL).
- 2 séances de 3h pour la mise en place de l'interaction avec l'utilisateur par l'intermédiaire de la WebCam (OpenCV).
- 1 séance de finalisation de l'application.

#### f. PENALITES

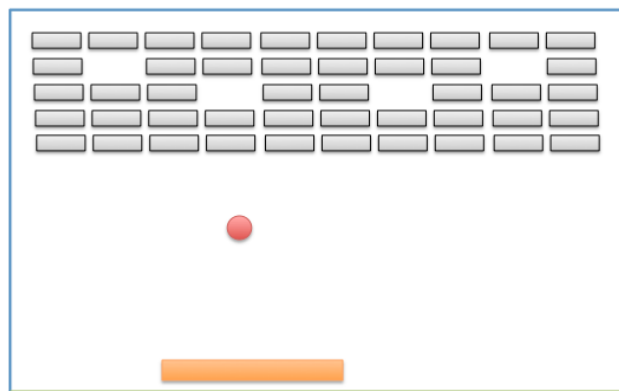
Si le projet est rendu en retard ou s'il manque des documents, une pénalité de 2 points par jour de retard sera appliquée (10% de la note).

## 2. CAHIER DES CHARGES DE L'APPLICATION

### a. DESCRIPTION GENERALE

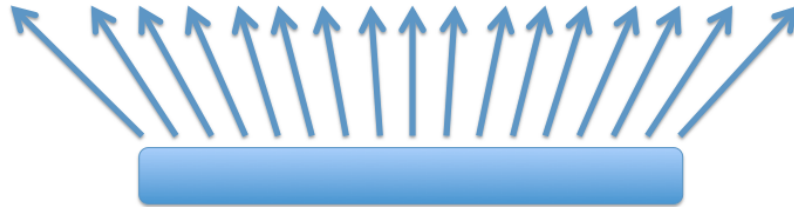
Vous devez réaliser un jeu de casse briques qui sera contrôlé avec une WebCam. Le jeu est constitué des éléments suivants :

- Une boule qui rebondit sur les murs et sur le palet, et qui détruit des briques (en rouge sur l'image ci-dessous).
- Des briques placées les unes à côté des autres (en gris).
- Un palet qui permet de diriger la boule (en orange).
- 3 murs sur lesquels rebondit la boule (en bleu).
- 1 mur, en dessous du palet, qui détruit la boule (en vert).



**Le jeu se déroule en 2 dimensions, même si les briques, le palet, la boule et les murs ont une représentation 3D.** Le jeu aura une dimension de 10 briques de larges sur 12 de profondeur. Les briques devront avoir un rapport d'un pour trois.

Les mouvements du palet sont contrôlés par la WebCam en détectant le déplacement de la main du joueur. L'orientation de la boule, lors d'un rebond sur le palet, sera fonction du point d'impact sur le palet, tel qu'illustré sur l'image suivante.



Au début de la partie, trois boules sont disponibles. Si les boules vont frapper le mur derrière le palet, elles sont détruites et le nombre de boules disponibles diminue. Lorsque la dernière boule est détruite la partie est terminée.

Les points sont incrémentés à chaque fois qu'une brique est détruite. Vous pouvez avoir des briques bonus qui donnent plus de points que d'autres. Lorsque l'ensemble des briques d'un écran est détruit, l'écran est repeuplé à nouveau avec de nouvelles briques (dont vous avez la liberté de disposition). Pour le premier écran, la vitesse de la boule est relativement lente. Au gré de l'avancement dans les écrans, la vitesse de la boule est accélérée.

#### **b. FONCTIONNALITES A IMPLEMENTER**

Les fonctionnalités suivantes doivent être implémentées dans le projet :

- Déplacement du palet commandé par le déplacement de la main du joueur à partir de la WebCam.
- Rebond de la boule sur les murs, sur le palet et sur les briques.
- Destruction des briques (disparition) lorsqu'elles sont touchées par la boule.
- Rebondissement sur le palet et contrôle de la direction de la boule en fonction du point d'impact sur le palet.
- Décompte des boules utilisées et contrôle de la fin de partie.

Les fonctionnalités suivantes constitueront un bonus sur la notation du projet :

- Génération aléatoire d'un nouveau niveau avec une vitesse supérieure de la boule par rapport au niveau précédent.
- Calcul des points.
- Choix de la taille du palet par le joueur
- Sauvegarde du score/nom de joueur et affichage du classement

### c. PRECISIONS SUR LES ELEMENTS GRAPHIQUES 3D

Le jeu devra être dessiné avec OpenGL, selon les principes vus lors des TP.

### d. PRECISIONS SUR L'ARCHITECTURE DE VOTRE PROJET

Une architecture de données adaptée devra être utilisée pour la gestion des objets. Il vous est conseillé de créer des objets briques, palets et murs dérivés d'une classe virtuelle.

### e. PRECISIONS SUR L'INTERACTION AVEC LA WEBCAM

Il s'agit de mettre en place des fonctionnalités d'interaction entre l'utilisateur et l'application par l'intermédiaire d'une *WebCam*. L'interface proposera une zone d'interaction présentant une partie du champ de la caméra. L'utilisateur devra commander le déplacement du palet en réalisant des gestes de balayage (défilement) devant la zone active de la caméra (figure 1). Deux gestes de base seront possibles :

1. balayage horizontal (droite ou gauche)
2. arrêt (recouvrement de la zone par la main)

Pour différencier les gestes 1 et 2, il faudra prendre en compte la direction du geste et sa vitesse. L'objet devra se déplacer dans la bonne direction, à une vitesse soit fixe, soit proportionnelle à la vitesse du balayage. Un effet de freinage pourra optionnellement être mis en place pour faire ralentir l'objet jusqu'à l'arrêt comme lors d'un défilement généré par une interface tactile (smartphone).

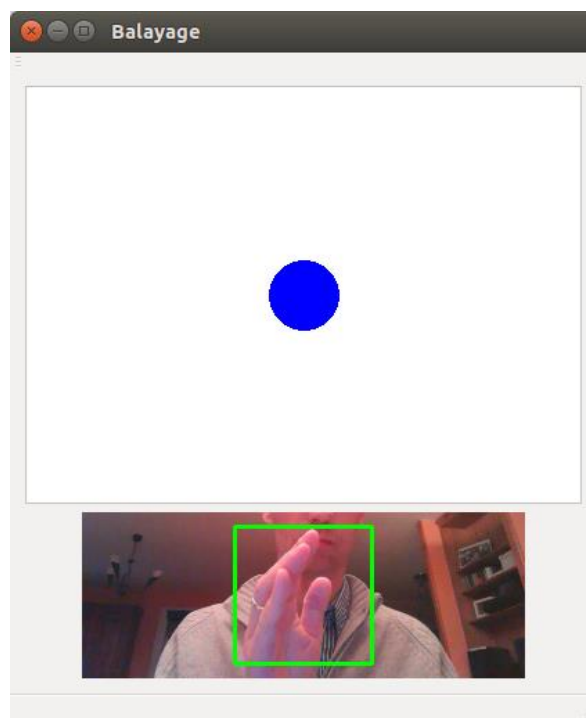


Figure 1 : Mise au point de l'interface par *WebCam*.

**Dans un premier temps, la gestion de l'interaction webcam/utilisateur devra se faire dans une fenêtre séparée de la fenêtre de rendu OpenGL.**

Les étapes à franchir sont les suivantes :

1. Installer la bibliothèque *OpenCV* dans son environnement de développement.
2. Mettre en place une application pour la mise au point de l'interface. Cette application aura une zone de captation du geste et une zone de dessin de l'objet en mouvement (ici une sphère pour la mise au point). On pourra partir de l'exemple *TestWebCamQt* fourni.
3. Etudier la fonction *matchTemplate* et son application à la détection de mouvement dans l'exemple *TestDetectMotion* et sur la page de documentation [ici](#).
4. Ajouter la détection des gestes et la visualisation du mouvement de la sphère à votre application. On pourra passer par les sous-étapes suivantes :
  - 4.1. Détection du geste de balayage horizontal.
  - 4.2. Détection du geste d'arrêt.
  - 4.3. Optimisation des paramètres pour augmenter la fiabilité et la fluidité (on pourra utiliser les fonctions de multi-threading de Qt comme *QtConcurrent::run* et la classe *QFutureWatcher* au besoin).