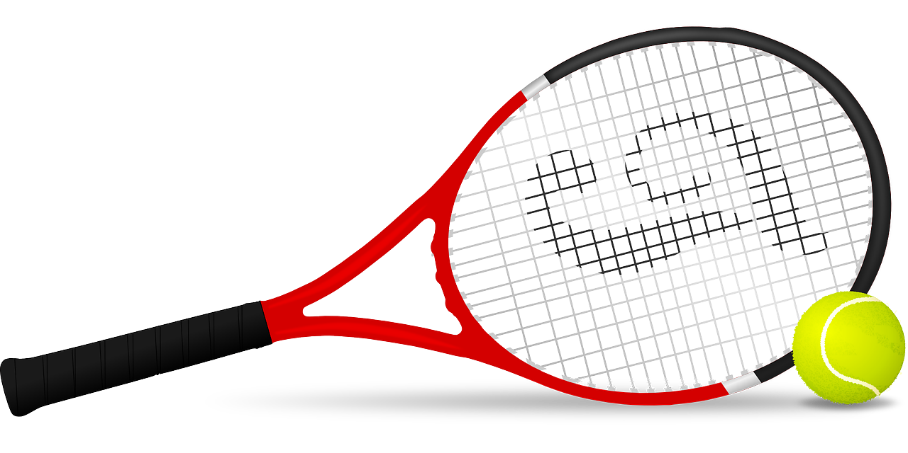
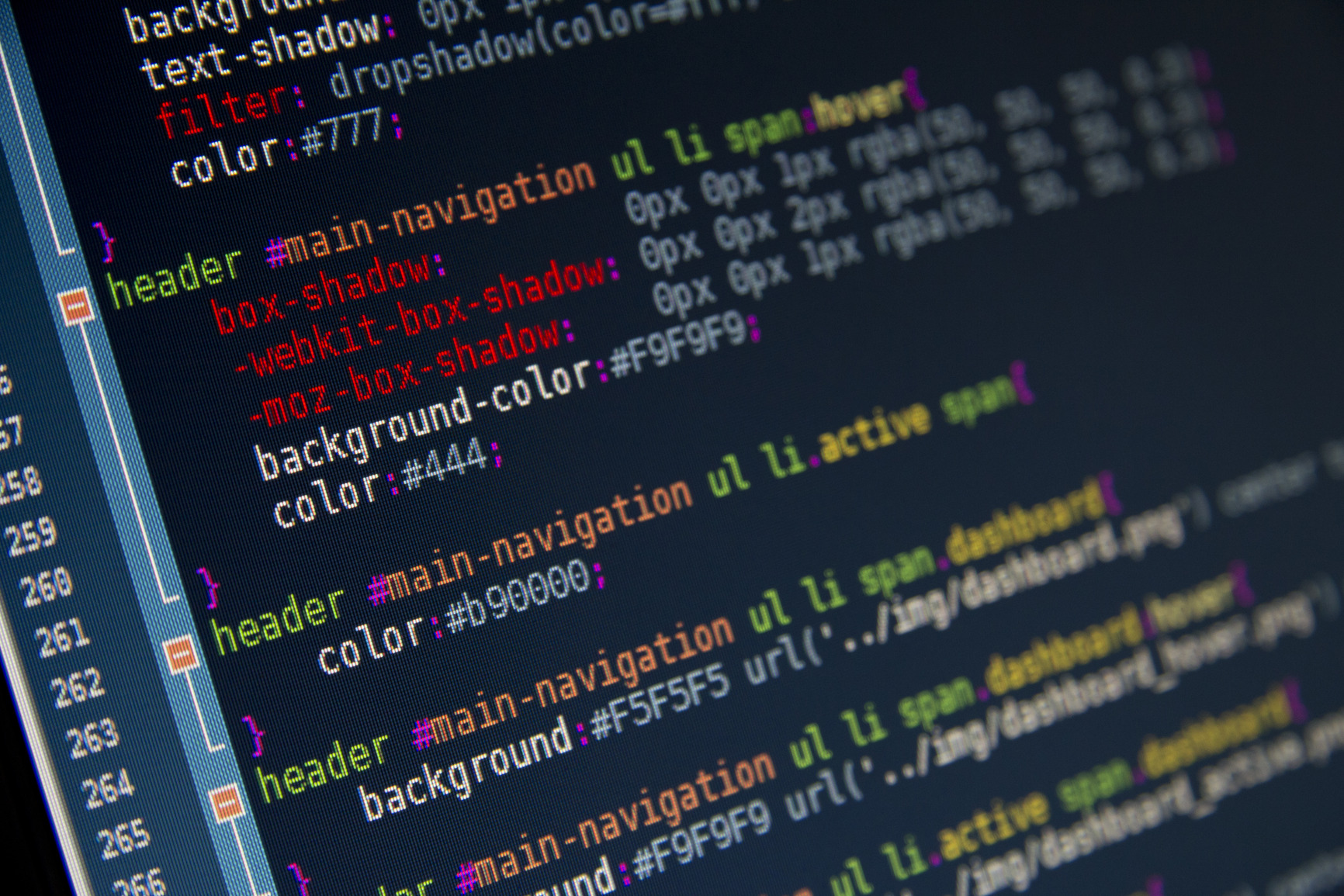
[](http://www.google.ch/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0CAcQjRxqFQoTCI7N4OLQiskCFYJOFAod4bQCNg&url=http%3A%2F%2Fwww.journaldugeek.com%2F2015%2F04%2F03%2Fcest-la-fin-pour-kinect-pour-windows%2F&psig=AFQjCNED-DTTnD3oQ4Egb0uIUXxGvCi16w&ust=1447408753625706) [](http://www.google.ch/url?sa=i&source=imgres&cd=&cad=rja&uact=8&ved=0CAkQjRwwAGoVChMI4u_PjsyKyQIVxmEPCh3Z1gae&url=https%3A%2F%2Fcodegeekz.com%2Fonline-javascript-editors-for-rapid-development%2F&psig=AFQjCNEKqAr_53b7dwaH_h2_RDPxVrjdHQ&ust=1447407731407638)

Réalisé par : Samuel Bétrisey

Professeur : Thibault Schönmann

Virtual Squash 2

Table des matières

[2 Cahier des charges 3](#_Toc435109271)

[2.1 Description générale 3](#_Toc435109272)

[2.2 Infrastructure 3](#_Toc435109273)

[2.2.1 Hardware 3](#_Toc435109274)

[2.2.2 Software 3](#_Toc435109275)

[2.2.3 Finance 3](#_Toc435109276)

[2.3 Détails sur la réalisation 3](#_Toc435109277)

[2.3.1 Besoins du client 3](#_Toc435109278)

[2.4 Contraintes et risques 3](#_Toc435109279)

[3 Planification 4](#_Toc435109280)

[3.1 Initiale 4](#_Toc435109281)

[3.2 Réelle 5](#_Toc435109282)

[3.3 Remarque sur la planification 6](#_Toc435109283)

[4 Documentation 6](#_Toc435109284)

[4.1 Organisation du code 6](#_Toc435109285)

[4.1.1 Classe Main 6](#_Toc435109286)

[4.1.2 Classe Player 6](#_Toc435109287)

[4.1.3 Classe Balle 6](#_Toc435109288)

[4.1.4 Classe Game 6](#_Toc435109289)

[4.2 Fonctionnalités 6](#_Toc435109290)

[4.2.1 Multi-Joueurs 6](#_Toc435109291)

[4.2.2 Taille de la balle 7](#_Toc435109292)

[4.3 Réglages de constantes possibles 7](#_Toc435109293)

[4.3.1 LONGUEUR\_ECRAN et LARGEUR\_ECRAN (Main.java) 7](#_Toc435109294)

[4.3.2 NOMBRE\_JOUEURS (Main.java) 7](#_Toc435109295)

[4.3.3 FACTEUR\_HORIZONTAL et FACTEUR\_VERTICAL (Main.java) 7](#_Toc435109296)

[4.3.4 TIMEOUT (Game.java) 7](#_Toc435109297)

[4.3.5 WIDTH et HEIGHT (Player.java) 7](#_Toc435109298)

[5 Améliorations possibles 7](#_Toc435109299)

[5.1 Multi-joueurs en réseau 7](#_Toc435109300)

[5.2 Utilisation d’un moteur 3D et physique 7](#_Toc435109301)

[6 Fichiers annexe 7](#_Toc435109302)

# Cahier des charges

## Description générale

Amélioration d’un jeu précédemment développé dans l’école. Il s’agit d’un jeu de squash qui sera jouable à deux grâce à la Kinect. Chaque joueur utilise sa main comme raquette et dois faire rebondir la balle avant qu’elle ne touche le mur derrière lui.

J’essaierai pendant une journée de voir si j’arrive à créer le jeu dans le navigateur en JavaScript. Si j’arrive sans trop de problèmes, je continuerai comme ça sinon le jeu sera développé en Java en utilisant la bibliothèque KinectPV2.

## Infrastructure

### Hardware

* Un ordinateur avec Windows 8 ou plus récent
* Un capteur Kinect 2.0
* Un adaptateur Kinect pour Windows

### Software

* IntelliJ IDEA 15 (<https://www.jetbrains.com/idea/>), licence gratuite pour les étudiants (IDE Java)
* KinectPV2 0.7.5 (<https://github.com/ThomasLengeling/KinectPV2>), open source ([licence MIT](https://fr.wikipedia.org/wiki/Licence_MIT))
  + Librairie Java permettant de récupérer le squelette depuis la Kinect
  + Intègre aussi Processing pour l’affichage graphique en temps réel.

### Finance

* [Un capteur Kinect 2.0](https://www.brack.ch/microsoft-xboxone-kinect-sensor-339740) 149.00 CHF
* [Un adaptateur Kinect pour Windows](https://www.brack.ch/microsoft-kinect-adapter-fuer-339741) 59.00 CHF

**Total : 208.00 CHF**

## Détails sur la réalisation

### Besoins du client

* Créer un squash jouable à la Kinect en se basant sur le projet crée il y a 2 ans
  + Simulation de la balle qui rebondit sur les murs, le plafond, le sol et au contact de la raquette
* Corriger un bug qui rendait la raquette beaucoup trop grande
  + La taille de la raquette doit correspondre à celle de la main.
* Doit fonctionner avec la Kinect V2
* Possibilité de jouer à 1 ou 2 joueurs
  + A 1 joueur compter le nombre de rebonds, augmentation de la vitesse de la balle tous les 10 rebonds
  + A 2 joueurs compter un point à chaque faute de l'adversaire, jusqu'à 21 avec 2 points d'écart

## Contraintes et risques

* Pour l’instant je n’ai pas encore reçu le nouveau modèle de la Kinect donc je ne l’ai pas encore testé. Mais il devrait être compatible avec KinectPV2.
* J’aimerai essayer de faire le jeu dans le navigateur mais si c’est trop compliqué, je le ferai en Java.

# Planification

## Initiale

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **27.08** | | **03.09** | | **10.09** | | **17.09** | | **24.09** | | **01.10** | | **08.10** | | **15.10** | | **29.10** | | **05.11** | | **12.11** | | **19.11** | |
| **Méta-tâches** | **Tâches** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** |
| **Cahier des charges** | Détails sur les tâches à réaliser |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Planification** | Planifier le travail |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Recherche** | Recherche sur les différentes possibilités de réalisation du projet, information sur les librairies utilisées |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Analyse du code** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Réception de la Kinect** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Test** | Déterminer si le jeu pourrait être développé dans le navigateur en JavaScript |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Développement** | Suivi de la main |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Affichage de la raquette |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Gestion des déplacements de la balle en se basant sur le code existant |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Faire rebondir la balle avec la main dans la bonne direction |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Deuxième joueur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Gestion de la partie** | Scores, tour des joueurs |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Ecran affichant le gagnant |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Finalisation** | Design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Ajout des dernières fonctionnalités, test et correction d’éventuels bugs |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Documentation** | Faire une documentation complète |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Préparation** | Préparer la présentation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Présentation** | Présentation du projet |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Réelle

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **27.08** | | **03.09** | | **10.09** | | **17.09** | | **24.09** | | **01.10** | | **08.10** | | **15.10** | | **29.10** | | **05.11** | | **12.11** | | **19.11** | |
| **Méta-tâches** | **Tâches** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** | **AM** | **PM** |
| **Cahier des charges** | Détails sur les tâches à réaliser |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Planification** | Planifier le travail |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Recherche** | Recherche sur les différentes possibilités de réalisation du projet, information sur les librairies utilisées |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Analyse du code** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Réception de la Kinect** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Test** | Déterminer si le jeu pourrait être développé dans le navigateur en JavaScript |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Développement** | Suivi de la main |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Affichage de la raquette |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Gestion des déplacements de la balle en se basant sur le code existant |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Faire rebondir la balle avec la main dans la bonne direction |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Deuxième joueur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Gestion de la partie** | Scores, tour des joueurs |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Ecran affichant le gagnant |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Finalisation** | Design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Ajout des dernières fonctionnalités, test et correction d’éventuels bugs |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Documentation** | Faire une documentation complète |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Préparation** | Préparer la présentation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Présentation** | Présentation du projet |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Remarque sur la planification

La livraison de la Kinect a eu une semaine de retard, donc j’ai pris un peu de retard sur la planification initiale et je n’ai pas essayé de développer le jeu en JavaScript dans le navigateur. J’ai tout de même fais quelques recherches.

Une autre tâche qui a pris un peu plus long que prévu est l’affichage de la raquette. J’ai passé beaucoup de temps pour afficher la raquette correctement. (Trigonométrie pour calculer les angles de la raquette)

Malgré ces contretemps, j’ai réussi à remplir tous les objectifs du cahier des charges.

# Documentation

## Organisation du code

### Classe Main

Classe de la fenêtre principale. Elle fait le lien entre la Kinect et les classes. En cas d’utilisation d’un autre appareil, il y aura seulement cette classe à modifier.

* Mets à jour les positions des joueurs (classe Player)
* Instancie les classes Ball et Game
* Affiche l’interface graphique (scores, vainqueur, bouton recommencer)

### Classe Player

* Stocke les positions du joueur et son score
* Affiche la raquette
* Calcule la direction de la frappe

### Classe Balle

Permet d’afficher une balle qui rebondit contre les murs.

On peut la faire rebondir dans une direction avec la méthode bounce(direction) et la faire accélérer avec la méthode accelerate(facteur).

On peut changer sa couleur (attribut color).

### Classe Game

* Gère les tours des joueurs
* Calcule des scores
* Vérifie si le joueur frappe la balle et fait bouger la balle en conséquence
* Désigne le vainqueur et si la partie est finie
* Choisi le mode de jeu en fonction du nombre de joueurs détectés

## Fonctionnalités

### Multi-Joueurs

1 joueur : Gagne 1 point à chaque rebond de la balle. S’il n’arrive pas à la renvoyer assez vite [(voir timeout)](#_TIMEOUT_(Game.java)) son score est remis à zéro. Son meilleur score est affiché.

2 joueurs : Les joueurs font rebondir la balle l’un après l’autre. Si un joueur n’arrive pas à la frapper à son tour, son adversaire marque 1 point. La partie se termine à 21 points avec 2 points d’écart.

### Simulation des rebonds – murs et fond

C’est la seule partie pour laquelle j’ai gardé pas mal de code du projet fait il y a 2 ans.

### Simulation des rebonds – raquette

Lorsque le joueur frappe la balle, elle part dans la direction du mouvement de sa main.

Pour donner un effet de rebond plus réaliste, j’ai calculé la direction de la frappe avec les 15 dernières positions, normalisé ce vecteur et multiplié par la norme du vecteur du déplacement de la balle pour conserver la vitesse de la balle.

Donc la balle prend la direction de la frappe tout en conservant sa vitesse.

### Taille de la balle

## Réglages de constantes possibles

### LONGUEUR\_ECRAN et LARGEUR\_ECRAN (Main.java)

Résolution de l’écran

### NOMBRE\_JOUEURS (Main.java)

Permet de limiter le nombre de joueurs

Mode de jeu selon [Multi-Joueurs](#_Multi-Joueurs)

### FACTEUR\_HORIZONTAL et FACTEUR\_VERTICAL (Main.java)

Utilisés lors de la conversion des données de la Kinect (mètres 🡪 pixels).

Pour que la main soit alignée à la raquette virtuelle, il faut faire varier ces valeurs qui dépendent de la taille de l’écran et de la distance du joueur.

### TIMEOUT (Game.java)

Temps que le joueur a pour frapper la balle lorsqu'elle arrive vers lui.

### WIDTH et HEIGHT (Player.java)

Taille de la raquette à 1 mètre. Elle variera automatiquement si la main se rapproche ou s’éloigne.

# Améliorations possibles

## Multi-joueurs en réseau

Un mode multi-joueurs en réseau serait tout à fait réalisable avec quelques jours de plus et une deuxième Kinect.

Connexion socket entre 2 PCs :

* Le PC « serveur » calcule les rebonds, scores et un joueur joue sur celui-ci
* Le PC « client » envoie la position de son joueur au serveur (objet Player sérialisé) et récupère la position de la balle, le score etc…

## Utilisation d’un moteur 3D et physique

Pour une simulation plus réaliste des rebonds, de la gravité et de la profondeur

Ça pourrait prendre plus de temps car il faudrait recoder pas mal de chose. Pour utiliser un espace 3D au lieu de l’espace 3D simulé en 2D.

# Fichiers annexe

* Journal de bord