

Option IRA – Réalité Augmentée
Mini-projet : Jeu d'échec augmenté

Timothé Rios - Nicolas Venot

mai 2021

Table des matières

1	Environnement logiciel	3
1.1	Unity	3
1.2	Vuforia	3
2	Choix de la technique de recalage	3
2.1	Premier essai	3
2.2	Deuxième essai	3
2.3	Choix final	3
3	Développement	5
3.1	Échiquier	5
3.2	Pièces	5
4	Interface	5
4.1	Interface de jeu	5
4.2	Fin de partie	5

1 Environnement logiciel

1.1 Unity

Unity 2021.1.13f1

1.2 Vuforia

Vuforia 9.8.5

2 Choix de la technique de recalage

2.1 Premier essai

Notre idée de départ était d'utiliser un véritable jeu d'échec, avec ses pièces et son plateau, et de les faire reconnaître par *Unity* et *Vuforia*. Ceci nous aurait permis d'afficher les déplacements possibles de pièces, les position d'échec ou de mat, les possibilités de roque, ... tout en gardant les sensations physiques d'un véritable jeu de société. Cependant, il nous est vite apparu que les pièces d'un jeu d'échec n'avait pas assez de *feature points*. En conséquence, la détection des pièces était trop peu efficace pour notre projet. De plus, la taille de l'échiquier et ses motifs répétitifs rendaient aussi impossible sa reconnaissance. Nous avons donc décidé d'abandonner cette méthode.

2.2 Deuxième essai

La démarche que nous avons testée en deuxième ne nécessitait plus de pièces à proprement parler. Chaque pièce du jeu d'échec était représentée par un petit *AR code* que nous pouvions ensuite déplacer autour d'un plus grand *AR code* correspondant au plateau. Cependant, la superposition des *AR codes* et des problèmes quant au passage de l'environnement réel 3D à la scène 2D faisaient perdre trop facilement le suivi des pièces et contrariaient le positionnement des objets virtuels (cases du plateau, mise en surbrillance des coups possibles,...). Nous avons décidé d'abandonner aussi cette méthode.

2.3 Choix final

La méthode que nous avons finalement choisie ne comporte qu'un seul *AR code*. Celui-ci permet de faire apparaître à la fois le plateau de jeu et les pièces d'échec. Les déplacements doivent donc se faire avec l'appareil de réalité augmentée (téléphone portable ou ordinateur). Cette méthode permet d'éviter tout problème de superposition des *AR codes* et le seul que nous utilisons est facilement reconnaissable par *Unity* et *Vuforia*.



FIGURE 1 – AR code utilisé pour le jeu d'échec

3 Développement

3.1 Échiquier

Afin de gérer les déplacements des pièces, nous avons utilisé une grille des cases de l'échiquier que nous superposons à ce dernier. Cette grille nous permet de calculer la position des centres des cases afin d'y envoyer la pièce que nous souhaitons bouger. C'est aussi cette grille qui nous permet de mettre en surbrillance les cases pertinentes pour le coup que nous voulons jouer.

3.2 Pièces

Chaque pièce virtuelle se voit attacher un script qui permet de gérer et d'identifier sa position au sein de l'échiquier. Ce script comporte aussi d'autres informations relatifs à la pièce, comme le joueur auquel elle appartient ou encore les pièces ennemies qui la menacent, s'il y en a. chaque pièce a un script qui identifie leur position, le joueur, et la menace possible

4 Interface

4.1 Interface de jeu

Au cours de la partie, sont affichées sur l'écran différentes informations. Cet affichage fixe nous informe du joueur dont c'est le tour de jouer et des mises en échec s'il y en a. Quand un joueur sélectionne une pièce, des informations supplémentaires apparaissent : les cases sur lesquelles la pièce sélectionnée peut être déplacée sont surlignées en vert, sauf si elles contiennent une pièce ennemie : elles sont dans ce cas surlignées en rouge.

4.2 Fin de partie

minimaliste, efficace,