

Rapport Projet interface **3D Irrlicht**

Killian Janod et Cédric Rihier

L3 Informatique C.E.R.I d'Avignon 2009/2010

Table des matières

I Les objectifs atteints.....	3
I.1 Le mode jeu un contre un :.....	3
I.2 Le mode joueur contre ordinateur :.....	3
I.3 Les évènements sont gérés via les touches du clavier.....	3
I.4 Menu :.....	3
I.5 Gestion des scores:.....	3
I.6 Le petit plus :.....	3
II Les fonctionnalités opérationnelles.....	3
II.1 IA, menu et affichages.....	3
II.2 les différents mouvements :	4
II.3 Scores, objets non sélectionnables.....	4
II.4 Quelques spécifications techniques.....	4
II.4.a Utilisation autant que possible de l'orienté objet : trois classes pour plus de modularité :	4
II.4.b création de plusieurs nodes pour représenter les différents objets :.....	4
III autres.....	4
IV Les problèmes non résolus.....	6
IV.1 Quelques ratés.....	6
IV.2 Les objectifs non atteints.....	6
V Nos difficultés.....	6
VI Synthèse.....	7

I Les objectifs atteints

I.1 Le mode jeu un contre un :

- placement des pions avec mouvements de translations
- alternance joueur 1/joueur 2
- test de victoire avec les méthodes du projet semestre 1

I.2 Le mode joueur contre ordinateur :

- placement des pions par l'ordinateur : minmax implémenté au premier semestre
- alternance ordinateur/joueur
- test de victoire avec les méthodes du projet semestre 1
- plusieurs niveaux de difficultés disponibles

I.3 Les événements sont gérés via les touches du clavier

I.4 Menu :

- choix du mode d'affichage : fenêtre ou plein écran
- choix du mode de jeu : un joueur ou deux joueurs
- pour le mode un joueur : choix de trois niveaux de difficultés
- les différentes sélections sont gérées avec la souris

I.5 Gestion des scores:

- scores joueurs et ordinateur : chaque partie ajoute un point à celui qui la gagne

I.6 Le petit plus :

- possibilité de lancer une rotation du plateau pour améliorer la visibilité pendant le jeu
- une touche de réinitialisation pour la rotation : le plateau se remet en position initiale

II Les fonctionnalités opérationnelles

II.1 IA, menu et affichages

- la "fusion" entre le jeu "Core" du premier semestre et le moteur 3d
- la gestion du menu avec les différentes options qui se transmettent au jeu lors de son lancement
- la gestion des différents affichages pour informer le joueur de l'état de la partie

II.2 les différents mouvements :

- halo de lumière pour la sélection d'un pion dans la pioche et pour la sélection d'une case libre sur le plateau

- déplacement d'un pion de la pioche vers le centre de l'aire de jeu
- déplacement d'un pion du centre vers la case sélectionnée par un joueur
- rotation du plateau sur son axe central : vers la gauche et vers la droite
- recentrage du plateau pour le remettre en position initiale

II.3 Scores, objets non sélectionnables

- la gestion des cases qui ne sont plus disponibles et des pions qui ne sont plus dans la pioche : pas de sélection possible de ces éléments
- le halo de sélection qui vient par défaut se mettre sur un pion disponible dans la pioche ou sur une case disponible du plateau
- gestion du score dans tous les modes de jeu, avec affichage et possibilité de faire plusieurs parties

II.4 Quelques spécifications techniques

II.4.a Utilisation autant que possible de l'orienté objet : trois classes pour plus de modularité :

- la classe Moteur3d qui gère l'affichage du jeu : on peut jouer sans implémentation du jeu "Core", mais les tests de victoire ne sont alors pas gérés
- la classe Menu3d indépendante du jeu qui permet de capter les différentes options laissées aux joueurs
- la classe MyEventReceiver qui permet de gérer les événements (touches clavier pour le jeu et souris pour le menu). Cette classe est utilisée dans les classes Menu3d et Moteur3d

II.4.b création de plusieurs nodes pour représenter les différents objets :

- le plateau
- la pioche
- le centre de jeu (EmptySceneNode)
- les pions

II.4.c autres

- Utilisation des méthodes addChild et setParent pour faciliter les différents déplacements des nodes pions entre les différents objets du jeu : pioche, centre, plateau
- Utilisation de différents drapeaux pour gérer correctement les animations lorsque celles-ci sont déclenchées par l'action d'une touche
- un code clairement commenté pour plus de lisibilité
- L'intégration du jeu "Core" a été facilitée par l'existence au préalable de méthodes suffisamment pratiques, mais une difficulté nous a donné du fil à retordre : la gestion de la pioche (cf : ce qui nous a posé problème).
- Nous avons fait nos propre textures pour les pions et le plateau, avec GIMP.

III Les problèmes non résolus

III.1 Quelques ratés

- lorsque la méthode de décision est lancée (pour que l'ordinateur choisisse le meilleur coup à jouer), le temps mis par celle-ci pour donner une réponse provoque un freeze qui coupe l'animation qui suit... Donc, l'effet de translation de cette animation n'est pas visible...le pion se 'téléporte' directement vers la case choisie par l'ordinateur. Ce problème disparaît lorsque la décision se fait plus rapide (soit en avançant dans le jeu, soit en choisissant un niveau de difficulté faible)
- Si l'on fait tourner le plateau (avec l'action rotation), le point de départ de l'animation d'un pion qui va du centre vers une case se déplace aussi. De ce fait, lorsque l'animation se lance, le pion est d'abord 'téléporté' vers ce nouveau point de départ avant que la translation vers la case s'effectue.
- Sous Windows, lorsqu'après avoir choisi les options dans le menu on lance le jeu, la fenêtre du menu ne se ferme pas (le device->drop() n'a pas l'air de fonctionner correctement). Ce problème ne se produit pas sous Linux

III.2 Les objectifs non atteints

- La caméra est fixe, on ne peut donc pas la déplacer avec le clic droit de la souris, ceci n'a pas été pris en compte.
- le jeu se joue entièrement au clavier, il n'y a donc pas de gestion des collisions entre le pointeur de la souris et les objets et donc pas de mise en surbrillance des objets pointés
- nous n'avons pas utilisé de mesh, seulement les objets par défaut que nous propose Irrlicht.
- Graphiquement, l'utilisation de mesh travaillées pour nos pions donnerait un rendu graphique plus agréable.

IV Nos difficultés

- au début, la gestion de l'échelle en fonction de l'objet auquel nos pions étaient rattachés
- L'appel à des fonctions dont on ne connaît pas tous les effets à priori (Device-> drop() sous Windows)
- Les modélisations différentes des objets du noyau et des objets 3d qui ont nécessité l'utilisation d'astuces pour les synchroniser (tableaux de correspondances)

V Synthèse

- Notre programme est une modélisation en trois dimensions du jeu de Qwarto
- Lorsque le programme se lance, un menu apparaît
- Une fois les options de jeu choisies dans le menu, le jeu se lance et les actions sont gérées par action sur les touches du clavier
- Ce jeu peut se jouer en mode classique à deux joueurs
- Il est aussi doté d'une Intelligence Artificielle qui vous permet de jouer contre l'ordinateur avec le choix de plusieurs niveaux de difficulté
- Vous avez la possibilité de faire plusieurs parties d'affilé. Un score indiquant alors le nombre de victoires pour chaque joueur (si mode deux joueurs) ou pour le joueur et l'ordinateur (si mode un joueur) donne l'avancement du jeu.
- Entre chaque partie, il y a un retour au menu pour pouvoir modifier les options de jeu.
- C'est par ce menu que l'on peut quitter le jeu définitivement