

**TP2 – Interactivité**

Par :

**Philippe Henry (111 073 034)**

**Bastien Paul Raymondeau (536 789 681)**

**Équipe #25**

Pour satisfaire les exigences du cours

**Programmation de jeux vidéo**

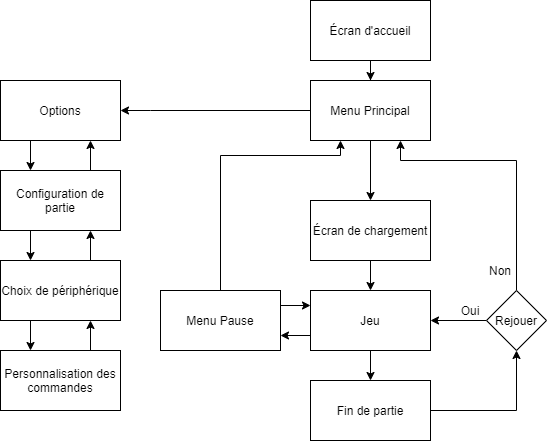
**IFT-2103**

Travail Présenté à

François Chéné

25 novembre 2020

Flot d’application



Au démarrage de l’application, un écran d’accueil apparait quelques secondes, puis l’utilisateur se trouve au menu principal. À partir de cet écran, il peut soit démarrer une partie en solo contre un UI, soit démarrer une partie en multijoueur ou soit aller dans le menu d’options.

Si l’utilisateur choisi une des 2 options pour démarrer une partie, un écran de chargement avec la progression lui est affiché. À la fin de ce chargement, le jeu démarre et l’utilisateur contrôle son avatar et compétitionne soit contre un AI, soit contre un autre joueur. À tout moment pendant le jeu, il est possible de mettre le jeu sur pause. Lorsque le jeu est sur pause, l’utilisateur peut soit continuer la partie, soit quitter et retourner au menu principal. À la fin de la séquence de jeu, l’écran de fin de partie avec le gagnant et les statistiques est affiché et le joueur peut redémarrer une partie ou retourner au menu principal.

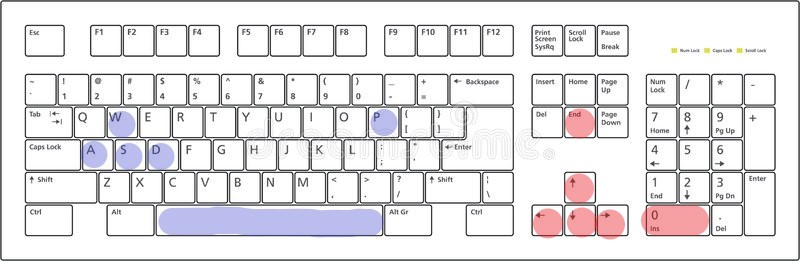
Si l’utilisateur, à partir du menu principal, choisi de modifier les options, une page avec les modifications de la partie (durée maximale, difficulté) lui est présenté. Cette page contient également l’option de modifier les commandes. En sélectionnant ce bouton, l’utilisateur arrive sur un écran où il peut décider, pour chacun des joueurs, le périphérique souhaité. Il peut également appuyer sur Personnalisation qui l’amène sur un écran lui permettant de réassigner les touches selon ce qu’il désire.

Schéma de contrôle des agents

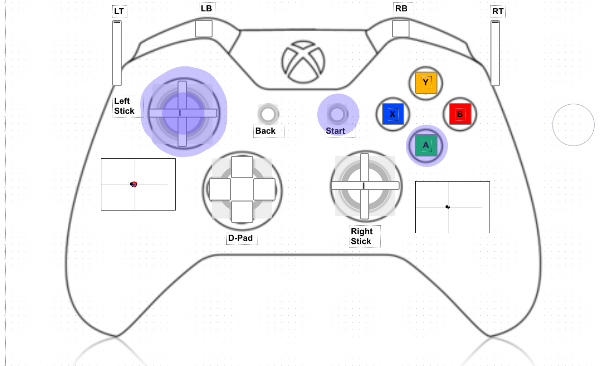
La figure suivante présente la configuration de base sur clavier qui permet aux utilisateurs de bouger l’avatar et de se promener dans les menus (à noter que la souris permet également de se déplacer dans les menus) :

La configuration est présentée sur clavier QWERTY (par défaut), mais un bouton dans l’écran de personnalisation des commandes permet de rapidement changer de disposition vers ASERTY

|  |  |  |
| --- | --- | --- |
| Joueur 1 | Joueur 2 | Action |
| W | Flèche↑ | Mouvement vers l’avant |
| S | Flèche↓ | Mouvement vers l’arrière |
| A | Flèche← | Mouvement vers la gauche |
| D | Flèche→ | Mouvement vers la droite |
| P | End | Pause |
| SPACE | 0 (NPad) | Sauter |



|  |  |
| --- | --- |
| Joueur 1-2 | Action |
| W | Joystick |
| S | Joystick |
| A | Joystick |
| D | Joystick |
| P | Bouton Start |
| SPACE | A |



Intelligence artificielle de l’agent autonome

Notre agent autonome est capable d’aller chercher des pièces dans la scène. Pour cela, il choisit une pièce de manière aléatoire, et applique l’algorithme de recherche de chemin A\* pour la récupérer le plus rapidement possible. L’implantation de cet algorithme dans notre projet est inspiré du vidéo « A\* Pathfinding (E01: algorithm explanation) » par Sebastian Lague[[1]](#endnote-1)

Fonctionnalités supplémentaires

## Personnalisation des méthodes d’entrées (Philippe Henry) :

Les entrées du clavier peuvent être personnalisables pour les 2 joueurs via l’interface suivante qui est dans le flot de menus d’options :



Il est donc possible pour chaque joueur de réassigner l’ensemble des touches au clavier, à savoir la touche pour avancer, pour reculer, pour tourner à droite, pour tourner à gauche, pour sauter et pour mettre le jeu en pause. Des boutons permettent également de revenir rapidement à la configuration QWERTY par default ou simplement d’alterner entre QWERTY et ASERTY pour le claver.

Chaque joueur peut également choisir son type de contrôleur via un menu déroulant dans l’écran affichée si dessous. À noter que les touches des manettes Xbox ne sont pas personnalisables.



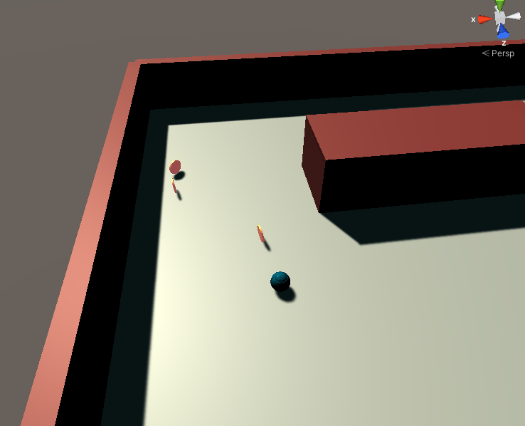
Pour les touches clavier, un gestionnaire de contrôle a été implémenté et permet de garder en mémoire les touches qui ont été sélectionnées lors de la dernière séance de jeu. Ce gestionnaire ce trouve dans le fichier ControlsManager.cs. Le gestionnaire de contrôle a également été adapté afin de permettre l’utilisation de ces touches entrées par l’utilisateur. Fait import à noter, le traitement des nouvelles entrées utilisateurs, à savoir particulièrement le contenu de ControlsManager.cs, a été inspiré du vidéo « How To Build a Custom Input Manager in Unity C# » par Studica News sur Youtube[[2]](#endnote-2).

## Intelligence artificielle (Bastien Paul Raymondeau)

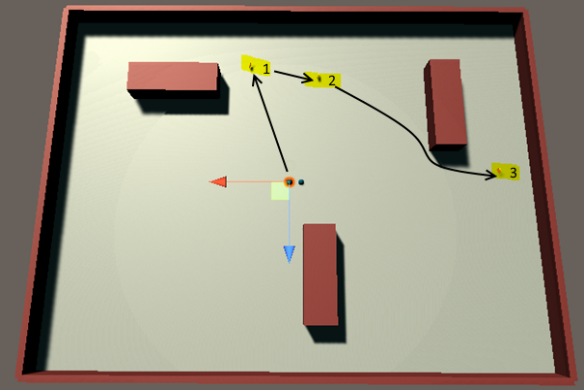
PEAS

Performance

L’approche de sélection aléatoire n’est pas suffisante, en effet les pièces étant dispersées partout dans la scène, l’agent va faire de nombreux allers-retours à cause de son manque de jugement au moment de sélectionner sa pièce cible.



Afin de corriger cela et rendre notre agent plus efficace, nous planifions qu’elle pièce récupérer et dans quel ordre afin de maximiser sa récolte. Par exemple considérons la scène suivante :



L’agent est en (1,0), et nous avons 3 pièces, respectivement en (-34, -2), (-4, -19), (8, -21). L’IA va donc planifier quelles pièces récupérer et dans quel ordre. Le niveau de difficulté va impacter sur le nombre de pièces que l’IA va prendre en compte (Facile = 2 pièces, Difficile = 4 pièces). Dans notre cas, avec un niveau de difficulté difficile, l’IA va récupérer les pièces dans l’ordre suivant : 1, 2, 3. Cela permet de maximiser les gains tout en minimisant le cout de déplacement.

Environnement

* Euclidien
* 2 dimensions (originalement en 3 dimensions, mais l’axe Y pour la hauteur est ignoré par l’IA)
* Jeu en temps réel
* Physique : Loi euclidienne
* Nature des dimensions : discrètes
* Forme des dimensions : Plan
* Grandeur des dimensions : Finis
* Physique : Loi de la physique cinématique et rebond
* Le joueur humain peut pousser l’IA
* Le joueur humain peut prendre des pièces cibles de l’IA

L’espace de jeu est un rectangle de 80 par 60 avec 3 obstacles rectangulaires.

Actuateurs

**Attraper une pièce**

Précondition : Collision avec la pièce ( de déplacement)

Effet : +1 point

Senseurs

L’agent perçoit l’ensemble des pièces sur la scène, leurs coordonnés, et il est également conscient de son propre positionnement. Les obstacles ne sont pas pris en compte lors de la planification d’action, mais l’agent est capable de les contourner le moment venu. L’IA n’est pas au courant de l’intention du joueur, ni de sa position sur le terrain.

## Ressources :

* Les textures utilisée pour les balles ont été téléchargées sur le site 3dtextures.me[[3]](#endnote-3)
* L’image de fond des menus a été prises sur le site pngtree.com[[4]](#endnote-4)

1. https://www.youtube.com/watch?v=-L-WgKMFuhE&list=PLFt\_AvWsXl0cq5Umv3pMC9SPnKjfp9eGW&ab\_channel=SebastianLague [↑](#endnote-ref-1)
2. https://www.youtube.com/watch?v=iSxifRKQKAA [↑](#endnote-ref-2)
3. <https://3dtextures.me/2020/08/21/metal-plate-sci-fi-002/> [↑](#endnote-ref-3)
4. https://pngtree.com/freebackground/wood-background\_582421.html [↑](#endnote-ref-4)