Documentation technique du projet

#567 Motion Controller Puissance 4

Équipe étudiante :

Oussama RECHAH
Said NAOUI
Chris Jordan TENE TALLA
Navidu LOKUPATHIRAGE
Tharidu LOKUPATHIRAGE

Commanditaires : Sylvain BORELLI, Adeline KEOPHILA
Mentor : Vlad VALICA

Table des matières

Introduction

Ce projet, intitulé Motion Control Puissance 4, a été développé dans le cadre du Cap Projet organisé par notre école ESIEA pour l'entreprise Thalès. Le projet a débuté au début du mois d'octobre et il s'est achevé en mars. L'objectif du projet était de développer une version du jeu de société Puissance 4, qui utilise la technologie de capture de mouvement pour permettre aux joueurs de jouer en utilisant des gestes plutôt que des pièces. Le projet a été développé en utilisant Unity et C# et a impliqué la mise en place d'un système de capture de mouvement utilisant une caméra Orbbec Astra Pro 3D.

Cette documentation technique décrit l'architecture système, les composants matériels et logiciels, l'architecture logicielle, le fonctionnement du projet, les exigences de système, l'installation et la configuration, l'utilisation et le dépannage du jeu Puissance 4 Motion Control. Bien que le projet n'ait pas été entièrement achevé, cette documentation fournit une vue d'ensemble détaillée de l'ensemble du processus de développement du jeu, y compris les défis rencontrés et les solutions mises en place pour surmonter ces défis.

I- Architecture système proposée

L'architecture système du Motion Control Puissance 4 est conçue pour permettre aux joueurs de jouer au jeu de société Puissance 4 en utilisant des gestes capturés à l'aide d'une caméra Orbbec Astra Pro 3D. Le système est composé de trois principaux composants : le système de capture de mouvement, l'interface utilisateur et la logique du jeu.

1) Système de capture de mouvement

Le système de capture de mouvement est composé des éléments suivants :

- **Caméra Orbbec Astra** : La caméra Orbbec Astra est utilisée pour capturer les mouvements du joueur. Elle est placée en face du joueur et est connectée à un ordinateur à l'aide d'un câble USB. La caméra utilise la technologie de capture de mouvement pour détecter les mouvements du joueur et les transmettre au logiciel de capture de mouvement.

- **Logiciel de capture de mouvement** : Le logiciel de capture de mouvement est installé sur l'ordinateur et est responsable de la communication avec la caméra Orbbec Astra. Le logiciel utilise l'API de la caméra pour capturer les mouvements du joueur et les transmettre à l'interface utilisateur. Le logiciel est écrit en C# et utilise les bibliothèques de développement Unity.

2) Interface utilisateur

L'interface utilisateur est développée à l'aide d'Unity et elle est chargée de recevoir les données de mouvement capturées par le système de capture de mouvement et de les utiliser pour interagir avec la logique du jeu. L'interface utilisateur est composée de plusieurs éléments :

- Affichage en temps réel des mouvements capturés : L'interface utilisateur affiche en temps réel les mouvements capturés par la caméra Orbbec Astra. Cet affichage permet au joueur de visualiser ses mouvements et de les ajuster en conséquence.

- Boutons de démarrage et d'arrêt de la capture de mouvement : L'interface utilisateur comprend des boutons pour démarrer et arrêter la capture de mouvement. Ces boutons permettent au joueur de contrôler quand la capture de mouvement démarre et s'arrête.

- Écran de jeu : L'écran de jeu est l'endroit où le joueur voit la grille de jeu et les informations de jeu. L'interface utilisateur utilise une grille de jeu 6x7 pour représenter le plateau de Puissance 4.

- Menus de configuration : L'interface utilisateur comprend également des menus de configuration pour personnaliser les paramètres du jeu, tels que la difficulté et le mode de jeu.

3) Logique du jeu

La logique du jeu est la partie la plus importante du système, car elle gère le déroulement du jeu, la vérification des mouvements valides, la détection de la victoire ou de la défaite et l'affichage des résultats du jeu. La logique du jeu est développée en C# et est intégrée à l'interface utilisateur à l'aide de scripts Unity. La logique du jeu comprend les éléments suivants :
Déroulement du jeu : La logique du jeu est responsable du déroulement du jeu, y compris la gestion des tours des joueurs et le placement des jetons.

3.1) Règles du jeu

Ce puissance 4 est une version numérique du jeu de société classique du même nom. Le but du jeu est d'aligner quatre jetons de sa propre couleur horizontalement, verticalement ou en diagonale sur une grille de 6x7 cases. Les joueurs jouent tour à tour en plaçant un jeton dans une colonne. Le jeu se termine lorsque l'un des joueurs réussit à aligner quatre jetons ou lorsque la grille est remplie sans qu'aucun joueur n'ait réussi à aligner quatre jetons.

3.2) Fonctionnalités de l’application

1. La grille de jeu est affichée à l'écran et les joueurs peuvent cliquer sur les colonnes pour placer leurs jetons.
2. La couleur des jetons des joueurs est déterminée en fonction de l'ordre dans lequel ils jouent.
3. Lorsqu'un joueur place un jeton dans une colonne, le jeu vérifie s'il y a une ligne de 4 jetons de la même couleur à l'horizontale, à la verticale ou en diagonale.
4. Si un joueur aligne quatre jetons, le jeu affiche un message de victoire et permet aux joueurs de recommencer ou de quitter le jeu.

5. La logique du jeu vérifie que les mouvements capturés par la caméra sont valides. Elle vérifie que le mouvement est effectué dans une colonne valide (c'est-à-dire qu'il n'y a pas déjà 6 jetons dans la colonne) et qu'il correspond au tour du joueur en cours.

6. Si la grille est remplie sans qu'aucun joueur n'ait réussi à aligner quatre jetons, le jeu affiche un message de match nul et permet aux joueurs de recommencer ou de quitter le jeu.

En résumé, l'architecture système du Motion Control Puissance 4 comprend une caméra Orbbec Astra pour capturer les mouvements des joueurs, un logiciel de capture de mouvement pour communiquer avec la caméra, une interface utilisateur pour afficher les mouvements capturés et gérer le jeu, et une logique de jeu pour gérer le déroulement du jeu, la vérification des mouvements valides et la détection de la victoire ou de la défaite.

II – Architecture logicielle

L’architecture de ce projet est basée sur le framework Unity qui est un moteur de jeu multiplateforme utilisé pour la création de jeux en 2D et 3D.

Le projet utilise principalement les langages de programmation C# et C++ pour l'implémentation de la logique du jeu et pour la capture de mouvement.

L'architecture logicielle du projet peut être divisée en trois couches principales : la couche de présentation, la couche de logique de jeu et la couche de capture de mouvement.

La couche de présentation utilise Unity pour créer l'interface utilisateur du jeu, y compris l'affichage de la grille de jeu, des jetons et des résultats. Cette couche est principalement responsable de l'affichage des données et de l'interaction avec les joueurs.

La couche de logique de jeu est responsable de la gestion des règles du jeu, y compris la vérification des mouvements valides, la gestion des tours des joueurs et la détermination du gagnant. Cette couche utilise principalement C# pour l'implémentation de la logique du jeu.

La couche de capture de mouvement est responsable de la capture des mouvements des joueurs à l'aide de la caméra Orbbec Astra et de leur traitement à l'aide de la bibliothèque de traitement d'image NiTE. Cette couche utilise principalement C++ pour la capture de mouvement et l'utilisation de l'API d'Orbbec Astra.

Toutes ces couches sont connectées et communiquent entre elles via des interfaces de programmation d'application (API). Les données de mouvement capturées par la couche de capture de mouvement sont envoyées à la couche de logique de jeu via une API, où elles sont traitées pour déterminer les mouvements des joueurs et vérifier s'ils sont valides. Les résultats de la partie sont ensuite envoyés à la couche de présentation via une API, où ils sont affichés pour les joueurs.

En résumé, l'architecture logicielle du Motion Control Puissance 4 est basée sur un modèle de couches où les différentes couches sont responsables de tâches spécifiques. Les couches sont connectées et communiquent entre elles via des API pour fournir une expérience de jeu intégrée pour les joueurs.

III- Technologies utilisées dans le cadre de ce projet

- **Unity** : C'est un moteur de jeu multiplateforme qui est utilisé pour créer le jeu Puissance 4. Il fournit une interface de développement visuel qui permet aux développeurs de créer des graphiques en 2D et en 3D, de gérer l'animation, de configurer le physique du jeu et de créer des scènes pour le jeu.

- **Langage de programmation** : Puissance 4 a été développé en C#, qui est un langage de programmation orienté objet utilisé pour développer des applications Windows et des jeux Unity.

- **Git** : C'est un logiciel de gestion de version de fichier.

- **Pilotes de la caméra** : Les pilotes de la caméra Orbbec Astra sont nécessaires pour permettre à l'ordinateur de reconnaître la caméra et de recevoir les données de capture de mouvement. Les pilotes peuvent être téléchargés depuis le site Web d'Orbbec et doivent être installés sur l'ordinateur.

La configuration de ces logiciels nécessite l'installation d'Unity et des pilotes de la caméra Orbbec Astra sur l'ordinateur. La configuration de NiTE et d'OpenCV dépend de leur utilisation spécifique dans le projet. Des bibliothèques supplémentaires peuvent également être nécessaires pour gérer les entrées et sorties audio du jeu.

IV – Relations entre les entités principales du projet

Le fonctionnement du projet est divisé en plusieurs étapes:

Capturer les données de la caméra : La caméra Orbbec Astra est utilisée pour capturer les mouvements des joueurs. Les données sont ensuite transmises au logiciel de traitement d'image pour être analysées.

Traitement d'image : Les données de la caméra sont analysées à l'aide d'une bibliothèque de traitement d'image telle que OpenCV pour détecter les mouvements des joueurs. Les mouvements sont ensuite traduits en actions de jeu.

Logique de jeu : La logique de jeu est responsable du déroulement du jeu, y compris la gestion des tours des joueurs et le placement des jetons. Elle vérifie également que les mouvements des joueurs sont valides et met à jour le plateau de jeu en conséquence.

Affichage des résultats : Les résultats du jeu sont affichés à l'écran, y compris le gagnant du jeu et le plateau de jeu final.

En résumé, le projet fonctionne en capturant les mouvements des joueurs à l'aide de la caméra Orbbec Astra, en les traitant à l'aide d'une bibliothèque de traitement d'image, en appliquant la logique de jeu pour déterminer les actions à effectuer, et en affichant les résultats à l'écran.

V- Déroulement du projet

La conception du projet peut être divisée en trois parties, à savoir la capture du mouvement, la logique de jeu.

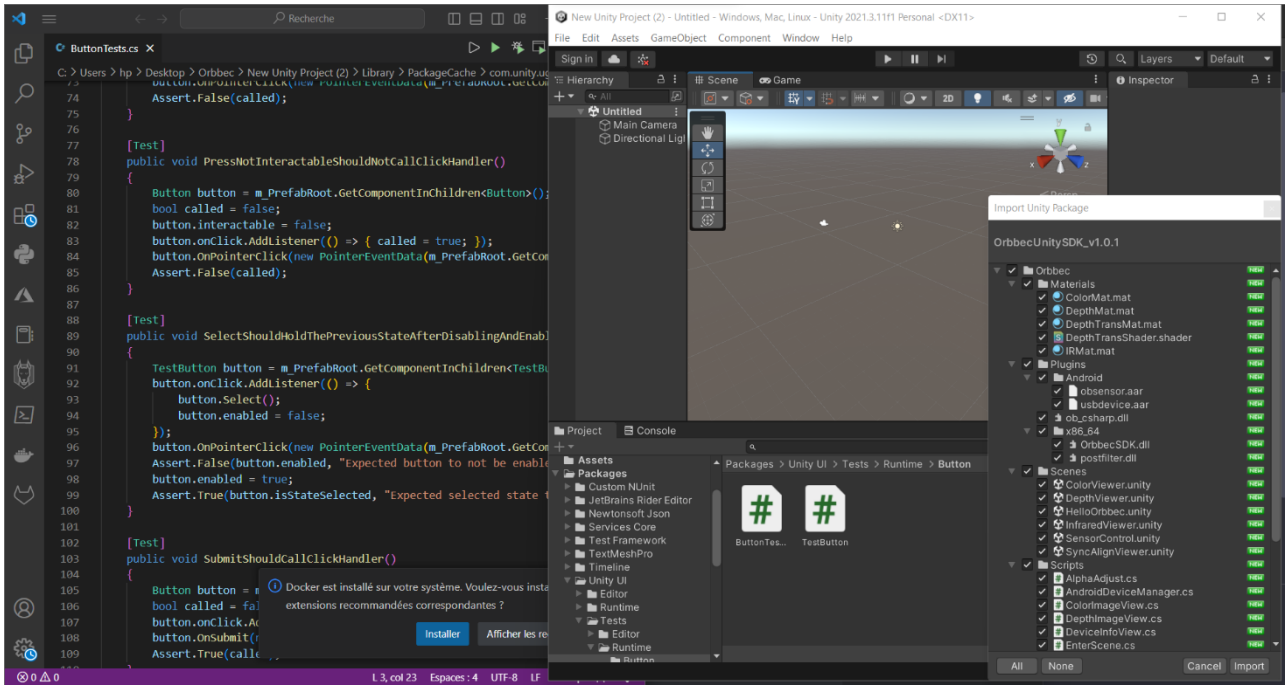
1) Capture des mouvements des joueurs

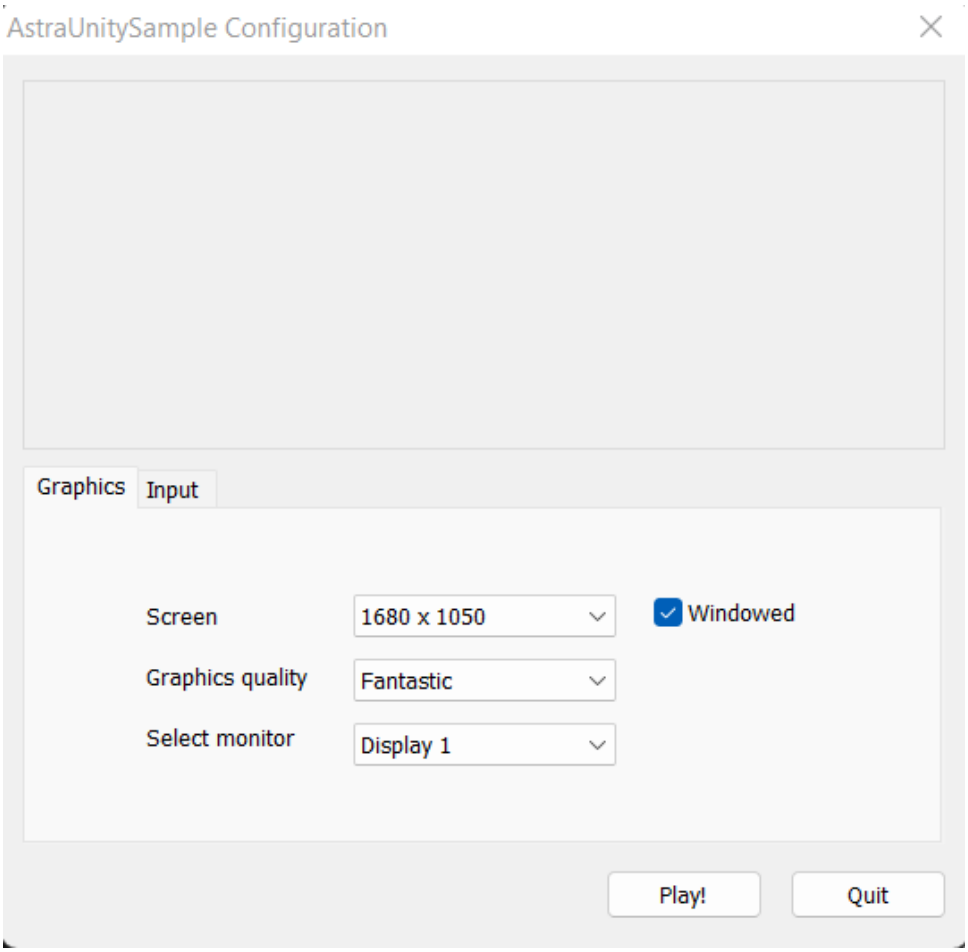
1) La caméra Orbbec Astra capture l'image de la scène de jeu, contenant les joueurs et la grille de jeu.

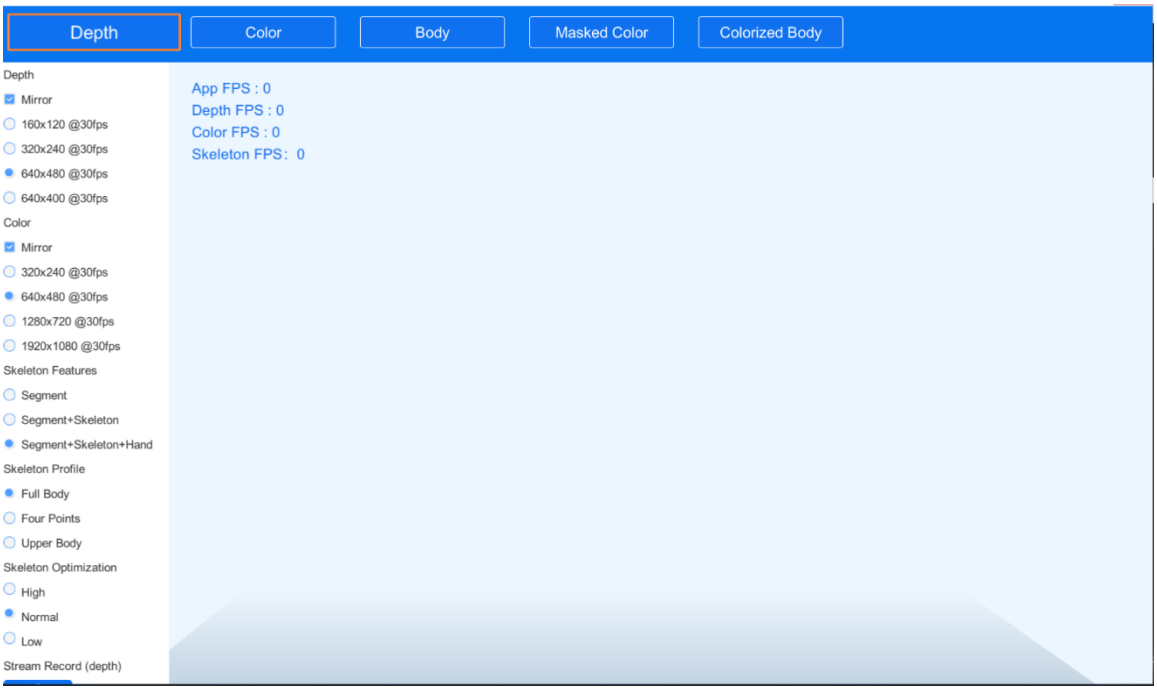
2) Le logiciel de capture de mouvement NiTE utilise les données de l'image pour suivre les mouvements des joueurs.

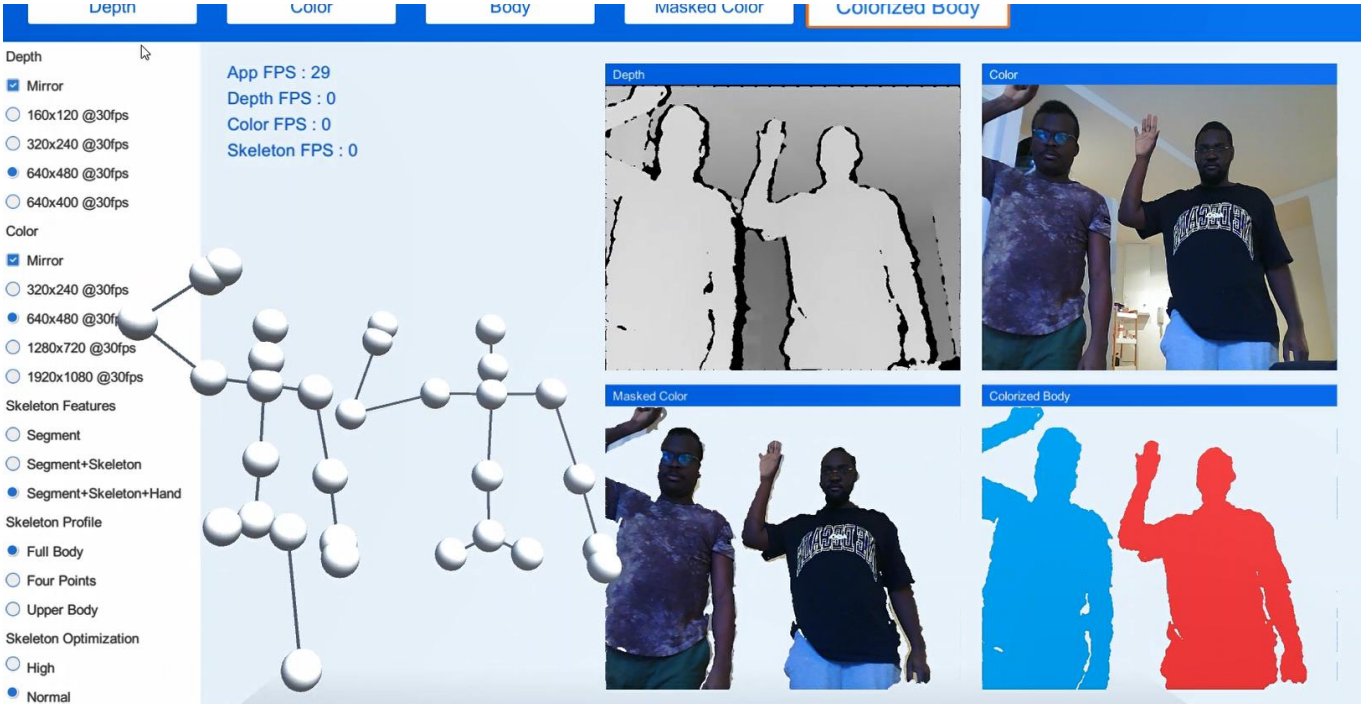
3) Les données des mouvements des joueurs sont ensuite envoyées à Unity via une interface de programmation d'application (API).

Voici quelques images de la caméra Astra Pro 3D en utilisation et d'Unity:




AstraUnitySample Configuration
Graphics
Input
Screen
1680 x 1050
Windowed
Graphics quality
Fantastic
Select monitor
Display 1
Play!
Quit


Depth
Color
Body
Masked Color
Colorized Body
Depth
Mirror
160x120 @30fps
320x240 @30fps
640x480 @30fps
640x400 @30fps
Color
Mirror
320x240 @30fps
640x480 @30fps
1280x720 @30fps
1920x1080 @30fps
Skeleton Features
Segment
Segment+Skeleton
Segment+Skeleton+Hand
Skeleton Profile
Full Body
Four Points
Upper Body
Skeleton Optimization
High
Normal
Low
Stream Record (depth)
App FPS : 0
Depth FPS : 0
Color FPS : 0
Skeleton FPS: 0



2) Conception du jeu Puissance 4

2.1) Développement de l'application

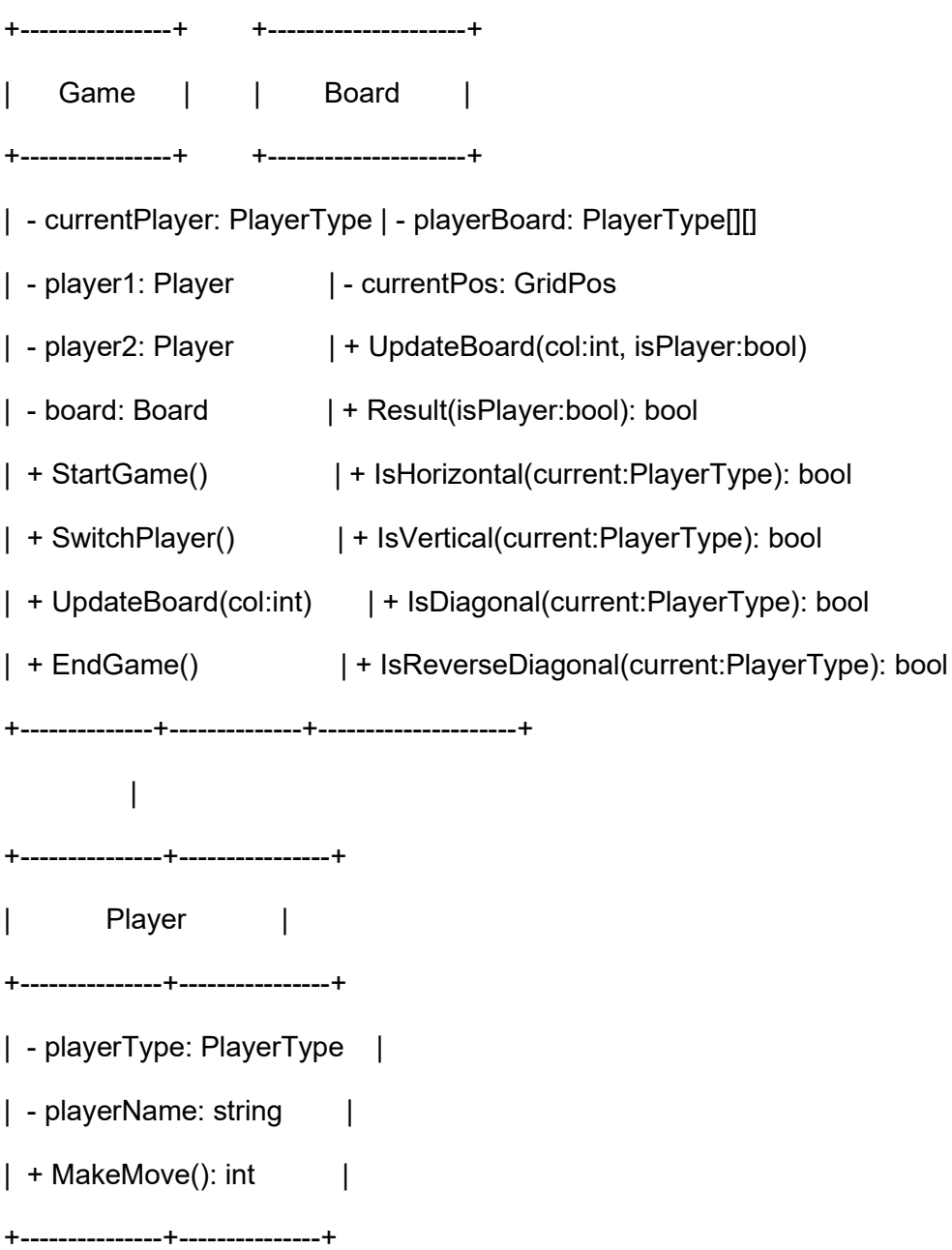
1. Création de la grille de jeu et de l'interface utilisateur.
2. Mise en place de la logique du jeu pour permettre aux joueurs de placer leurs jetons dans les colonnes.
3. Mise en place de la vérification de la victoire et de l'affichage du message de victoire.
4. Ajout d'une option pour recommencer ou quitter le jeu.

5. Mise en place de la vérification du match nul et de l'affichage du message de match nul.

2.2) Tests de l'application

1. Vérification du bon fonctionnement de la grille de jeu et de l'interface utilisateur.
2. Vérification de la logique du jeu pour s'assurer que les joueurs peuvent placer leurs jetons et que les jetons sont placés correctement.
3. Vérification de la vérification de la victoire pour s'assurer que le jeu détecte correctement quand un joueur a aligné quatre jetons.
4. Vérification de l'affichage du message de victoire et de l'option pour recommencer ou quitter le jeu.
5. Vérification de la vérification du match nul et de l'affichage du message de match nul.

2.3.1) Diagramme de classes

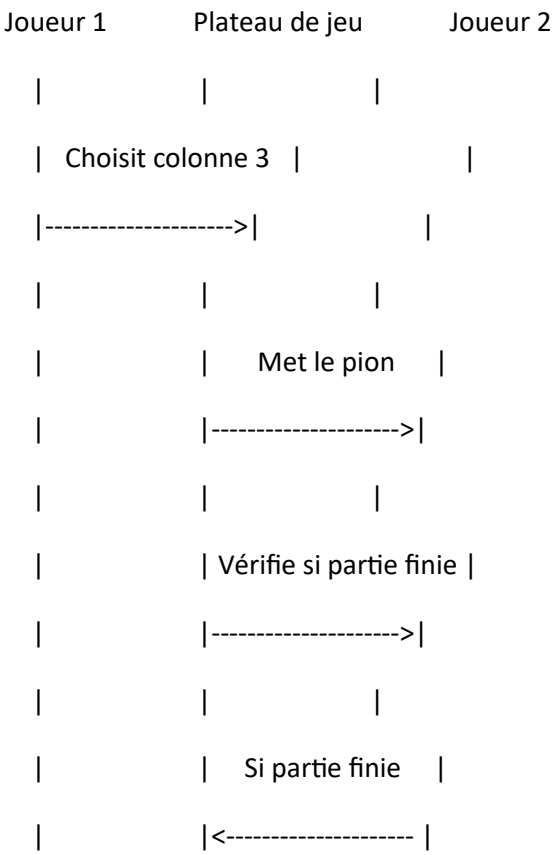


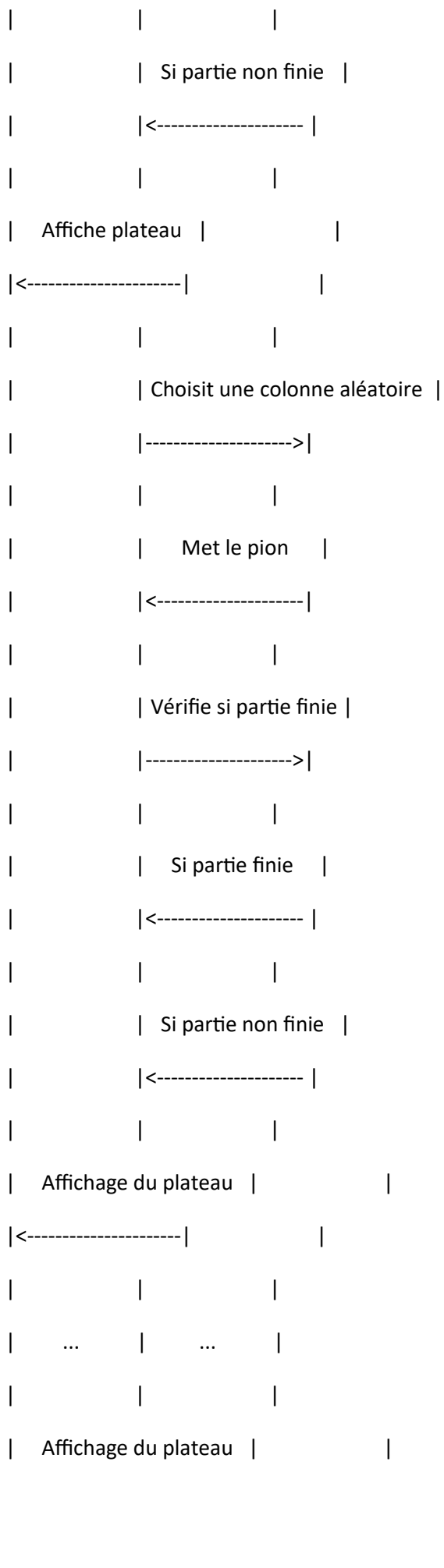
2.3.2) Explication du diagramme de classes

- La classe **Game** représente le jeu de Puissance 4 et contient des informations sur les joueurs (**player1** et **player2**) et le plateau de jeu (**board**).

- La classe **Player** représente un joueur et contient des informations sur son type (**player Type**) et son nom (**playerName**).
- La classe **Board** représente le plateau de jeu et contient une matrice de **Player Type** pour représenter les jetons des joueurs et une position actuelle (**current Pos**).
- La méthode **StartGame()** de **Game** initialise les joueurs et le plateau, et la méthode **EndGame()** termine le jeu.
- La méthode **Switch Player()** de **Game** permet de changer de joueur après chaque coup.
- La méthode **MakeMove()** de **Player** est appelée pour chaque coup et renvoie la colonne choisie par le joueur.
- La méthode **Update Board()** de **Board** est appelée pour mettre à jour le plateau avec le coup joué.
- Les méthodes IS**Horizontal()**, **IsVertical()**, **IsDiagonal()** et **IsReverseDiagonal()** de **Board** sont appelées pour vérifier s'il y a un gagnant.
- La méthode **Result()** de **Board** renvoie **true** s'il y a un gagnant, sinon **false**.

2.4.1) Diagramme de séquence




Si partie non finie
Affiche plateau
Choisit une colonne aléatoire
Met le pion
Vérifie si partie finie
Si partie finie
Si partie non finie
Affichage du plateau
...
...
Affichage du plateau

|<----------------------| |

| | |

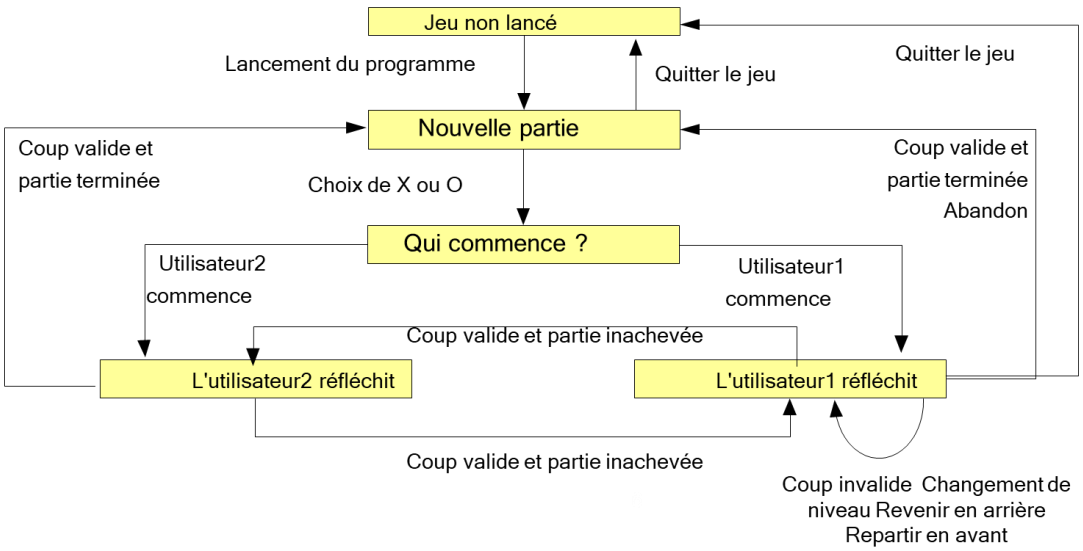
| Affichage du résultat | |

|<----------------------| |

2.4.2) Explication du diagramme de séquence

Ce diagramme représente l'interaction entre deux joueurs et le plateau de jeu pour une partie de Puissance 4. Les joueurs choisissent chacun leur tour une colonne où placer leur pion, le plateau est mis à jour et vérifié pour détecter la fin de la partie. Si la partie n'est pas finie, le plateau est affiché et le tour passe au joueur suivant. Le processus est répété jusqu'à ce que la partie soit terminée, et le résultat final est affiché.

2.5) Cas d'utilisation



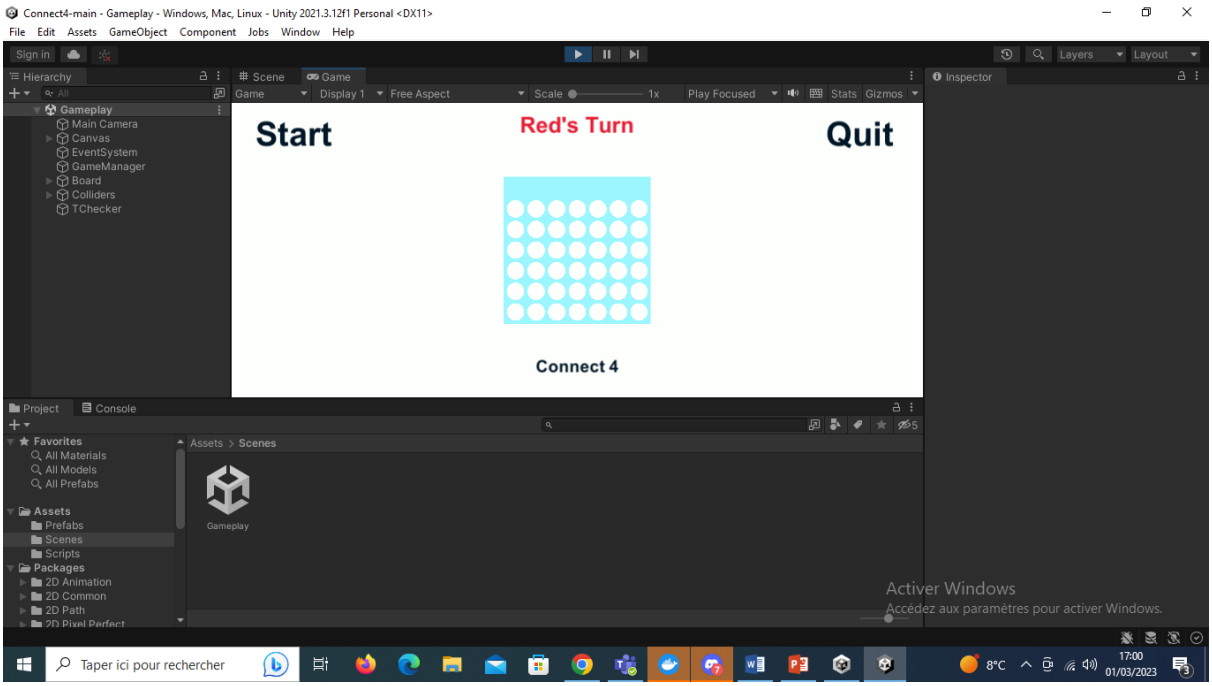
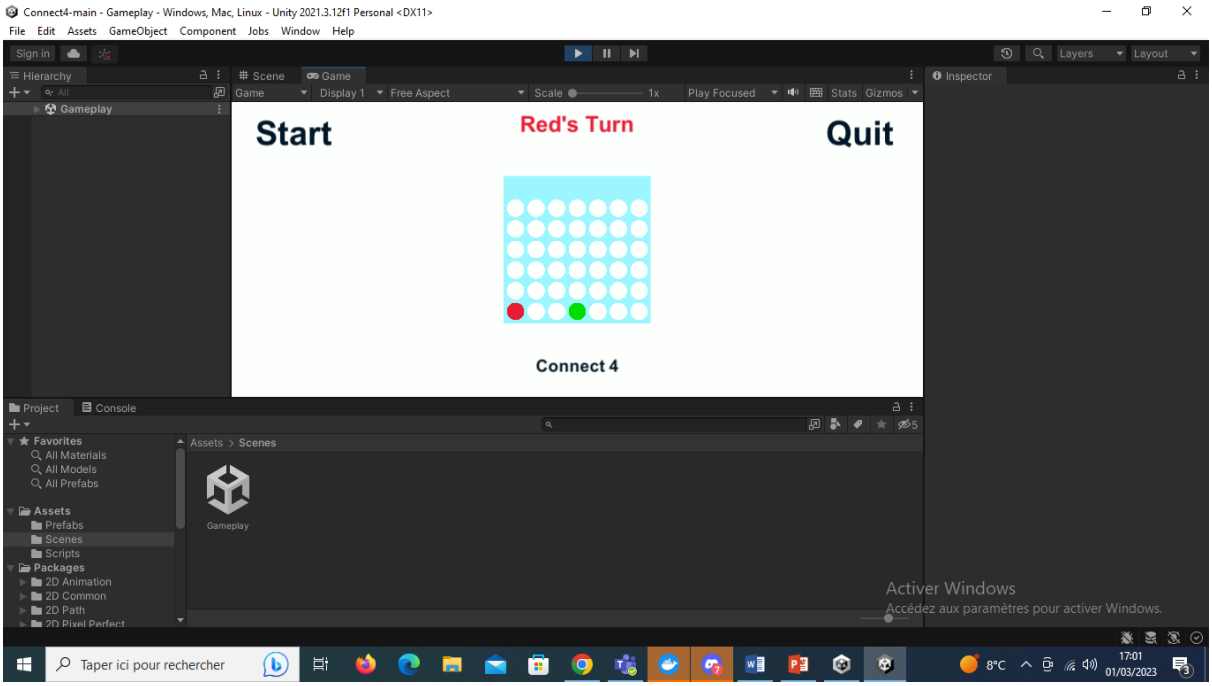


Figure 1 – Début de la partie



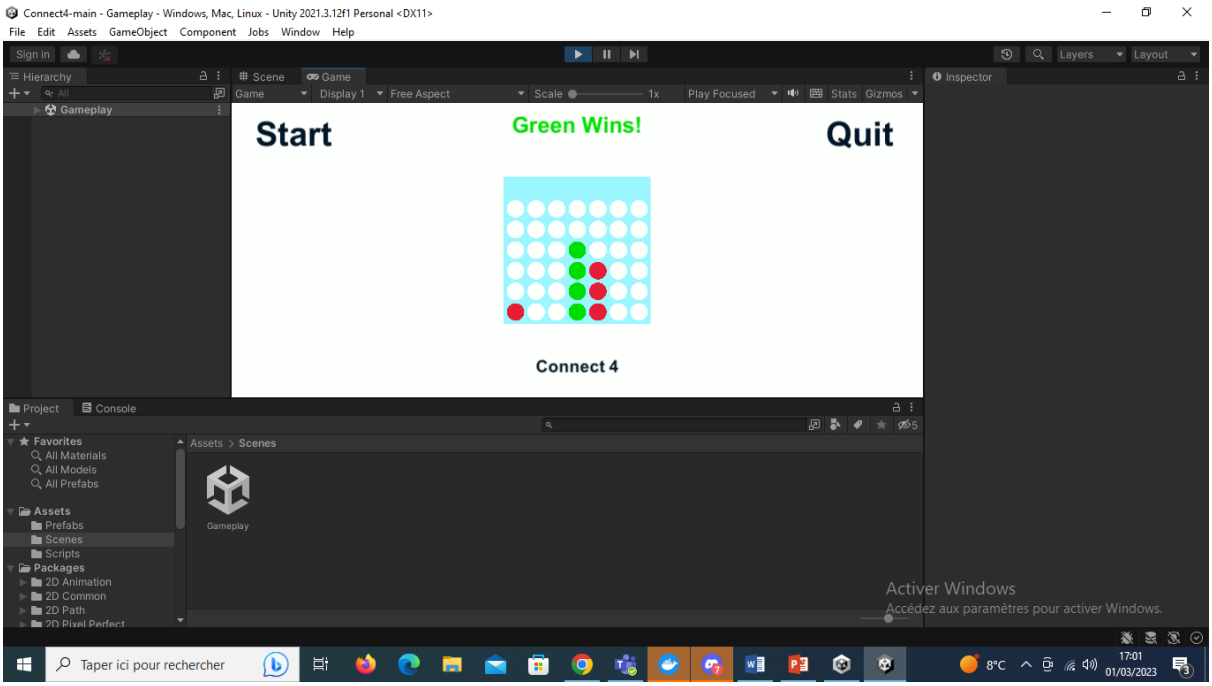


Figure 2 - Victoire du joueur Vert

VI – Mise en place du projet

1) Environnement nécessaire pour ce projet

Caméra : La caméra Orbbec Astra est nécessaire pour capturer les mouvements des joueurs.

Ordinateur : L'ordinateur doit avoir une configuration matérielle suffisante pour exécuter Unity, la bibliothèque OpenCV et le pilote de la caméra. Les spécifications minimales recommandées sont :

- Processeur : Intel Core i5 ou supérieur
- Mémoire vive : 8 Go ou supérieur
- Carte graphique : NVIDIA GeForce GTX 1050 ou supérieur
- Espace de stockage : 1 Go ou supérieur

Logiciels : Les logiciels suivants doivent être installés sur l'ordinateur pour exécuter le projet :

- Unity (version 2019.3.0f6 ou supérieure) : pour le développement et l'exécution du jeu
- Visual Studio (version 2017 ou supérieure) : pour la programmation en C#
- Le pilote de la caméra Orbbec Astra : pour la reconnaissance des mouvements des joueurs
- La bibliothèque OpenCV : pour le traitement des images
- Plateforme de développement : Le projet a été développé sous Windows 10, mais il peut également être développé sur d'autres plates-formes compatibles avec Unity et les autres logiciels requis.

2) Instructions pour installer et configurer le projet Motion Controller Puissance 4

- Installer Unity :

Téléchargez et installez la version 2019.3.0f6 ou supérieure de Unity depuis le site officiel de Unity.
Une fois l'installation terminée, ouvrez Unity et créez un nouveau projet en sélectionnant "New" dans l'écran d'accueil.

- Installer Visual Studio :

Téléchargez et installez Visual Studio 2017 ou une version ultérieure depuis le site officiel de Microsoft.
Lors de l'installation, assurez-vous de sélectionner l'option de développement pour C#.

- Installer le pilote de la caméra Orbbec Astra :

Téléchargez et installez le pilote de la caméra Orbbec Astra depuis le site officiel d'Orbbec.
Branchez la caméra sur votre ordinateur à l'aide du câble USB fourni.

- Installer la bibliothèque OpenCV :

Télécharger et importer les packages nécessaires :
Téléchargez les packages Unity « Kinect v2 Examples with MS-SDK » et « Windows Kinect v2 Examples » depuis l'Asset Store.
Importez les packages dans votre projet Unity en sélectionnant "Import Package" dans le menu « Assets ».

- Importer le projet Puissance 4 :

Téléchargez le code source du projet depuis le dépôt Git.
Le lien git du projet est le suivant : https://gitlab.esiea.fr/nlokupathirage/puissance-4.git

Importez le projet dans Unity en sélectionnant « Add » dans l'écran d'accueil de Unity, puis en naviguant jusqu'au dossier contenant le code source.

- Configurer la caméra :

Dans Unity, ouvrez la scène « OrbbecAstraExample ».
Sélectionnez l'objet "AstraController" dans la hiérarchie et faites défiler jusqu'à la section « Sensor Configuration ».
Sélectionnez le profil « Body Tracking » dans la liste déroulante "Profile".
Activez l'option "Body Tracking Enabled".

- Compiler et exécuter le projet :

Dans Unity, ouvrez la scène "Puissance4".
Sélectionnez "File" -> "Build Settings".
Sélectionnez votre plateforme de destination et cliquez sur "Build and Run" pour compiler et exécuter le projet.
Ces instructions doivent permettre de configurer le projet correctement. Cependant, si des problèmes surviennent, il est recommandé de consulter la documentation fournie avec chaque composant et logiciel utilisé.

3) Lancement du jeu de Puissance 4 et début de capture des mouvement des joueurs

Pour jouer au jeu de Puissance 4 avec des mouvements capturés à l'aide de la caméra Orbbec Astra, voici les étapes à suivre :

1. Installer et lancer le logiciel Unity sur l'ordinateur.
2. Ouvrir le projet de jeu Puissance 4 dans Unity.
3. Connecter la caméra Orbbec Astra à l'ordinateur et installer les pilotes nécessaires pour la faire fonctionner.
4. Configurer la caméra Orbbec Astra dans Unity en utilisant les scripts fournis pour capturer les mouvements des joueurs.
5. Démarrer le jeu en cliquant sur le bouton "Play" dans Unity.
6. Les joueurs peuvent maintenant commencer à jouer en plaçant leurs mains devant la caméra pour déplacer les jetons sur le plateau de jeu.
7. La logique du jeu gère les tours des joueurs et la vérification des mouvements valides.
8. Une fois qu'un joueur a réussi à aligner 4 jetons de sa couleur en ligne droite, diagonale ou verticale, le jeu est terminé et le résultat est affiché à l'écran.

Il est important de noter que le jeu a été développé dans le cadre d'un projet scolaire et n'a pas été complètement finalisé. Les instructions ci-dessus sont fournies à titre indicatif et peuvent nécessiter des ajustements pour fonctionner correctement en fonction de la configuration matérielle et logicielle spécifique de chaque utilisateur.

VII – Alternative à la caméra Orbec Astra Pro 3D

Si vous n’avez pas réussi avec la Caméra Astra Pro 3D, notez qu’il est tout à fait possible d’utiliser la Kinect V1 pour avoir un même résultat.

1) Documentation générale

Voici les caractéristiques de la Kinect V1 :

- Caméra RGB : capture des images en couleurs à 30 images par seconde avec une résolution de 640 x 480 pixels.
- Capteur de profondeur : utilise la technologie de la lumière structurée pour détecter la distance entre les objets et la caméra. Il capture des images en noir et blanc à 30 images par seconde avec une résolution de 320 x 240 pixels.

Microphone en réseau : permet aux joueurs de donner des commandes vocales à la console de jeux vidéo.

❖ Algorithmes de traitement :

- Suivi de corps : permet de détecter et suivre les mouvements de jusqu'à six joueurs en temps réel. Il utilise un modèle de squelette virtuel pour décrire la pose du joueur.
- Reconnaissance de gestes : permet de reconnaître des gestes spécifiques effectués par les joueurs. Il utilise des algorithmes d'apprentissage automatique pour entraîner un modèle de reconnaissance de gestes.
- Reconnaissance vocale : permet de reconnaître les commandes vocales données par les joueurs. Il utilise des modèles de reconnaissance vocale pour identifier les mots et phrases prononcés par les joueurs.

Spécifications techniques :

- Interface : USB 2.0
- Profondeur de champ : de 1,2 m à 3,5 m
- Angle de vue : 57 degrés horizontalement, 43 degrés verticalement
- Alimentation : adaptateur secteur 12V DC 1,08A

Pour utiliser la Kinect V1 sur un ordinateur, il est recommandé d'avoir une configuration matérielle suffisamment puissante pour prendre en charge les exigences de la caméra et du capteur de profondeur.

Voici les spécifications minimales recommandées pour utiliser la Kinect V1 :

- Processeur : Intel Core 2 Duo 2.66 GHz ou équivalent
- Mémoire vive (RAM) : 2 Go
- Carte graphique : ATI Radeon HD 5670 ou NVIDIA GeForce GT 8800
- Système d'exploitation : Windows 7 ou supérieur
- Port USB : 2.0

Télécharger la version 1.8 du SDK Kinect pour Windows depuis le site web de Microsoft. Le pilote de la Kinect V1 se trouve dans le SDK.

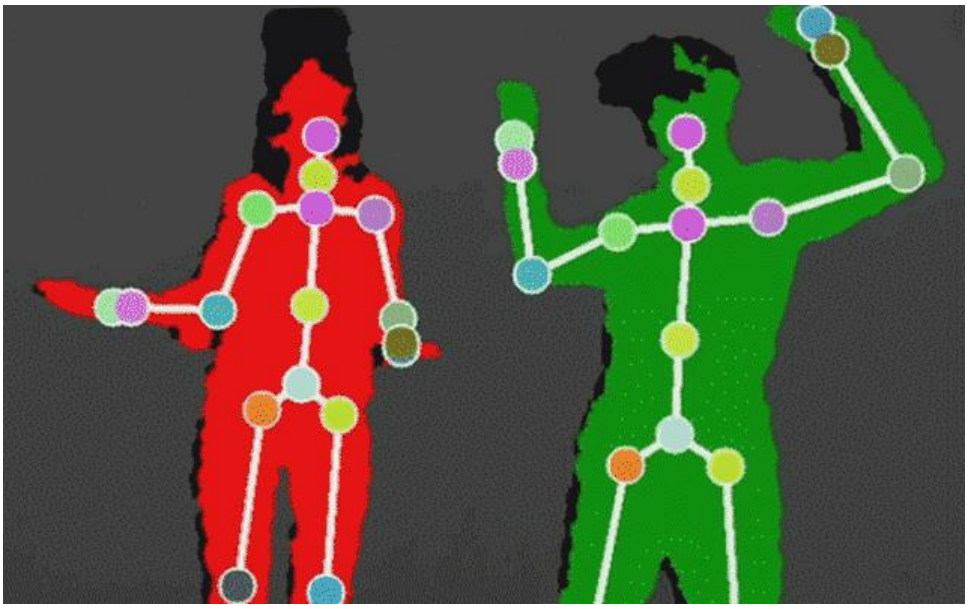
2) Liaison de la Kinect avec Unity

Voici les étapes à suivre pour connecter une Kinect dans Unity pour jouer à un jeu de Puissance 4 :

- Tout d'abord, vous devez avoir une Kinect et un ordinateur compatible avec la Kinect pour le connecter à votre ordinateur. Si vous n'avez pas encore connecté votre Kinect, suivez les instructions fournies avec la Kinect pour l'installer correctement.
- Ouvrez Unity et créez un nouveau projet. Dans le menu "Assets", sélectionnez "Import Package" puis "Kinect" pour installer le package Kinect.
- Sélectionnez "Window" dans la barre de menu, puis "Kinect". Cela ouvrira l'interface utilisateur Kinect dans Unity.
- Dans l'interface utilisateur Kinect, activez les fonctions que vous souhaitez utiliser, par exemple, la reconnaissance de mouvements.
- Créez un objet vide dans votre scène de jeu, nommez-le "Kinect Controller" et ajoutez-lui un script. Ce script servira de contrôleur pour la Kinect.
- Dans le script Kinect Controller, ajoutez le code pour configurer les paramètres de la Kinect et récupérer les données de mouvement. Vous pouvez utiliser la documentation Kinect pour vous aider à comprendre comment accéder aux données de la Kinect.
- Créez une grille pour le jeu Puissance 4. Vous pouvez utiliser des objets vides pour créer des emplacements de jetons, ou créer un tableau de positions pour les jetons dans le script du contrôleur Kinect.
- Ajoutez le code pour détecter les mouvements de la main du joueur et placer les jetons sur la grille. Vous pouvez utiliser des collisions pour détecter quand la main du joueur est sur un emplacement de jeton valide.
- Testez votre jeu en connectant votre Kinect à votre ordinateur et en jouant à Puissance 4 en utilisant vos mouvements de la main.

Ces étapes devraient vous permettre de connecter une Kinect à Unity et de créer un jeu de Puissance 4 jouable avec des mouvements de la main. N'hésitez pas à consulter la documentation Unity et Kinect pour en savoir plus sur les fonctions disponibles et les meilleures pratiques pour utiliser la Kinect dans Unity.

Voici une image de ce que vous pouvez obtenir à la fin :



Il est important de noter que le jeu a été développé dans le cadre d'un projet scolaire et n'a pas été complètement finalisé. Les instructions ci-dessus sont fournies à titre indicatif et peuvent nécessiter des ajustements pour fonctionner correctement en fonction de la configuration matérielle et logicielle spécifique de chaque utilisateur.