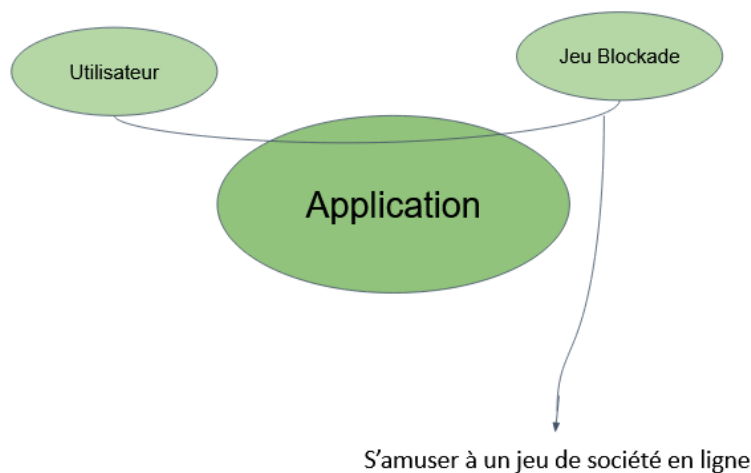


Cahier des charges de la partie IHM

Diagramme des interactions (Doha)



1) AUTEURS CAHIER DES CHARGES PARTIE IHM

Nom et Prénom	Partie traitée dans l'implémentation du jeu	Rôle
ABERKANE Doha		- Référente partie IHM
LAROCHE Nolan		- Partie IHM
GADET Martin	Menu de démarrage et de fin de jeu	- Partie IHM
MONTIGNY Thomas	Un mode développeur pour regarder et paramétrer une partie entre 2 IA. Un plateau vu du dessus et plusieurs paramètres à gauche de l'écran. Faire tout le code qui est prêt à prendre une matrice en argument pour ajouter un mur, et un id de pion et un mouvement pour déplacer un pion.	- Partie IHM
BOUKHARI Wassim	- Animation de déplacement des pions - Animation lorsqu'un mur est placé sur le plateau	- Partie IHM

2) Besoins et contraintes liés au projet

1) BESOINS

WASSIM ET DOHA

L'utilisateur a besoin d'une interface graphique permettant d'interagir avec le programme Blockade. Il faudra aussi réagir au code des autres code en développement afin de réagir graphiquement aux messages qu'il envoie.

Outils à utiliser:

- Unity (tous les packages avec les Assets, éléments, etc.)
- ProBuilder (Plugin Unity)
- Blender
- Développement en C#

Fonctionnalités Principales:

NOLAN

1. Accueil

Un menu qui accueille l'utilisateur et qui lui demande d'appuyer sur n'importe quelle touche pour continuer. Ce qui une fois fait le dirigera vers le Menu de base.

2. Menu de base

Le menu de base sera composé de trois boutons. Tout d'abord un bouton Jouer qui une fois que l'utilisateur a appuyer dessus permet de lancer une partie ou à terme d'accéder au menu de près partie et de choisir de jouer "en ligne" ou "hors ligne". Ensuite un menu Options qui permet d'afficher les différentes options du jeu tels que : le choix de la langue, la gestion du son et pour finir la gestion des animations (le choix de les activer ou non) . Pour finir un bouton Comment jouer? qui permet à l'utilisateur de revoir les règles du jeu et qui explique comment jouer a Blockade.

3. Fin de jeu avec une animation

Animation sur le terrain montrant la victoire/défaite du joueur en jeu avec l'affichage d'un écran de victoire/défaite pour le joueur.

4. Création de 4 pions (2 pour chaque joueur)

Création de 4 pions pour permettre le jeu possible.

5. Plateau 11x14 sur Unity avec 4 pions aux positions (4,4), (8,4), (4,11), et (8,11)

Création d'un plateau avec 4 pions aux bonnes positions déterminées par les règles du jeu.

6. Visualisation de l'état du plateau

Il y aura 2 caméras possibles , une caméra vue du dessus et une en tant que joueur de son côté du plateau. De son point de vue, le joueur pourra voir sur les côtés du plateau les murs qui lui restent.

7. Visualisation de la position du pion et du mur avant la pose

Création d'un point de vue du plateau pour permettre à l'utilisateur de voir l'intégralité du plateau de jeu et de préparer son coup à jouer pour son prochain coup.

8. Animation de déplacement des pions

Création d'une animation pour rendre plus "vivant" le déplacement des pions lorsqu'un joueur réalise un coup.

9. Animation lorsqu'un mur est placé sur le plateau

Création d'une animation lorsqu'un mur est placé sur le plateau pour rendre l'action plus vivante pour la partie.

Fonctionnalités Secondaires:

WASSIM DOHA

- Personnalisation du pion et du personnage
 - L'utilisateur doit pouvoir avoir accès à un menu permettant de personnaliser son personnage et son pion. Cela permettrait de modifier certaines animations au sein de la partie
- Score du joueur
- Menu avec un arrière plan animé
- Animation en arrière plan pendant la partie

II) CONTRAINTES

DOHA WASSIM

- Utilisation d'assets gratuits
- Interface "user-friendly" + fournir des commentaires clairs et précis aux joueurs pour les aider à comprendre et à utiliser l'interface utilisateur
- Compatibilité avec le code fait par les autres groupes

3) Résultats attendus

MARTIN ET THOMAS

- Plateau 3D avec des pions et murs personnalisés selon un thème médiéval.
- Un mode développeur pour regarder et paramétrer une partie entre 2 IA. Un plateau vu du dessus et plusieurs paramètres à gauche de l'écran. TODO
- Faire tout le code qui est prêt à prendre une matrice en argument pour ajouter un mur, et un id de pion et un mouvement pour déplacer un pion.
- Utilisation d'assets provenant de l'Asset Store Unity:
 - Blacksmith: [lien](#)
 - Medieval Stone Keep: [lien](#)
 - LowPoly Style Free Rocks and Plants: [lien](#)
 - Low Poly Fantasy Valley with Village: [lien](#)
 - Mountain Valley Stylized Fantasy RPG Environment: [lien](#)
 - Medieval Music Pack Vol. 2: [lien](#)
- Utilisation d'assets provenant de Kenney:
 - Graveyard Kit: [lien](#)
 - Mini Dungeon: [lien](#)
 - Fantasy Town Kit: [lien](#)
 - Survival Kit: [lien](#)
 - Medieval Town (Base): [lien](#)



Exemple de ce à quoi pourrait ressembler les alentours du plateau de jeu