

Projet Informatique 1^{ère} Année
PRO 3600

Jeu XKube

MOREAU Romain
LANTIGNY Valentin
ROUX Basile
FOISSY Thomas

Enseignant responsable : **SIMATIC Michel**

24 Février 2023

Sommaire

1	Introduction	3
2	Cahier des charges.....	4
3	Développement	7
3.1	Analyse du problème liste des fonctionnalités	7
4	Conclusion	11
5	Annexes.....	12
5.1	Annexe 1 : Gestion de projet.....	12
5.2	Annexe 2 : code source	14

1 Introduction

L'objectif de notre projet est de développer une application pour mobile. Cette application se basera sur un jeu de logique inspiré du Morpion et de la Pipopipette. Ce dernier se joue normalement sur papier avec 2 joueurs qui s'affrontent. On représente initialement une grille de 4x4 cases dont on ne trace que les coins. On doit alors avoir 25 points disposés sur 5 lignes et 5 colonnes. Chacun leur tour, les joueurs possèdent 2 coups qu'ils devront utiliser en reliant 2 points l'un avec l'autre pour former une barre horizontale ou verticale. Ce sont les joueurs qui tracent la grille petit à petit.

Lorsqu'une des cases voit ces 4 côtés être tracés, l'un des joueurs peut dépenser l'un de ces coups pour tracer une croix de sa couleur sur cette case. L'objectif sera d'être le premier à aligner 3 croix de sa couleur en ligne, en colonne ou en diagonal comme au morpion.

D'apparence simpliste, il s'avère enfaîte très compliqué de trouver une stratégie gagnante dans 100% des cas, car tenter d'anticiper un coup fait augmenter le nombre de possibilité de manière exponentielle comme au échec. Par ailleurs, une partie ne ressemble jamais à une autre et le jeu semble symétrique pour les 2 joueurs : le premier qui joue n'a pas plus de chance de gagner que le deuxième.

En développant ce jeu sur mobile, l'objectif serait de pouvoir faire une partie rapidement avec d'autres joueurs. Cela faciliterait la mise en place du jeu : plus besoin de feuille et de crayon.

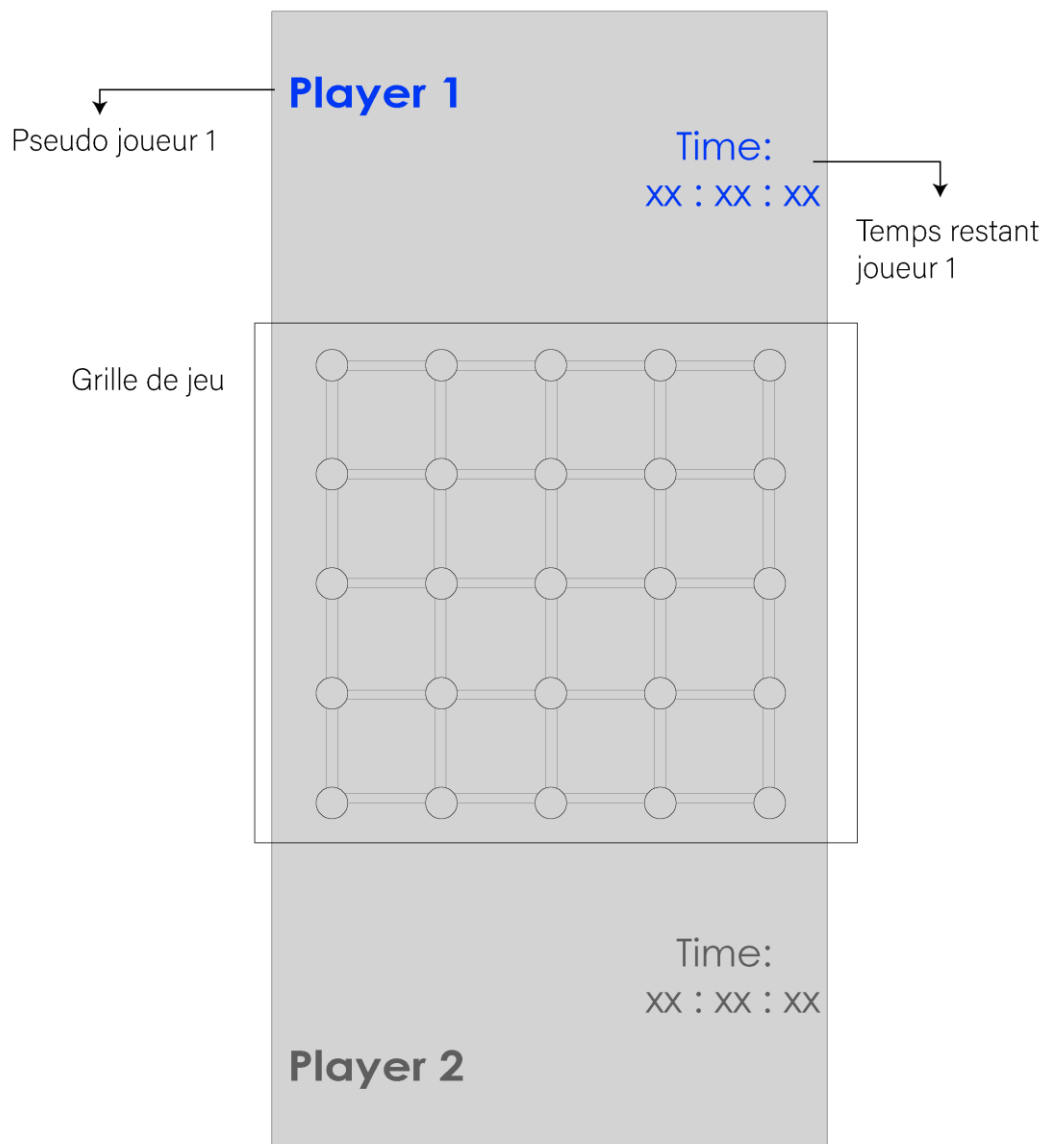
Par ailleurs, cette application qui nécessite peu de fonctionnalités sera une bonne occasion pour toucher à des outils et des domaines que nous n'avons pour l'instant jamais utiliser : Utiliser un IDE spécialisé pour les applications mobiles (Android studio), faire fonctionner un jeu multijoueur sur différents appareils, utiliser les réseaux de neurones.

2 Cahier des charges



Écran Principal de L'Application

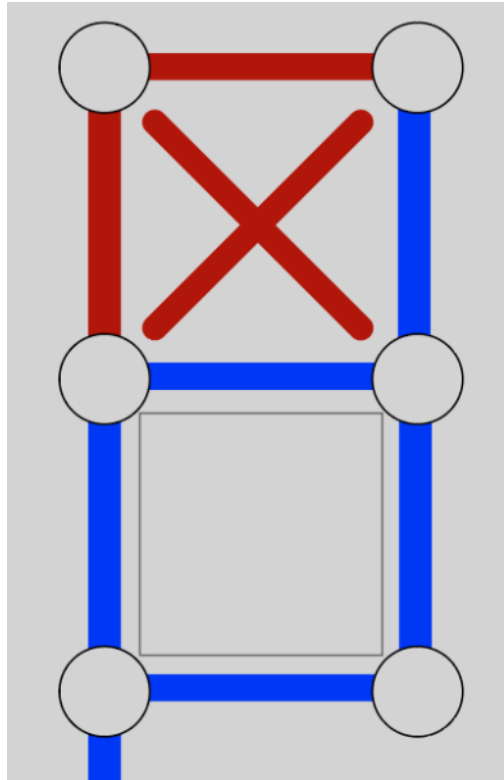
L'application s'ouvrirait sur un écran permettant de choisir son mode de jeu. On pourra jouer à 2 joueurs sur le même appareil, ou chacun sur son propre téléphone en connexion locale. Il y aura également possibilité de jouer en solo contre une IA, On retrouve également le Logo de l'application ainsi que plusieurs options : gérer la luminosité, le son, obtenir un tutoriel, changer la langue, ajouter un shop avec des accessoires pour les joueurs, changer l'apparence du jeu, diminuer la qualité des animations pour les petits portables.



Écran Lors du Lancement du Jeu

Une fois une partie lancée, l'écran de jeu ressemblera à ceci avec les emplacements pour cliquer pré-dessinés, les pseudos des 2 joueurs et un chrono pour éviter que la partie s'éternise. Le joueur qui doit jouer est celui dont le pseudo est en couleur.

Le fond n'est pas encore défini mais devrait également indiquer par sa couleur, quel joueur doit jouer.



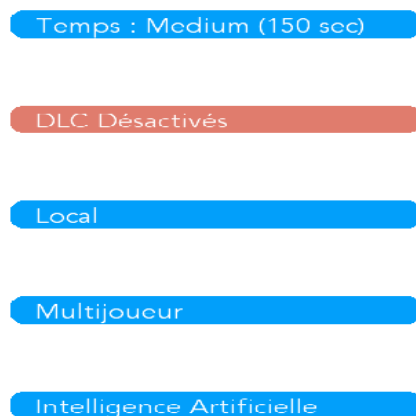
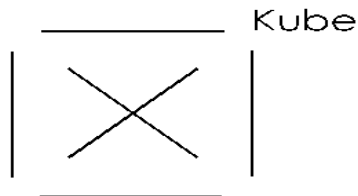
Aspect après quelques coups des joueurs

Chaque joueur aura une couleur arbitraire qui lui sera attribué : par défaut bleu ou rouge. Chacun de ses coups sera représenté avec cette couleur. En particulier, le dernier coup joué par un joueur sera d'une couleur légèrement plus clair pour qu'on puisse clairement identifier quels coups ont été joué au dernier tour.

L'objectif est que le jeu se déroule selon les règles du jeu de base : 2 coups par joueur, interdiction de joué une croix si les 4 côtés ne sont pas remplis, victoire du joueur qui aligne 3 croix peu importe la direction.

3 Développement

3.1 Analyse du problème liste des fonctionnalités



Écran Principal de L'Application

Nous avons pu implémenter les 3 modes de jeux (Local, Multijoueur et IA) que l'on peut lancer en appuyant sur le bouton associé.

- Le mode local permet à 2 personnes de jouer sur le même appareil en appuyant chacun leur tour sur l'écran.

- Le mode Multijoueur redirige vers un écran d'attente et lance une partie lorsqu'un 2ème joueur demande également à jouer en multijoueur.

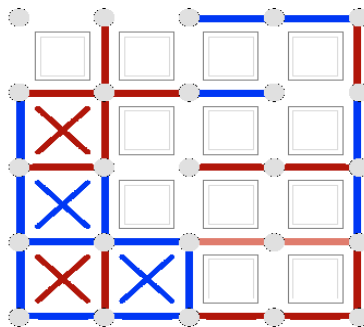
- Le mode IA se joue en solo : le joueur commence et l'IA joue derrière lui en essayant toujours de gagner.

Dans tous les cas, la partie se termine quand un joueur remplit une condition de victoire ou que

le chrono de son adversaire arrive à 0. Un écran apparaît, félicitant le joueur concerné. On peut alors revenir à l'écran d'accueil en cliquant une nouvelle fois sur l'écran.

Nous n'avons toute fois pas créé les options initialement annoncées faute de temps et de priorité. A la place 2 autre les remplacent : La possibilité de modifier le temps du chronomètre et l'activation d'une option : le DLC qui ajoute une modification au mode de jeu de base.

1 50 6

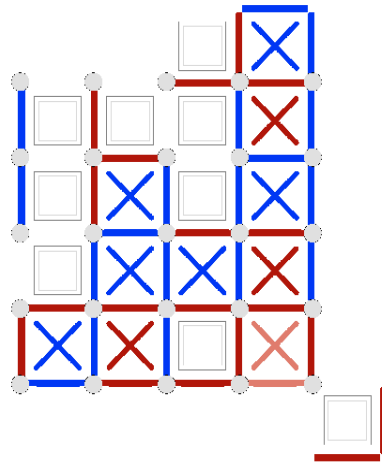


2 29 7

Écran durant la partie

Les règles du jeu sur papier ont bien été retranscrit dans l'application. Les conditions de victoires sont toutes implémentées (alignement et temps épuisé).

1 20 2



2 29 6

La mécanique de DLC fonctionne très bien. Il s'agit d'une modification aléatoire au début de la partie qui ajoute un ou plusieurs carrés sur le côté du terrain. Ces carrés peuvent également servir à faire des alignements pour gagner.

Multijoueur

Le multijoueur fonctionne grâce à un serveur node.js en javascript. Lorsqu'un joueur clique sur le bouton multijoueur, que ce soit sur la version .apk ou .jar, il est placé dans une salle d'attente. Lorsqu'un 2ème joueur se connecte, une partie se lance automatiquement entre les 2 joueurs.

L'intelligence Artificiel

L'intelligence artificiel fonctionne dans les grandes lignes de la manière suivante : elle joue toujours en 2ème face à un joueur humain. Elle utilise un système de comportement. Même en réalisant de nombreuses parties sur le jeu en tant qu'humain nous n'avons pas (encore) trouvé de stratégie qui soit 100 % gagnante. Cependant, nous avons tout de même identifié des comportements à prioriser dans différents cas. Ainsi, pour chacun de ces cas, nous avons codé un comportement à l'IA pour qu'elle agisse comme nous pensons qu'il est le mieux d'agir. Ce n'est donc pas de la vraie intelligence, plutôt un enchaînement de if else et de for. Cependant, nous pouvons raisonnablement penser qu'il n'y aurait pas de meilleur comportement à adopter pour tous ceux que nous avons fait sur mesure. Cela réduit par ailleurs considérablement la complexité de l'IA.

Dans les cas de figures où il y a une plus grande part de subjectivité, si aucun comportement n'est adapté à la situation, alors nous utilisons une intelligence artificielle basée sur un modèle minimax. C'est un comportement plus général, bien plus complexe, mais qui permet de

trouver le meilleur coup à jouer quand celui-ci n'est pas évident.

4 Conclusion

Les premières heures consacrées aux projets ont permis d'accomplir plusieurs objectifs.

En premier lieu, une conception préliminaire a permis de définir notre interface de jeu, ainsi que les spécialités nécessaires pour son bon fonctionnement. Dans un second temps, le développement du code source permet d'avoir une application fonctionnelle : le jeu se lance sur Android, et fonctionne correctement (alternance des couleurs en fonction du joueur, etc.). Enfin, l'interface et les graphismes ont été conçus, permettant d'avoir par la suite une application plus esthétiques.

Pour les prochaines étapes, il reste à concevoir et à mettre en place une connexion pour jouer en multijoueur ainsi que la formation d'une intelligence artificielle afin de jouer seul. Une fois ces différentes étapes accomplies, il restera l'implémentation des outils divers assurant une meilleure expérience pour le joueur.

5 Annexes

5.1 Annexe 1 : Gestion de projet

5.1.1 Planning prévisionnel

PLAN DE CHARGES PRÉVISIONNEL

Description de l'activité	Charge en %	Charge en H	Charge en H / Participant			
			Romain	Thomas	Basile	Valentin
Total	100%	211	53	51	53	54
Gestion de projets	22%	46	12	11	11	12
Réunion de lancement	2%	4	1	1	1	1
Planning prévisionnel et Suivi d'activités	2%	4	1	1	1	1
Réunions de suivi	11%	24	6	6	6	6
Rédaction des rapports	3%	6	2	1	1	2
Mise en place de Git et gestion de l'espace	4%	8	2	2	2	2
Spécification	4%	8	2	2	2	2
Définition des fonctionnalités	4%	8	2	2	2	2
Conception préliminaire	11%	24	6	6	6	6
Définition du type d'application	2%	4	1	1	1	1
Définition du langage et des outils nécessaires	2%	4	1	1	1	1
Définition des fonctionnalités	6%	12	3	3	3	3
Définition des graphismes	2%	4	1	1	1	1
Conception détaillée	21%	45	12	11	11	11
Définition des classes	2%	4	1	1	1	1
Définition des méthodes	8%	16	4	4	4	4
Essais	6%	12	3	3	3	3
Auto-formation	4%	8	2	2	2	2
Interfaces et graphismes	2%	5	2	1	1	1
Code (jeu fonctionnel)	16%	33	9	9	9	9
Codage des classes	4%	8	2	2	2	2
Codage des méthodes	10%	21	6	6	6	6
Essais	2%	4	1	1	1	1

Intégration des fonctionnalités	17%	36	8	8	10	10
Implémentation de l'IA	9%	20	5	5	5	5
Implémentation du multijoueur	5%	10	1	1	4	4
Implémentations diverses	3%	6	2	2	1	1
Soutenance	8%	16	4	4	4	4
Préparation de la soutenance	6%	12	3	3	3	3
Soutenance	2%	4	1	1	1	1

5.1.2 Suivi d'activités

SUIVI D'ACTIVITÉS (Charge Consommée)

Description de l'activité	Charge en %	Charge en H	Charge en H / Participant			
			Romain	Thoma s	Basile	Valenti n
Total	163%	343	55	55	118	115
Gestion de projets	11%	38	10	9	9	10
Réunion de lancement	1%	4	1	1	1	1
Planning prévisionnel et Suivi d'activités	1%	4	1	1	1	1
Réunions de suivi	3%	12	3	3	3	3
Rédaction des rapports	1%	3	1	1	0	1
Mise en place de Git et gestion de l'espace	4%	15	4	3	4	4
Spécification	2%	8	2	2	2	2
Définition des fonctionnalités	2%	8	2	2	2	2
Conception préliminaire	9%	30	9	6	9	9
Définition du type d'application	1%	4	1	1	1	1
Définition du langage et des outils nécessaires	2%	6	1	1	2	2
Définition des fonctionnalités	5%	16	3	3	5	5
Définition des graphismes	1%	4	4	1	1	1
Conception détaillée	19%	64	11	11	21	21
Définition des classes	3%	10	1	1	4	4
Définiton des méthodes	3%	12	1	1	5	5
Essais	3%	10	1	1	4	4
Auto-formation	6%	20	5	5	5	5
Interfaces et graphismes	3%	12	3	3	3	3
Code (jeu fonctionnel)	35%	120	12	12	48	48

Codage des classes	13%	44	2	2	20	20
Codage des méthodes	15%	50	5	5	20	20
Essais	8%	26	5	5	8	8

Intégration des fonctionnalités	14%	49	6	10	24	20
Implémentation de l'IA	5%	17	1	5	10	1
Implémentation du multijoueur	8%	27	1	1	10	15
Implémentations diverses	1%	5	4	4	4	4

Soutenance	6%	20	5	5	5	5
Préparation de la soutenance	5%	16	4	4	4	4
Soutenance	1%	4	1	1	1	1

5.2 Annexe 2 : code source

Lien vers le gitlab avec le code source :

<https://gitlab.com/Racoon-r/Projet-dev>