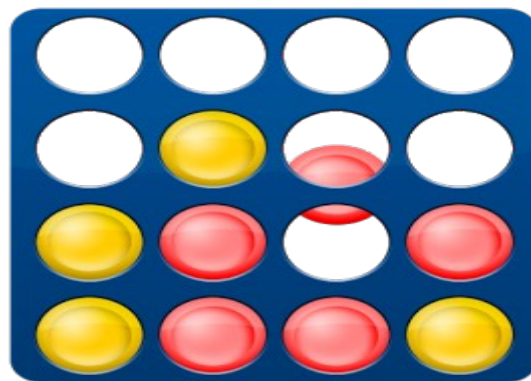


Rapport de projet

Puissance 4



Binôme : AMOURA Gaya
ABDELLI Idir

2020 /2021

Sommaire :

1) Contexte du projet

2) Conception

Premier diagramme

Diagramme finale

Diagramme d'utilisation

Quelques algorithmes pertinents.

3) Méthodologie

4) Répartition des tâches

5) Difficultés rencontrées

6) Conclusion

1) Contexte :

Le projet consiste en la réalisation du jeu de stratégie et de réflexion « Puissance 4 », pour pouvoir y jouer à un ou deux joueurs avec différents niveaux de difficulté.

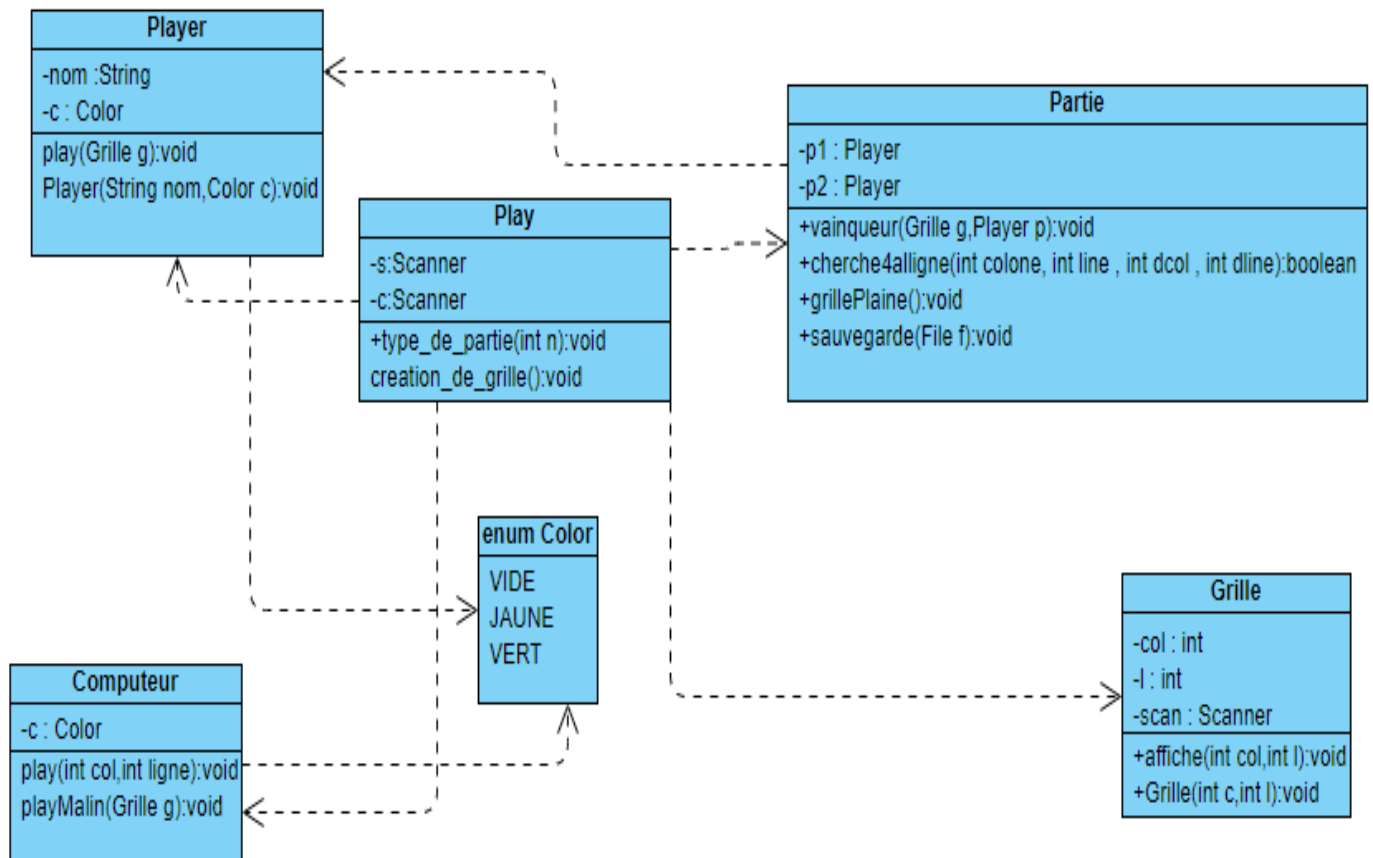
En premier lieu nous avons établi un UML de base nous permettant de schématiser le déroulement du programme que nous avons ensuite implémenter en langage JAVA. Le code est accompagné de sa documentation qui explique le rôle et le fonctionnement de chaque fonction.

L'ensemble du code est disponible dans la présente archive où se situe ce rapport.

2) Conception :

La première conception du diagramme a été faite pour schématiser les différentes classe et étapes du programme, nous nous sommes inspirés des explications fournies en cours et sur l'énoncé du projet ainsi nous avons imaginé la figure 1 :

Premier diagramme

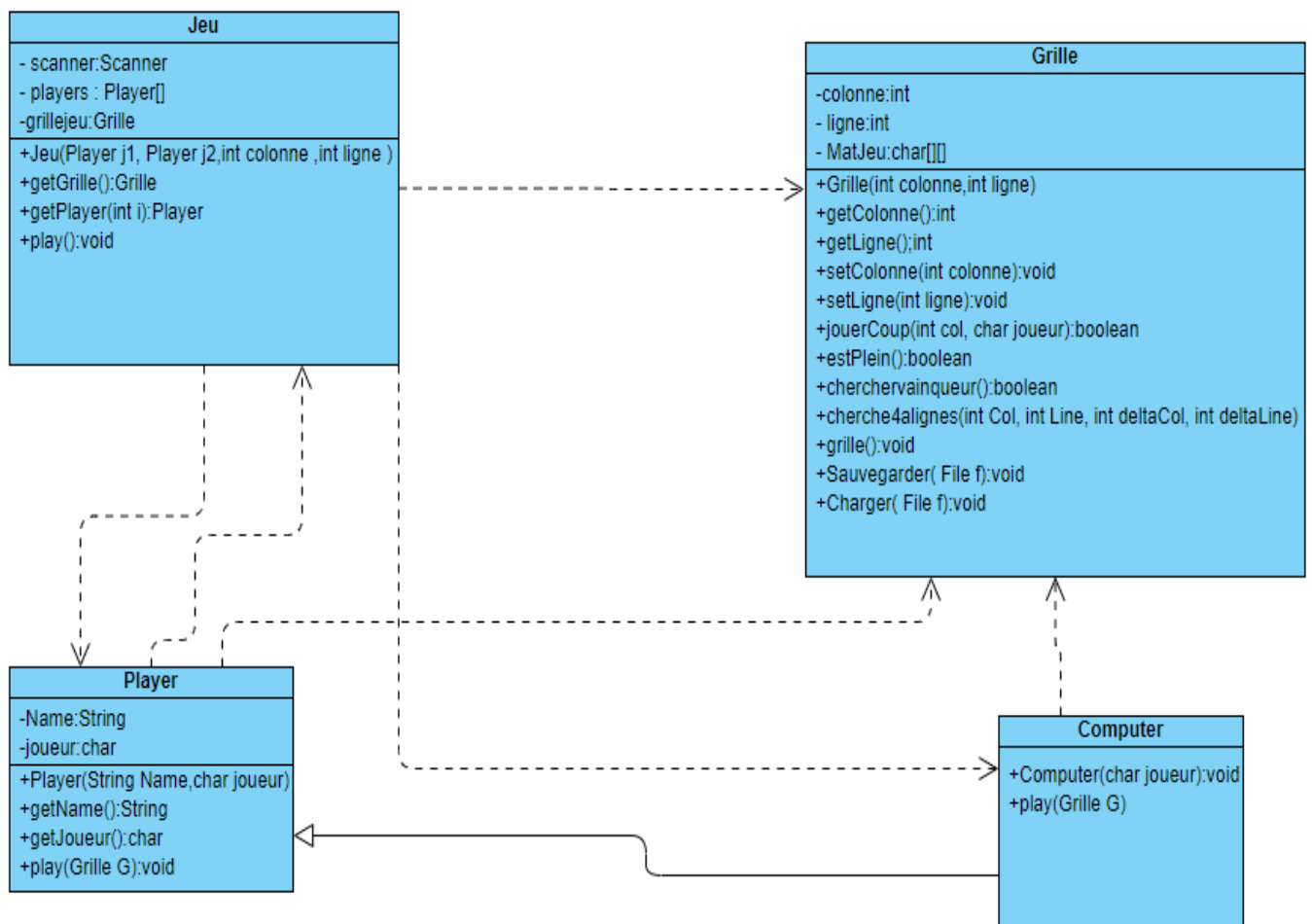


Après plus de réflexion et à partir des différentes idées et améliorations au cours du codage le diagramme a été changé 3 fois avec l'apparition de nouvelles classes et la simplification de plusieurs d'entre elles, car nous nous sommes rendus

compte de plusieurs cas non traiter dès le premier diagramme.

Voici si dessous le diagramme final qui regroupe tout notre travail et la forme finale de notre projet.

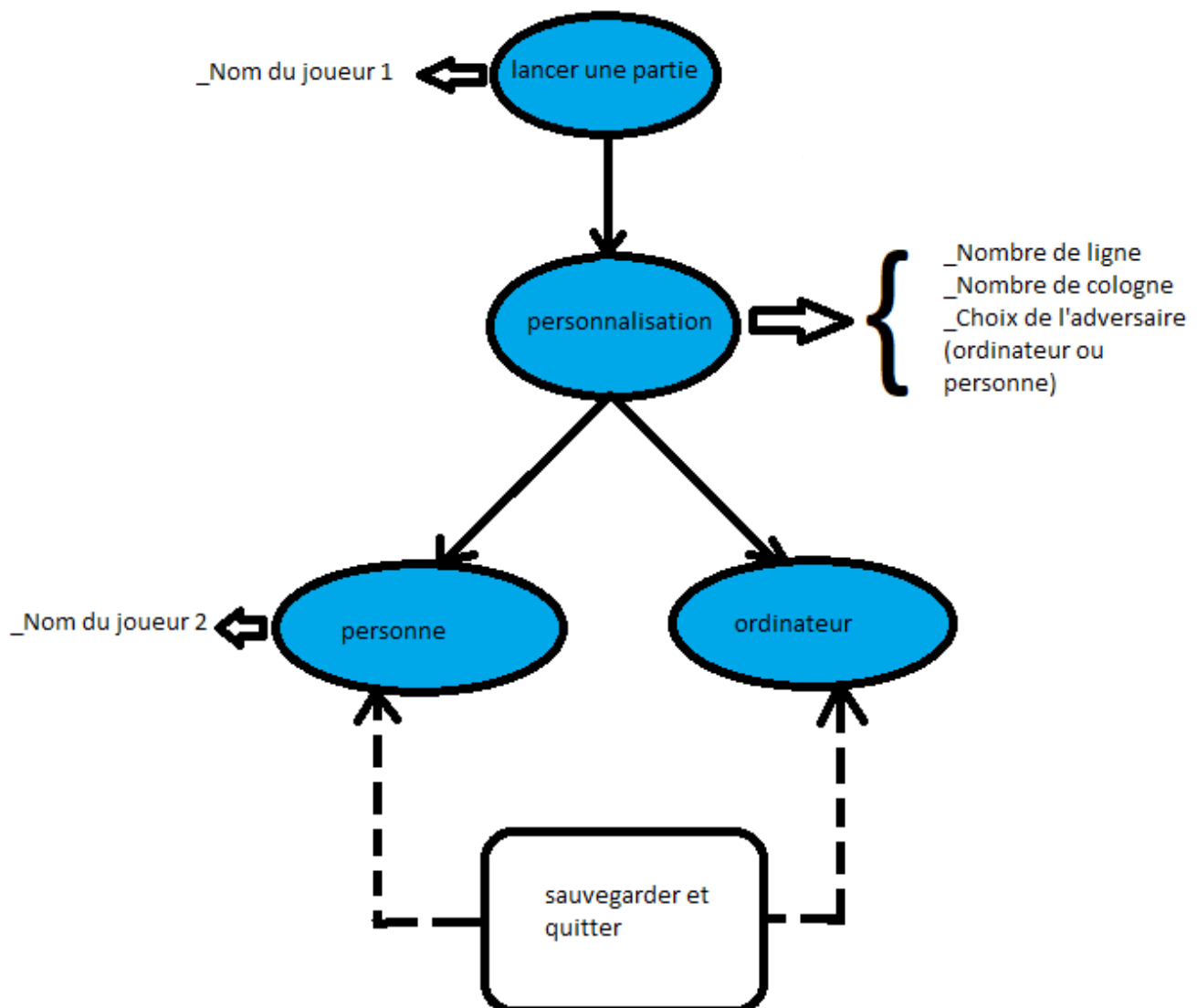
Diagramme final



Étapes de fonctionnement :

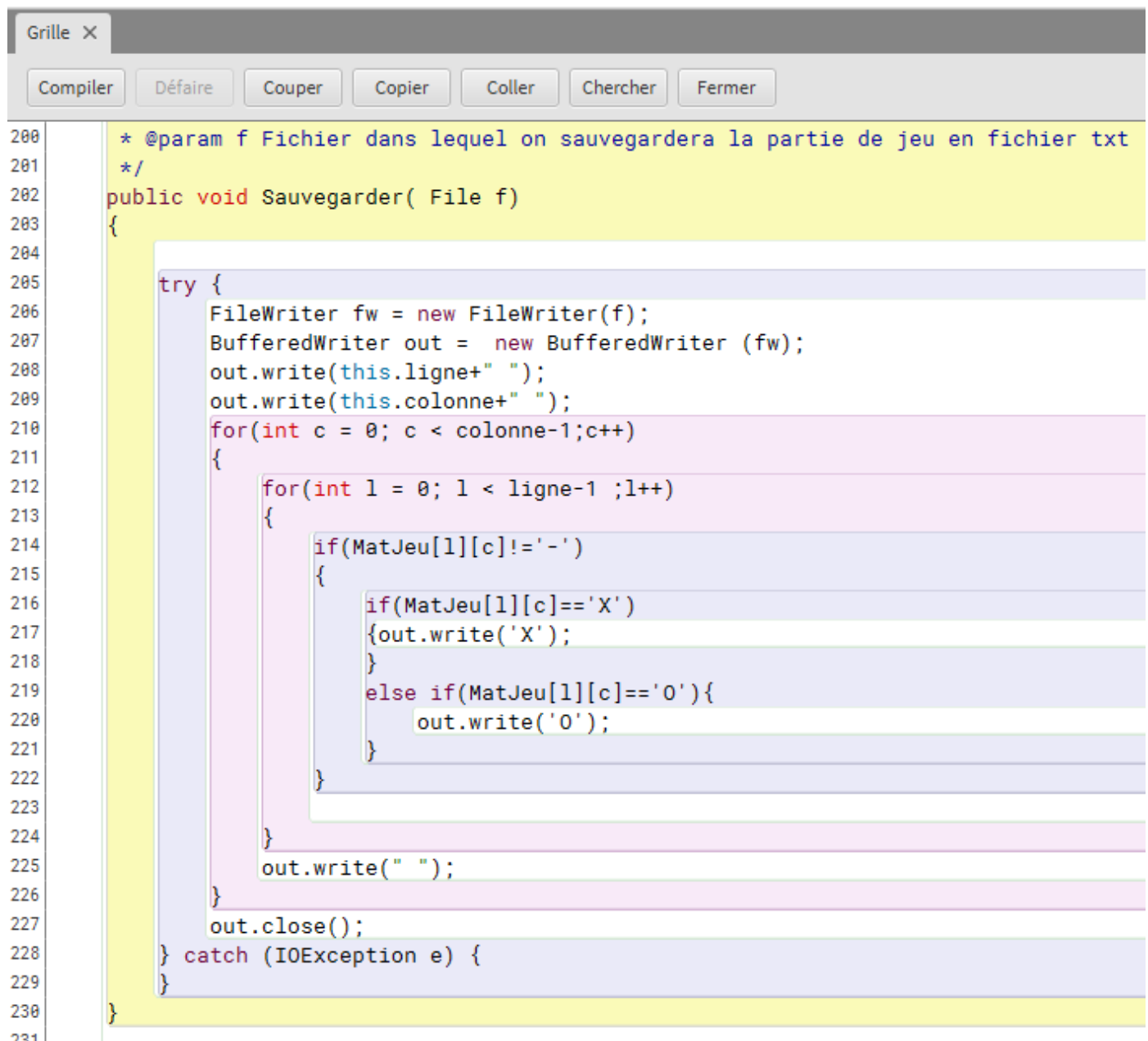
lancer la partie puis on va vous demander de saisir un nom de joueur ainsi que les dimensions voulue de la grille (nombre de colonnes ,nombre de ligne) ,après il suffit de choisir si on veut jouer contre l'ordinateur, dans le cas ou sa seras non une partie joueur contre joueur est automatiquement lancer ,ainsi le jeux demanderas de saisir le nom du deuxième joueur ,notons que a chaque étapes de la partie un message de sauvegarde s'affichera qui permettras de sauvegarder a la parties de jeux actuel.

Diagramme d'utilisation



Fonctionnalités pertinentes

Parmi les fonctionnalités les plus pertinentes de notre programme en retrouveras la fonction de sauvegarde (figure 1), celle qui cherche une combinaison gagnante (figure 2) et la méthode principale de lancement du jeu (figure 3).



```
200  * @param f Fichier dans lequel on sauvegardera la partie de jeu en fichier txt
201  */
202  public void Sauvegarder( File f)
203  {
204
205      try {
206          FileWriter fw = new FileWriter(f);
207          BufferedWriter out = new BufferedWriter (fw);
208          out.write(this.ligne+" ");
209          out.write(this.colonne+" ");
210          for(int c = 0; c < colonne-1;c++)
211          {
212              for(int l = 0; l < ligne-1 ;l++)
213              {
214                  if(MatJeu[l][c]!='-')
215                  {
216                      if(MatJeu[l][c]=='X')
217                      {out.write('X');}
218                  }
219                  else if(MatJeu[l][c]=='O'){
220                      out.write('O');
221                  }
222              }
223          }
224          out.write(" ");
225      }
226      out.close();
227  } catch (IOException e) {
228  }
229  }
230
231
```

Figure 1

```
121 private boolean cherche4alignes(int Col, int Line, int deltaCol, int deltaLine) {
122     char joueur = '-';
123     int compteur = 0;
124
125     int C = Col;
126     int R = Line;
127
128     while ((C >= 0) && (C < colonne-1) && (R >= 0) && (R < ligne-1)) {
129         if (MatJeu[R][C] != joueur) {
130             joueur = MatJeu[R][C];
131             compteur = 1;
132         } else {
133             compteur++;
134         }
135         if ((joueur != '-') && (compteur == 4)) {
136             return true;
137         }
138
139         C += deltaCol;
140
141         R += deltaLine;
142     }
143
144     return false;
145 }
```

Figure 2


```

24 public void play() {
25     int vainqueur = -1;
26     int JoueurCourant = 0;
27
28     while (vainqueur == -1) {
29         players[JoueurCourant].play(grilleJeu);
30         if (grilleJeu.estPlein()) {
31             vainqueur = -1;
32         }
33
34
35         if (grilleJeu.chercheVainqueur()) {
36             vainqueur = JoueurCourant;
37         }
38
39         JoueurCourant++;
40         JoueurCourant %= 2;
41     }
42
43     System.out.println("Partie terminée!!");
44     grilleJeu.grille();
45     if (vainqueur == -1) {
46         System.out.println("Match nul!!");
47     } else {
48         System.out.println("Le vainqueur est " + players[vainqueur].getName());
49     }
50 }
51

```

Figure 3

3) Méthodologie :

1) ABDELLI Idir :

Pour le projet « puissance 4 », en premier lieu nous avons cherché à comprendre le principe du jeu puis après avoir assisté à la séance destinée au projet, nous avons élaboré une conception de base qui est notre 1^{er} diagramme de classe en UML pour avoir une idée de la manière de l'implémentation du programme. Celui-ci nous a permis de nous organiser pour répartir les tâches d'une manière équitable.

2) AMOURA Gaya :

La méthodologie globale s'est mise en place après la séance dédiée au projet, dès lors, la priorité fut de mettre en place une conception UML. S'en est suivi la réalisation d'un diagramme de cas d'utilisation, puis nous avons alors commencé à implémenter les différentes classes du programme, en ayant au préalable mis en place un dépôt GIT.

3) Répartitions des tâches

ABDELLI Idir :

mon rôle était de mettre en place la conception de base du jeu donc construire la classe d'où le jeu devait être lancé qui était la classe jeu, puis j'ai implémenté deux autres classes qui sont « Player » et « Computer » que au début j'ai implémenté indépendamment puis après avoir transmis à mon binôme il m'a suggéré d'utiliser la notion d'héritage qui m'a facilité le codage, à la fin des trois classes de base mon camarade avait corrigé et encore simplifié l'UML le tout était à faire dans une seule et dernière classe où se trouvent toutes les méthodes du principe du jeu la classe « Grille », on a revu la structure mise en place ensemble et ainsi nous avons fait le début de la classe c'est-à-dire la première méthode exemple la méthode qui détermine si la grille est pleine juste pour qu'on soit d'accord sur le principe de fonctionnement que j'ai mis à la base, puis je lui ai laissé continuer les méthodes qui implémentaient les règles du jeu et les différentes options demander telle que la sauvegarde, entre temps j'ai entamé le rapport tout en gardant un œil sur la suite du codage.

AMOURA Gaya :

Après nous nous sommes mis d'accord avec mon binôme sur la structure de base, je me suis concentrer sur les détail du jeu comment chaque règle devait être implémenter et être vérifier a la bonne étapes du programme, j'ai élaborer alors le schéma d'une classe qui contenait l'intégralité des consigne du jeu , mon ami étant charge de construire le lancement et la création des deux personnage qui vont s'affronter ,en ayant consulter son code je lui est suggérer une amélioration qui a était rapidement faite , puis nous avons entamer ensemble le début de la classe principale.

Ensuite, j'ai commencé à chercher des algorithmes à propos de la sauvegarde et du chargement des parties et aussi commencé les améliorations d'affichage et de mise en place des options de jeu.

Difficultés rencontrées :

Au début nous avons eu des difficultés sur comment implémenter la grille et les différents symboles la composant, ensuite grâce a la séance dédié au projet nous avons eu l'idée de le faire avec un tableau a deux dimensions.

Au fur et à mesure de l'avancement du codage nous nous sommes heurtés à un autre problème celui de traite toutes les combinaisons gagnantes surtout sur les combinaisons en diagonale.

Arriver le moment d'entamer la partie sauvegarde et chargement d'une partie nous nous sommes confrontés à un nouveau problème de taille, en effet le fait de ne pas avoir vu en cour ses deux notions nous a grandement bloquer car on est passé d'une

implémentation a un travaille de recherche pour acquérir les principes permettant de faire cela.

Le fait de ne pas avoir pu implémenter les algorithmes d'arbre n-aire en cour nous a réellement bloquer et manque de temps nous n'avons pas pu réaliser la partie MinMax.

Conclusion :

Ce travail nous a permis d'acquérir encore de nouvelles compétences et de consolider celles déjà en notre possession et nous a donné la chance dévaluer nos propres connaissances, nous a aussi appris à mieux analyser et résoudre certains problèmes de programmation.

Sources :

Lien vers le

GIT :<https://git-etudiants.lacl.fr/gaya.amoura/puissance4.git>

Lien de video youtube : <https://www.youtube.com/watch?v=XcXrWsQo4Ng&t=5s>

<https://www.youtube.com/watch?v=8TSVW7SV0KA&list=PLrSOXFDHBtfHkq8dd3BbSaopVgRSYtgPv&index=18>

