Quarto

HLIN405 - T.E.R.

Présentation du Quarto

Le Quarto se joue à deux joueur et est composé de :

- Un plateau de 16 cases (4 par 4)

- 16 pièces différentes, suivant 4 caractéristiques :
  - la couleur (claire ou foncée)
  - la hauteur (haute ou basse)
  - le sommet (pleine ou creuse)
  - la forme (ronde ou carrée).

<u>But du jeu :</u> être le premier à aligner 4 pièces

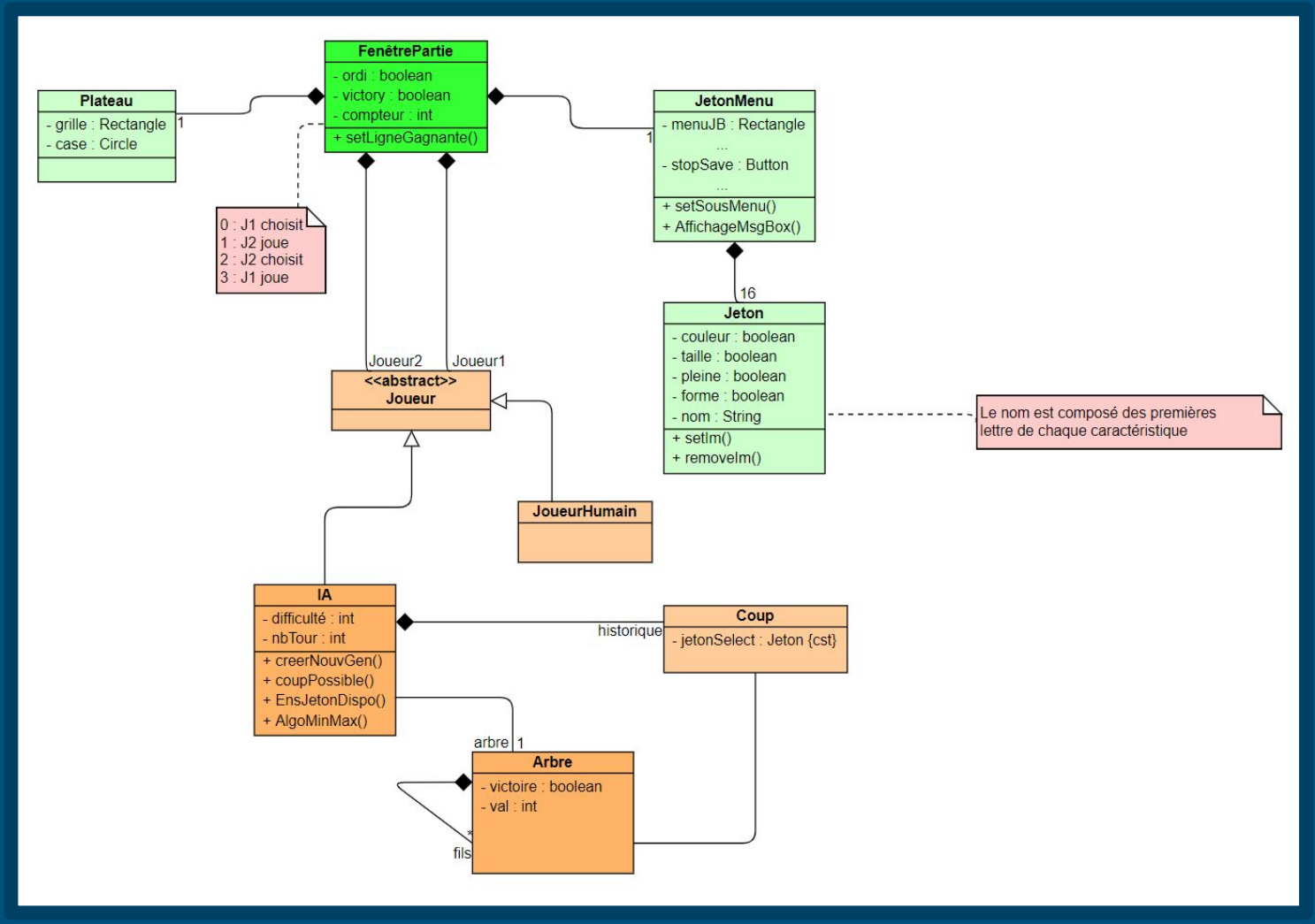
ayant une caractéristique commune

La partie se joue tour par tour. Un tour se déroule en 2 phase :

- Jouer un pion sur une case libre (le pion est imposé par l'adversaire)
- Choisir le pion que jouera ensuite l'adversaire.

Sommaire

La Quarto
Diagramme UML



Organisation du projet

Aspect graphique : différents outils

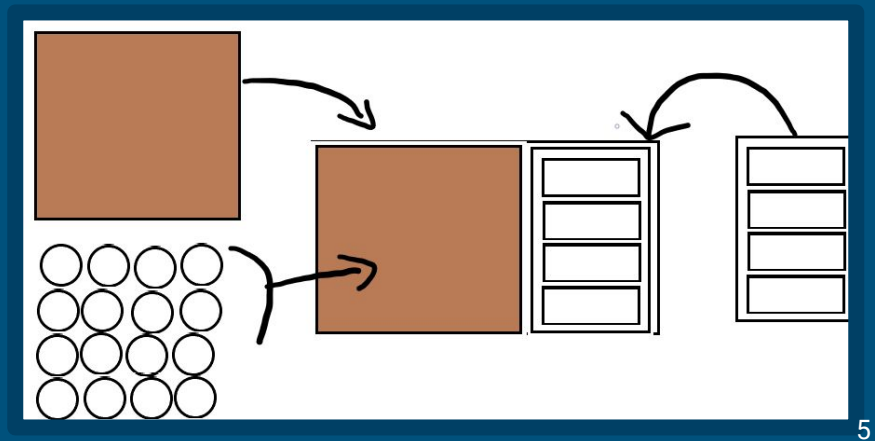
Nous avons utilisé la bibliothèque JavaFX pour réaliser ce projet, on le verra plus en détails ci-après.

Les outils : Eclipse IDE Java

Photoshop → Jeton 2D : 

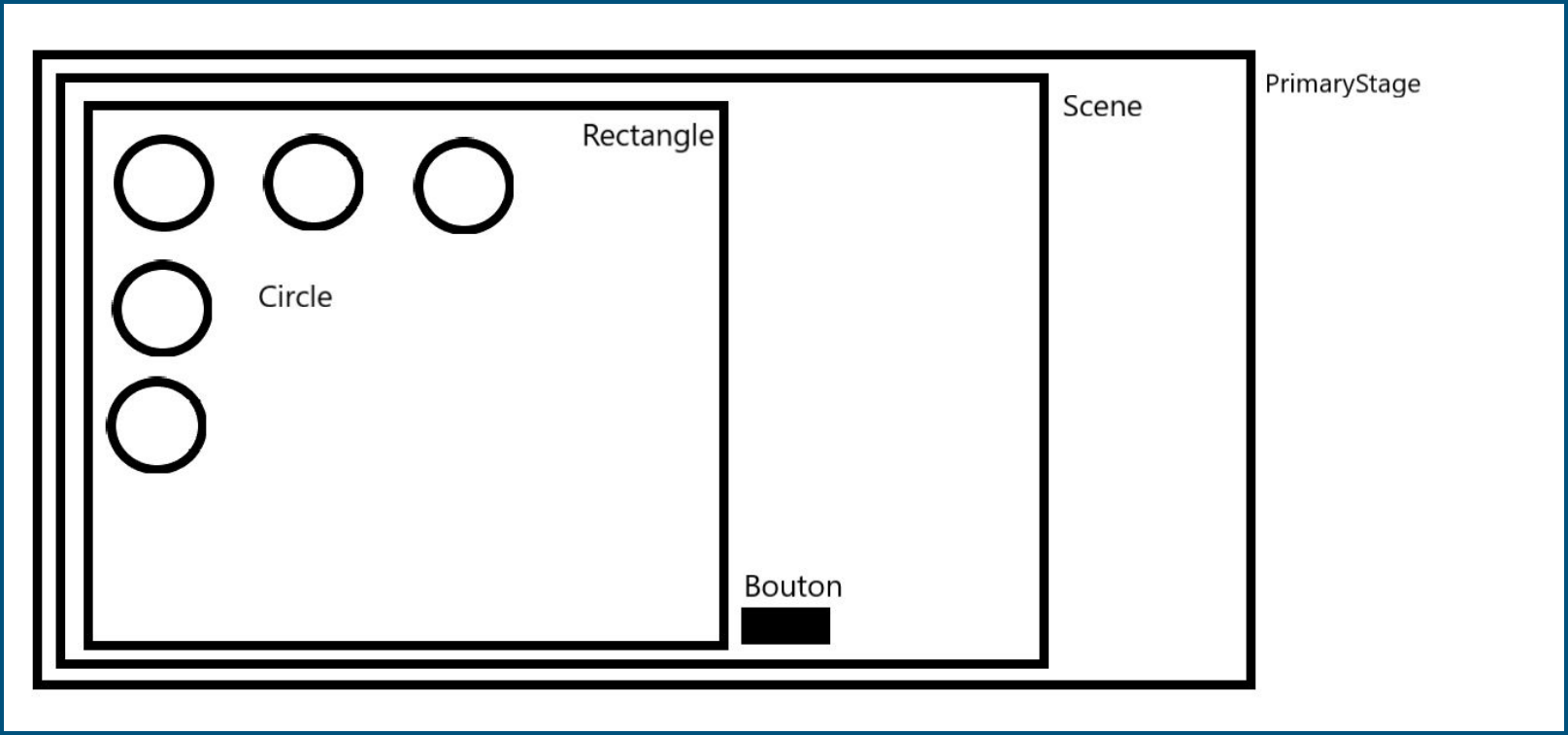
Cinema 4D → Jeton 3D : 

JavaFX : niveau développement du jeu, le Quarto est composé de : un plateau, un menu latéral, des jetons et un affichage.



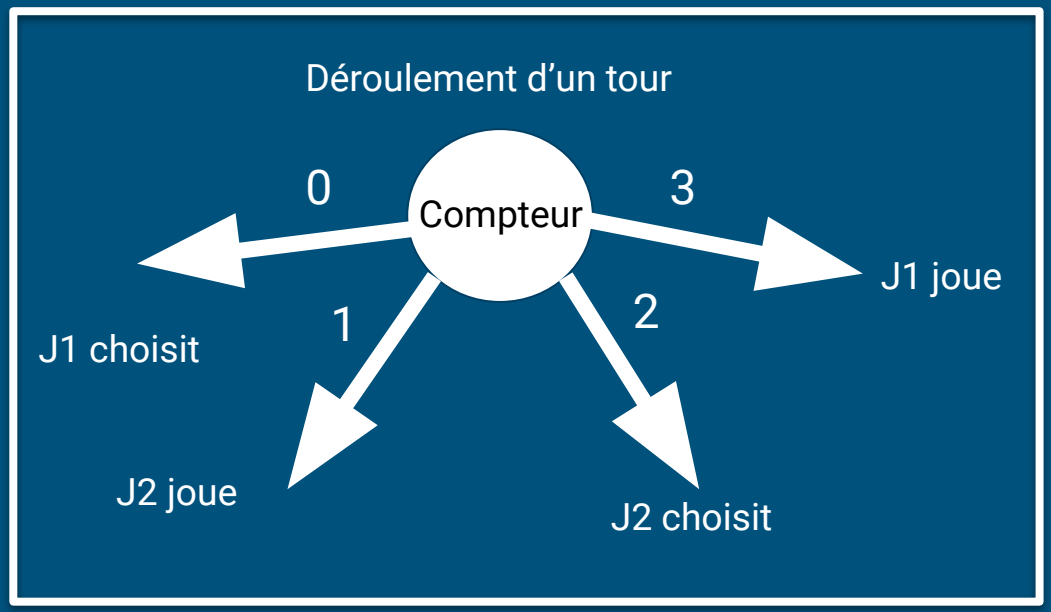
Organisation du projet
Aspect graphique : JavaFX

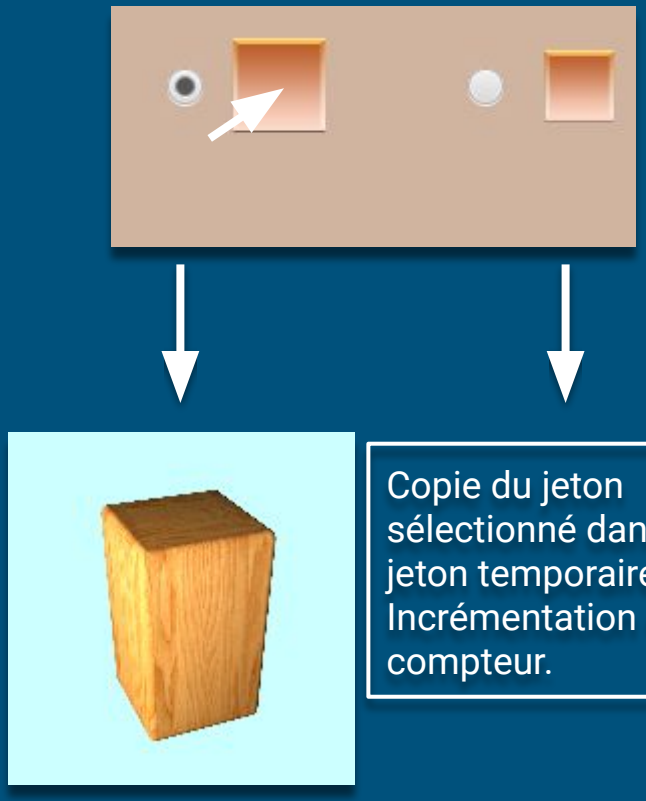
javafx.graphics : définit l'API relative aux conteneurs et support CSS



Tous les éléments sont groupés au même parent "root" avec la méthode : root.getChildren().add(myObject);

Déroulement d'une partie
Choix du jeton et compteur





Copie du jeton sélectionné dans un jeton temporaire. Incrémentation du compteur.

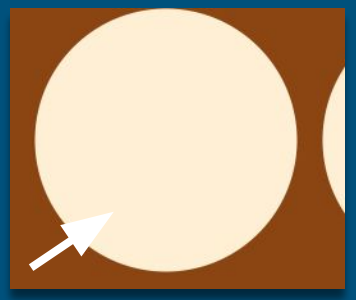
Déroulement d'une partie
Event, pose du jeton et victoire

Exemple d'event

```
select.setOnMouseClicked(e->{

    int colonne = (int)(e.getX() / (selec
    int rang = (int)(e.getY() / (select.g

    if (jetonTmp != null && !getGrilleJet
```

Event problématique

```
//on ajoute un evenement à chaque bouton lorsq
tableauBouton[0].setOnMouseClicked(e->jetonSele
tableauBouton[1].setOnMouseClicked(e->jetonSele
tableauBouton[2].setOnMouseClicked(e->jetonSele
tableauBouton[3].setOnMouseClicked(e->jetonSele
tableauBouton[4].setOnMouseClicked(e->jetonSele
tableauBouton[5].setOnMouseClicked(e->jetonSele
```



Vérifications

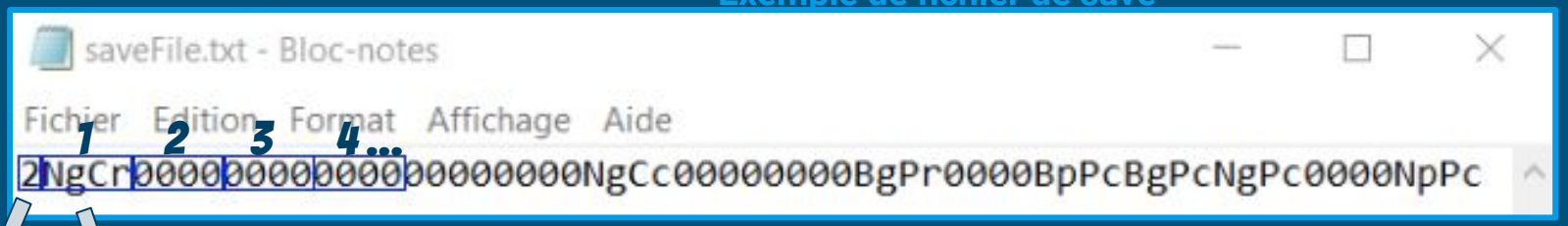
Récupération des coordonnées

Pose du jeton

Victoire ?

Organisation du projet
Sauvegarde de la partie

Exemple de fichier de save



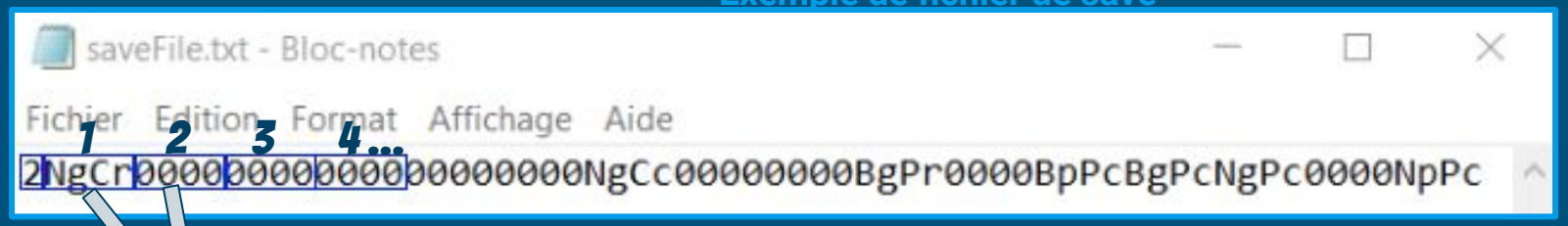
J2 Choisit

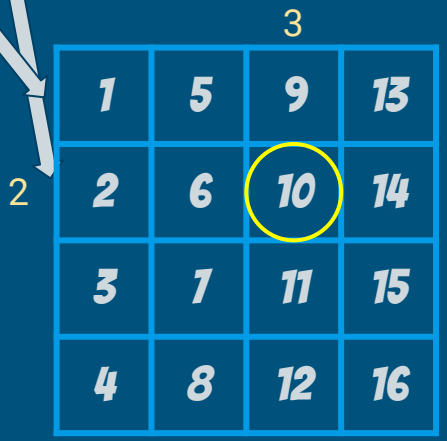
1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

```
this.getJetonMenu().getStopSave().setOnAction((ActionEvent e) ->{

    if(!victory) {
        try {
            File saveFile = new File("saveFile.txt");

            PrintWriter writer = new PrintWriter(saveFile);
            writer.print(compteur);
            for(int i=0;i<4;i++) {
                for(int j=0;j<4;j++) {
                    if(this.getGrilleJetonsPlacesIJ(i,j).getInit()) {
                        writer.print(this.getGrilleJetonsPlacesIJ(i, j).getName());
                        }
                    else {
                        writer.print("0000");
                    }
                }
            }
            writer.close();
            } catch (IOException e1) {
                // TODO Bloc catch gï¿½nï¿½rï¿½ automatiquement
                e1.printStackTrace();
            }
        root.getChildren().clear();
        Main.adMenu();
    }

});
```

Organisation du projet
Chargement de la partie

Exemple de fichier de save





colonne(10) = 3

```
public final int colum(int x) {
    if(x%4 == 0) {
        return (x/4)-1;
    }
    else {
        return x/4;
    }
}
```

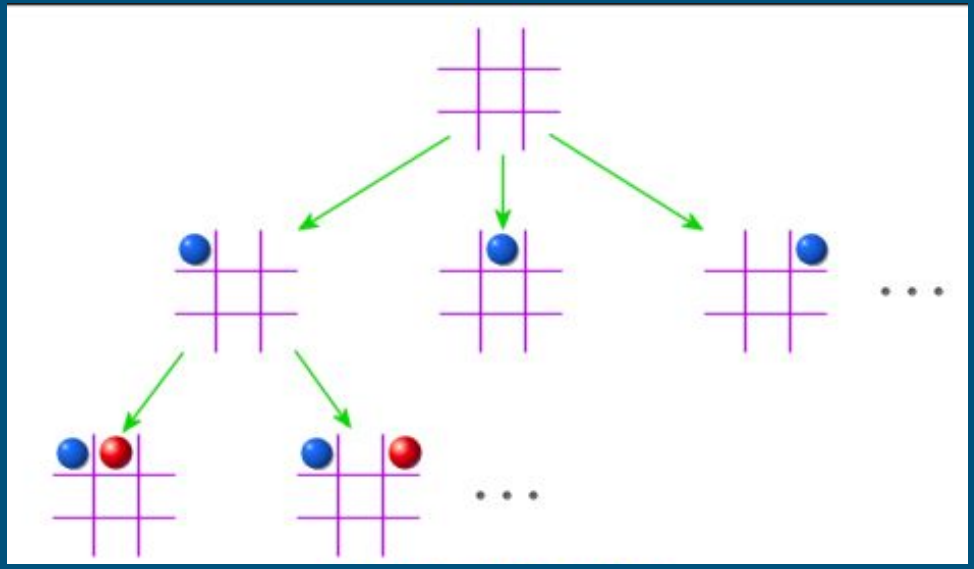
ligne(10) = 2

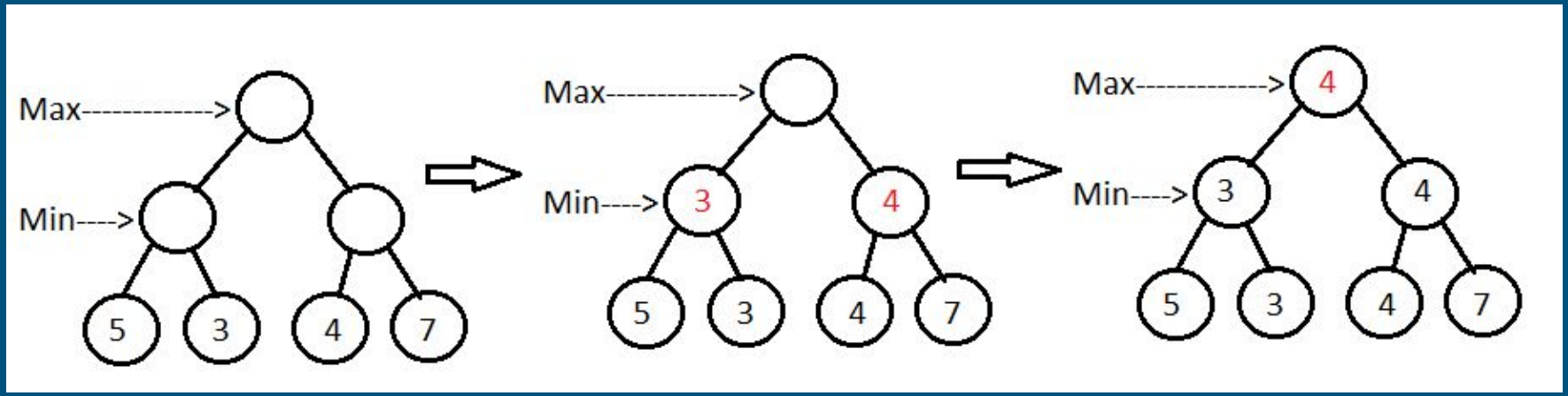
```
public final int ligne(int x) {
    if(x%4 == 0) {
        return 3;
    }
    else {
        return (x%4)-1;
    }
}
```

```
public void chargementPartie() {
    InputStreamReader characterStream = null;
    try {
        FileInputStream byteStream = new FileInputStream("saveFile.txt");

        characterStream = new InputStreamReader(byteStream, "UTF8");
        int character;
        int i = 0; int j = 1;
        String temp = "";
        while ((character = characterStream.read()) != -1) {
            if(i==0) { //Remise du compteur
                compteur = Character.getNumericValue((char)character);
                jetonMenu.affichageMessageBox(compteur,fenY);
            }
            else {
                temp+=(char)character;
                if(i==4) {
                    if(!temp.equals("0000")) {
                        int k=0;
                        while(!temp.equals(JetonMenu.getTableauDeJeton(k).getName())) {
                            k++;
                        }
                        jetonTmp = JetonMenu.getTableauDeJeton(k);
                        String l = jetonTmp.getIm();
                        Image im = new Image(l);
                        getGrilleJetonsPlacesIJ(colum(j),ligne(j)).setCouleur(jetonTmp.getCouleur());
                        getGrilleJetonsPlacesIJ(colum(j),ligne(j)).setTaille(jetonTmp.getTaille());
                        getGrilleJetonsPlacesIJ(colum(j),ligne(j)).setPleine(jetonTmp.getPleine());
                        getGrilleJetonsPlacesIJ(colum(j),ligne(j)).setForme(jetonTmp.getForme());
                        getGrilleJetonsPlacesIJ(colum(j),ligne(j)).setInit(true);
                        getGrilleJetonsPlacesIJ(colum(j),ligne(j)).putThisImage(im);
                        getGrilleJetonsPlacesIJ(colum(j),ligne(j)).setOpacity(1);

                        jetonMenu.getTableauBouton(k).setOpacity(0.3);
                        jetonMenu.getTableauBouton(k).setDisable(true);
                        jetonMenu.getTableauBouton(k).setSelected(false);

                        jetonTmp=null;

                    }
                    temp = "";
                    j++;
                }
            }

            if(i!=4) {i++;}
            else {i=1;}

        }
    }catch (IOException e1) {
        // TODO Bloc catch gï¿½nï¿½rï¿½ automatiquement
        e1.printStackTrace();
    }
}
```

L'Intelligence Artificielle
Algorithme MinMax

- Simulation de toutes les issues possibles
- Sélection du Coup le plus intéressant





L'Intelligence Artificielle
Réalisation en Java

Classes Coup et Arbre :

```java
public class Coup {

    private final byte x;
    private final byte y;

    private final Jeton jetonSelect;

    public Coup(int x, int y, Jeton jetonSelect) {
        this.x = (byte) x;
        this.y = (byte) y;
        this.jetonSelect = jetonSelect;
    }
```

```java
public class Arbre {

    private Coup coup;
    private byte val;
    private boolean victoire;
    private ArrayList<Arbre> fils;

    public Arbre() {
        this.coup = null;
        this.val = 0;
        this.fils = new ArrayList<>();
        this.victoire = false;

    }

    public Arbre(Coup coup) {
        this.coup = coup;
        this.val = 0;
        this.fils = new ArrayList<Arbre>();
        this.victoire = false;
    }
```

```java
private int algoMinMax (Arbre noeud, boolean tourIA) {

    if (noeud.getFils() == null || noeud.getFils().isEmpty()) {
        return noeud.coupGagnant() ? (tourIA ? 100 : -100) : 0;
    }

    int tmp;

    if (!tourIA) {

        for (Arbre fils : noeud.getFils()) {

            tmp = algoMinMax(fils, true);
        }

        noeud.setVal(noeud.getFils().get(0).getVal()) ;
        for (Arbre fils : noeud.getFils()) {

            tmp = fils.getVal();

            if (noeud.getVal()>tmp) {
                noeud.setVal(tmp);
            }
        }

    }

    else {

        for (Arbre fils : noeud.getFils()) {

            tmp = algoMinMax(fils, false);
        }

        noeud.setVal(noeud.getFils().get(0).getVal()) ;
        for (Arbre fils : noeud.getFils()) {

            tmp = fils.getVal();

            if (noeud.getVal()<tmp) {
                noeud.setVal(tmp);
            }
        }

    }
```

```java
public Coup jouerTour(Coup c) {

    if (nbTour >= 16) {
        return null;
    }

    nbTour++;
    historique.add(c);
    Coup tmp;

    if (nbTour < 4) {
        tmp = coupAlea();
    }


    else {

        if (nbTour == 5 || nbTour == 4) {
            initialiserArbre();
        }
        else {
            arbre = Arbre.getFils(arbre, c);
            creerNouvGen(arbre, historique);
        }

        algoMinMax(arbre, true);
        tmp = meilleurCoup();

        arbre = Arbre.getFils(arbre, tmp);
        creerNouvGen(arbre, historique);

    }

    historique.add(tmp);
    nbTour++;

    return tmp;
}
```

Perspectives et démonstration

- Options : Thème sombre avec CSS par exemple
- Interface 3D
- Niveau de difficulté plus élevé pour l’IA
- Sauvegarde de partie contre l’ordinateur
- Musique d’ambiance et son interactif

FIN

Nous remercions Vincent Boudet de nous avoir encadré tout au long de ce semestre.

Merci de votre attention !

CAPRIO LOMBARDI Thomas, DUCHON Damien, GOLDSTEIN Solal,

GUILLEN Victor et QUINTANE Alexis