

Mini-projet LU2IN002 - 2020-2021

<i>Nom</i> : Balasubramaniam	<i>Nom</i> :
<i>Prénom</i> : Rishi	<i>Prénom</i> :
<i>N° étudiant</i> : 3671268	<i>N° étudiant</i> :

Thème de simulation choisi (en 2 lignes max.)

La simulation de la récolte de pièces par des pirates et sa transformation en nourritures.

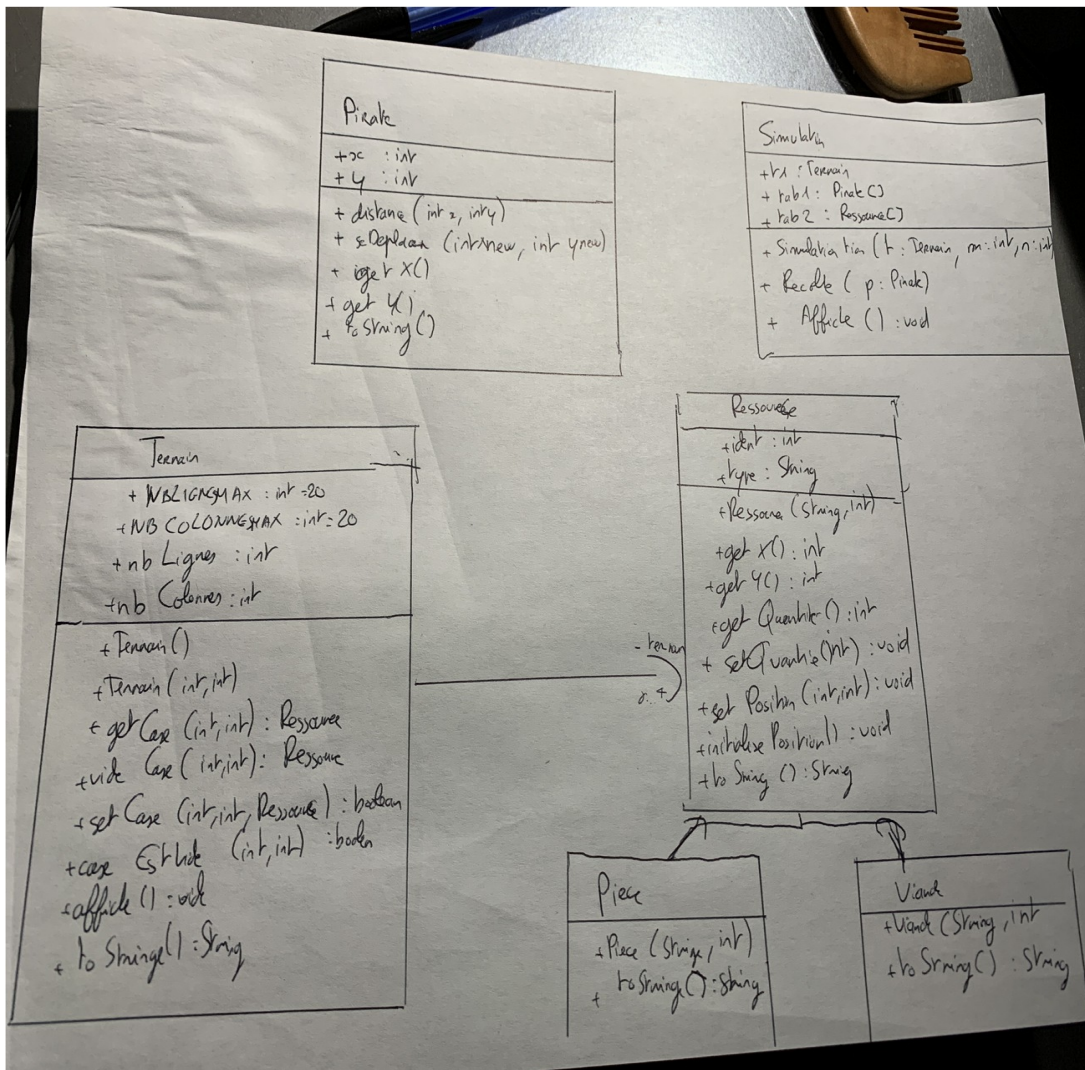
Description des classes et de leur rôle dans la simulation (2 lignes max par classe)

Les agents sont les pirates qui se déplacent sur le terrain qui contient des pièces et les échanges contre de la viande. Les pièces et la viande sont des ressources.

Décrire, dans les grandes lignes, ce qui se passe durant la simulation (max. 5-6 lignes)

On essaye de créer plusieurs agents et de ressources et de voir les statistiques de la quantité qui a pu être récoltée. Chaque fois qu'un agent qui est un pirate dans mon cas, passe dans la même case que qu'une ressource "pièce" elle l'a récolte et produit de la viande dans la case où il y avait la pièce.

Schéma UML fournisseur des classes (dessin "à la main" scanné ou photo acceptés)



--

<i>Checklist des contraintes prises en compte:</i>	<i>Nom(s) des classe(s) correspondante(s)</i>
Classe contenant un tableau ou une liste d'objets	Simulation, Terrain
Classe statique contenant que des méthodes statiques	TestTerrain, TestSimulation
Héritage	Piece,Viande
Classe avec composition	
Classe avec un constructeur par copie ou clone()	Non
Noms des classes créées (entre 4 et 10 classes)	5

Copier / coller de vos classes à partir d'ici :

```
public class Pirate{
    private int x;
    private int y;

    public double distance(int x, int y){
        return Math.sqrt(Math.pow((x-this.x),2)+Math.pow((y-this.y),2));
    }

    public void seDeplacer(int xnew,int ynew){
        this.x = xnew;
        this.y = ynew;
    }

    public int getX(){
        return x;
    }

    public int getY(){
        return y;
    }

    public String toString(){
        return "("+x+", "+y+")";
    }
}

public class Simulation{
    private Terrain t1;
    public Pirate[] tab1;
    public Ressource[] tab2;

    public Simulation(Terrain t,int m,int n){
        tab1 = new Pirate[n];
        tab2 = new Ressource[m];
        this.t1 = t;

        for(int i=0; i<tab2.length;i++){
            tab2[i] = new Piece("Piece",(int)(Math.random()*(1000+1)));
            int lig = (int)(Math.random()*(t1.nbLignes));
            int col = (int)(Math.random()*(t1.nbColonnes));
            while((t1.caseEstVide(lig,col) == false)){
                lig = (int)(Math.random()*(t1.nbLignes));
                col = (int)(Math.random()*(t1.nbColonnes));
            }
            t1.setCase(lig,col,tab2[i]);
            System.out.print(lig+" ");
            System.out.println(col);
        }
    }
}
```

```

    }

    for(int j=0;j < tab1.length;j++){
        tab1[j] = new Pirate();
    }
}

public void Recolte( Pirate p){
    if((t1.caseEstVide(p.getX(),p.getY()) == false) && (t1.getCas(p.getX(),p.getY())
instanceof Piece)){
        Ressource r1 = t1.getCas(p.getX(),p.getY());
        System.out.println(r1.toString());
        r1.initialisePosition();
        Ressource r2 = new Viande("Viande",(int)(Math.random()*(1000+1)));
        t1.setCas(p.getX(),p.getY(),r2);
        System.out.println(r2.toString());
    }

}

public void Affiche(){
    t1.affiche();
}

}

```

```

public class TestSimulation{
    public static void main(String[] args){
        Terrain t = new Terrain(5,5);
        Simulation s1;
        s1 = new Simulation(t,10,10);
        s1.Affiche();

        System.out.println(s1.tab1[0].toString());
        s1.tab1[0].seDeplacer(1,1);
        System.out.println(s1.tab1[0].toString());

        Ressource r1 = new Piece("Piece",10);
        Ressource r2 = new Viande("Viande",2);

        System.out.println(r1.toString());
        System.out.println(r2.toString());
        System.out.println(s1.tab2[0].toString());

        s1.Recolte(s1.tab1[0]);
        s1.Affiche();
    }
}

```

```
    }  
}
```

```
public class Piece extends Ressource{  
    public Piece(java.lang.String type, int quantite){  
        super(type,quantite);  
    }  
  
    public String toString(){  
        return super.toString();  
    }  
}
```

```
public class Viande extends Ressource{  
    public Viande(java.lang.String type, int quantite){  
        super(type,quantite);  
    }  
  
    public String toString(){  
        return super.toString();  
    }  
}
```