Développement d'un jeu

Touché-Coulé

PEYEN Kévin

Rapport de projet

Licence Informatique – 2e année

année 2018-2019

Sommaire

Liste des figures

Figure 1 : Fenêtre du jeu

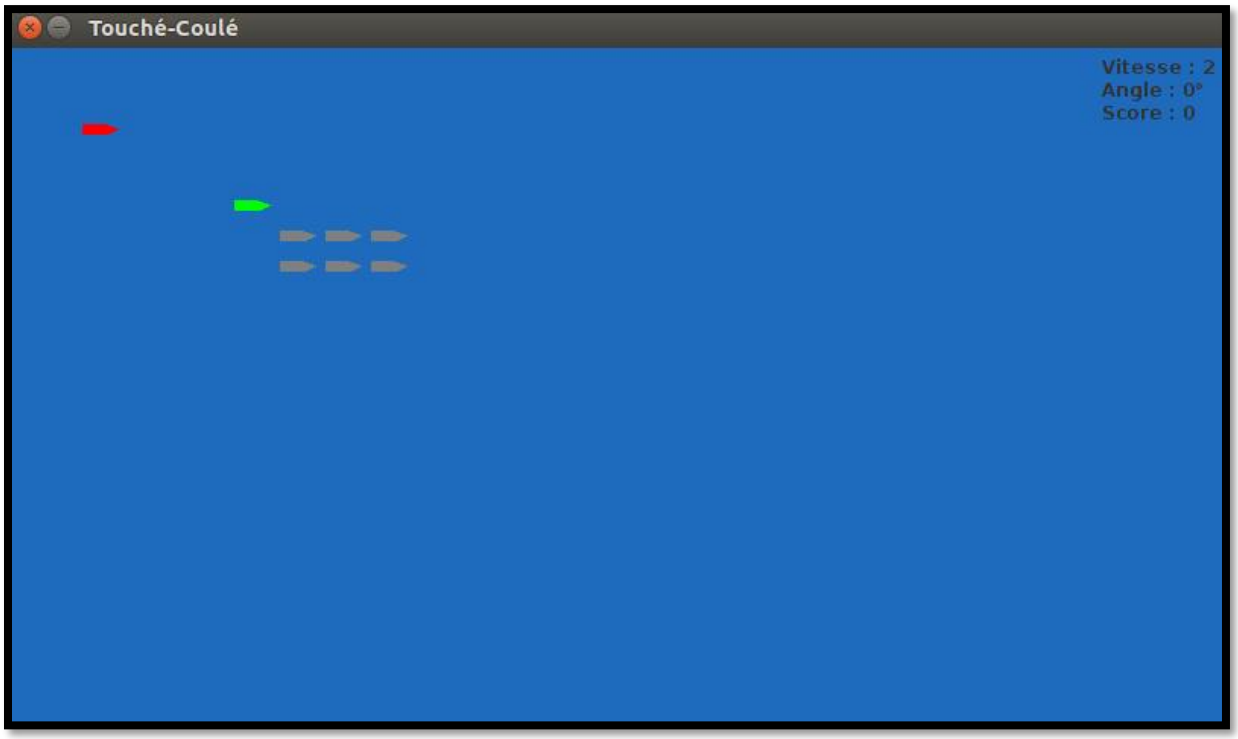

Figure 2 : Diagramme de class

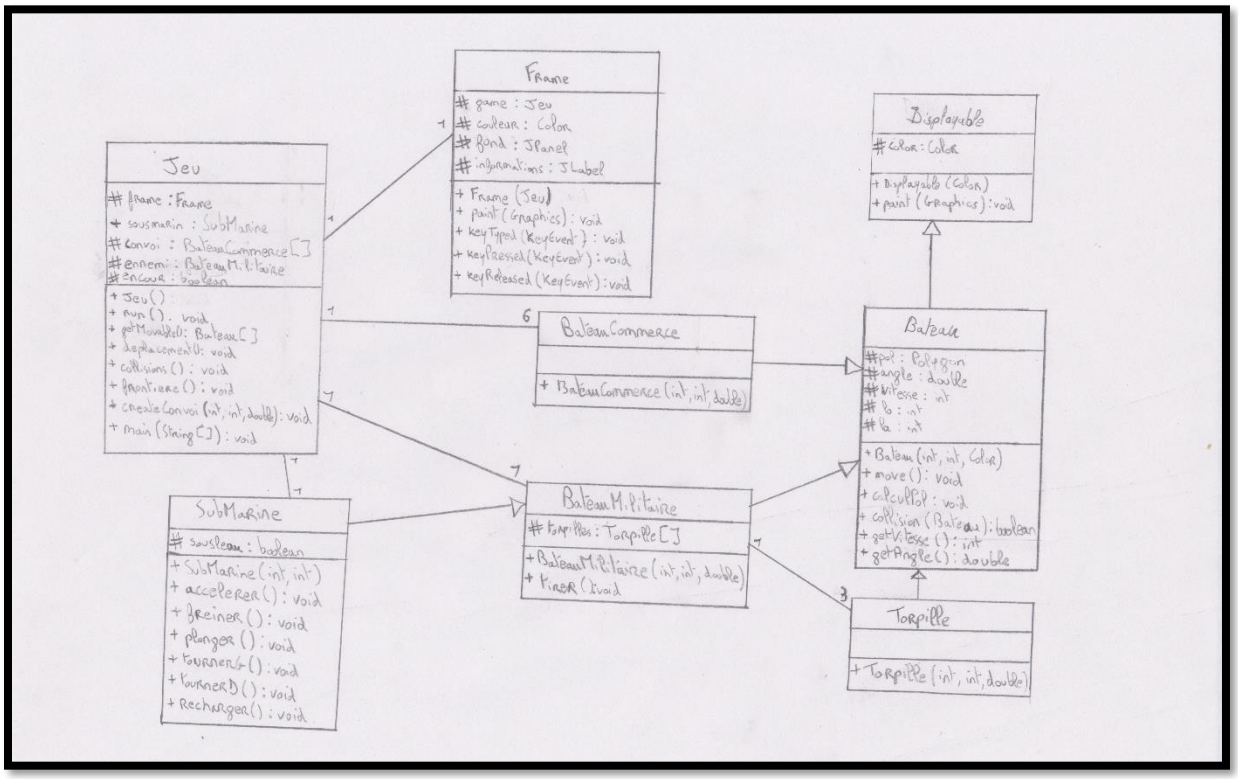

Figure 3 : Méthode getMovable

```
public Bateau[] getMovable(){
    int i=0,j=0;
    Bateau[] objet = new Bateau[20];
    // Rajoute le sous marin dans le tableau
    objet[j]=this.sousmarin;
    j++;
    // Rajoute les torpilles du sous marin dans le tableau
    for(i=2;i>=0;i--){
        if(this.sousmarin.torpilles[i]!=null){
            objet[j]=this.sousmarin.torpilles[i];
            j++;
        }
    }
    // Rajoute le bateau ennemi dans le tableau
    if(this.ennemi!=null){
        objet[j]=this.ennemi;
        j++;
    }
    // Rajoute le convoi dans le tableau
    for(i=0;i<6;i++){
        if(this.convoi[i]!=null){
            objet[j]=this.convoi[i];
            j++;
        }
    }
    return objet;
}
```

Figure 4 : Méthode paint

```
public void paint(Graphics gr){
    int i=0;
    gr.setColor(couleur);

    String info="<html><body>Vitesse : "+(game.sousmarin.getVitesse()/5);
    info += "<br>Angle : "+(int)((game.sousmarin.getAngle()*180)/Math.PI)+"°";
    info += "<br> Score : 0</body></html>";
    this.informations.setText(info);
    this.fond.paint(gr); // colorie le fond en bleu

    Bateau[] objet=this.game.getMovable(); // recupere tout les objets du jeu
    while(objet[i]!=null){
        objet[i].paint(gr);     // affiche les objets
        i++;
    }
}
```

Figure 5 : Méthode keyPressed

```
public void keyPressed(KeyEvent e){
    if(e.getKeyCode()==KeyEvent.VK_UP){
        this.game.sousmarin.accelerer();
    }
    if(e.getKeyCode()==KeyEvent.VK_DOWN){
        this.game.sousmarin.freiner();
    }
    if(e.getKeyCode()==KeyEvent.VK_LEFT){
        this.game.sousmarin.tournerG();
    }
    if(e.getKeyCode()==KeyEvent.VK_RIGHT){
        this.game.sousmarin.tournerD();
    }
    if(e.getKeyCode()==KeyEvent.VK_SPACE){
        this.game.sousmarin.tirer();
    }
    if(e.getKeyCode()==KeyEvent.VK_R){
        this.game.sousmarin.recharger();
    }
    if(e.getKeyCode()==KeyEvent.VK_E){
        this.game.sousmarin.plonger();
    }
}
```

Figure 6 : Méthode collision

```
public boolean collision(Bateau bat){
    int i=0;
    for(i=0;i<5;i++){
        if(this.pol.contains(bat.pol.xpoints[i],bat.pol.ypoints[i])){
            System.out.println("Collision ! ");
            return true;
        }
        /*if(this.pol.intersects(bat.pol.xpoints[0],bat.pol.ypoints[0],bat.lo,bat.la)){
            System.out.println("Collision ! inter");
            return true;
        }*/
    }
    return false;
}
```

Introduction

Ce projet m'a était confié dans le cadre de ma 2e année en licence Informatique. L'objectif du projet consiste en la réalisation d'un jeu où le joueur contrôle un sous-marin qui doit couler des navires de commerce. Programmer en programmation objet, il devra respecter les principes de modularité, d'abstraction, d'encapsulation et de documentation.

Dans ce dossier nous verrons tout d'abord une présentation du jeu puis sa réalisation.

I. Présentation du jeu

Dans cette partie nous allons expliciter les règles et le fonctionnement du jeu, Touché-Coulé.

1. Présentation générale

Le jeu consiste à contrôler un sous-marin qui doit couler des navires de commerce. On peut faire varier la direction, la vitesse et la profondeur du sous-marin, et lancer des torpilles (en nombre limité). Le sous-marin ne peut pas descendre au-delà d'une certaine profondeur.

Les navires de commerce se déplacent en convoi, toujours dans la même direction à une vitesse assez faible. Ils sont protégés par un navire militaire qui sert d'éclaireur.

Le sous-marin peut être détecté, soit parce qu'il a tiré une torpille soit parce qu'un navire militaire s'est approché très près du sous-marin et que celui-ci n'était pas immergé. Si le sous-marin est détecté, le navire militaire converge vers son emplacement. Le score du joueur dépend du nombre de bateaux de commerce qu'il arrive à couler.

L'interface graphique permet d'afficher la position des bateaux et leur direction mais également d'afficher la direction, la vitesse et la profondeur du sous-marin.

2. Déroulement du jeu

Lors du lancement du jeu, il commence automatiquement.
Le sous-marin du joueur est représenté par le bateau de couleur rouge, le vert est un navire militaire tant dis que les gris sont les navires de commerce autrement dit, les cibles.

Les flèches haut et bas servent à accélérer ou freiner, il existe 3 modes de vitesses différentes.
Les flèches gauche et droite servent à changer le cap. La touche E permet d'immerger le sous-marin ou de le remonter à la surface mais à condition qu'il soit à l'arrêt.
La touche espace permet de tirer une torpille, limité à 3 torpilles ensuite il faut recharger avec la touche R.

Pour quitter le jeu il suffit de fermer la fenêtre.

Figure 1 : Fenêtre du jeu

II. Réalisation

Dans cette partie nous allons voir certaines class du jeu.

1. Le diagramme de class

Ce diagramme représente les différentes class et leur lien.

Figure 2 : Diagramme de class

2. La class Jeu

Cette class est la class principale du jeu c'est celle qui permet la gestion du jeu.

➔ Sa méthode getMovable() permet de récupérer tous les objets déplaçable du jeu.
- Idée : On appellera cette fonction afin d'effectuer ensuite le mouvement sur le tableau de retour.

Figure 3 : Méthode getMovable

3. La class Frame

Cette class est celle qui représente la fenêtre du jeu.

➔ Sa méthode paint(Graphics) permet toute la partie graphique du jeu dans la fenêtre.
- Idée : Elle fera appel à toute les autres méthodes paint.

Figure 4 : Méthode paint

➔ Sa méthode keyPressed(KeyEvent) permet l'interaction avec le clavier.
- Idée : Selon l'événement détecter elle fera appel a une méthode.

Figure 5 : Méthode keyPressed

4. La class abstraite Bateau

Cette class modélise un bateau qui pourra être afficher.

➔ Sa méthode collision(Bateau) permet la détection de collision entre deux Bateaux.
- Idée : Renvoi vrai si l'un des points du polygone du paramètre est contenu dans l'autre polygone.

Figure 6 : Méthode collision

III. Problème & Améliorations

1. Problème majeur

A ce jour, le problème majeur du projet est la détection de collision. Lorsque le convoi est à l'arrêt, les collisions fonctionnent correctement mais dès qu'il y a du mouvement elle ne fonctionne plus du tout.

2. Améliorations possibles

Plusieurs améliorations sont envisageables :

- ➔ Créé la class Convoi afin de composer un convoi avec un certain nombre de bateau de commerce ou militaire. Et pouvoir ensuite les faire apparaitre aléatoirement selon certains points d'apparitions.
- ➔ Une barre de menu qui permet plusieurs options.
- ➔ Enregistrer les plus gros scores.

Conclusion

Le projet est bien avancé malgré les problèmes rencontrés mais est loin d'être aboutis, il reste encore certaines fonctionnalités à terminer.