



Projet POO : Touché-Coulé

Préparé par : Peyen Kévin

23/04/2019

Sommaire

Introduction

I. Présentation

- 1. Présentation générale
- 2. Déroulement

II. Réalisation

- 1. Diagramme de classes
- 2. Classe Jeu
- 3. Classe Frame
- 4. Classe abstraite Bateau

Problème majeur

Améliorations envisageables

Conclusion

Introduction

Objectifs du projet :

- Développer un jeu
- Respecter les principes de POO



Sommaire

Introduction

I. Présentation

- 1. Présentation générale
- 2. Déroulement

II. Réalisation

- 1. Diagramme de classes
- 2. Classe Jeu
- 3. Classe Frame
- 4. Classe abstraite Bateau

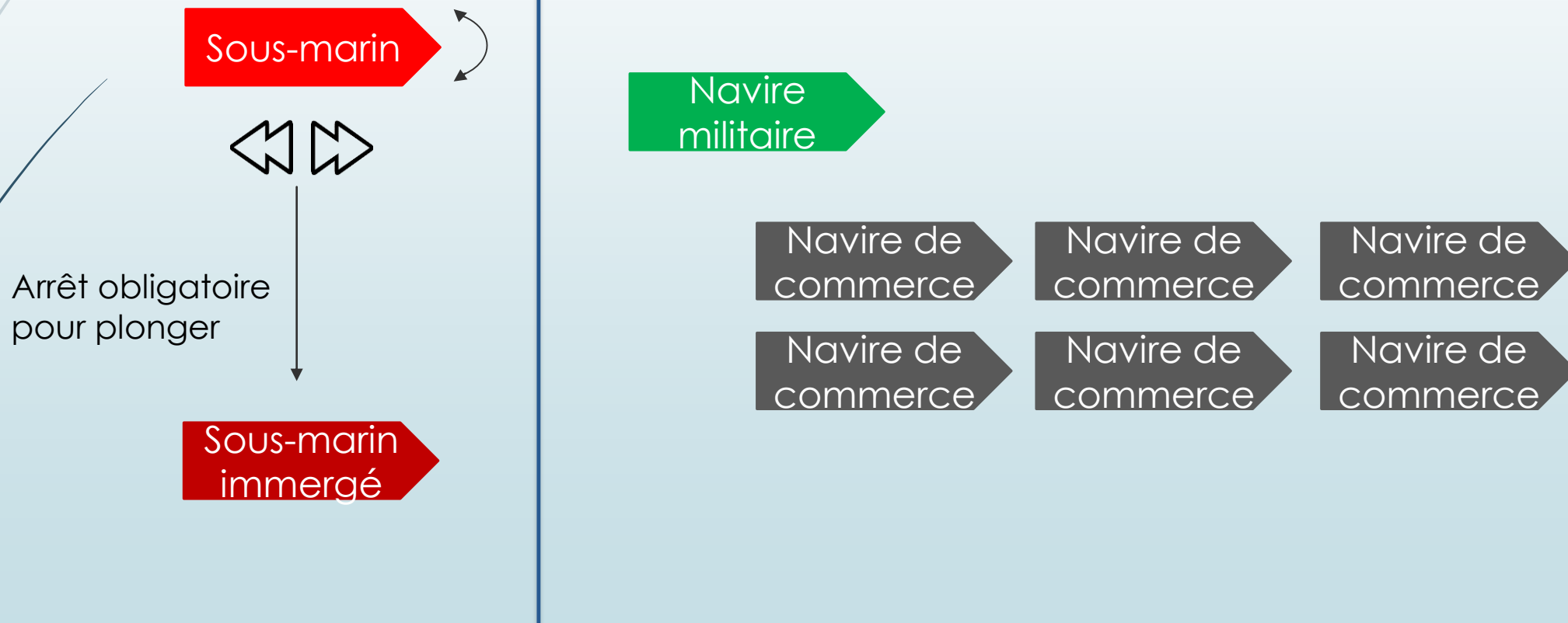
Problème majeur

Améliorations envisageables

Conclusion

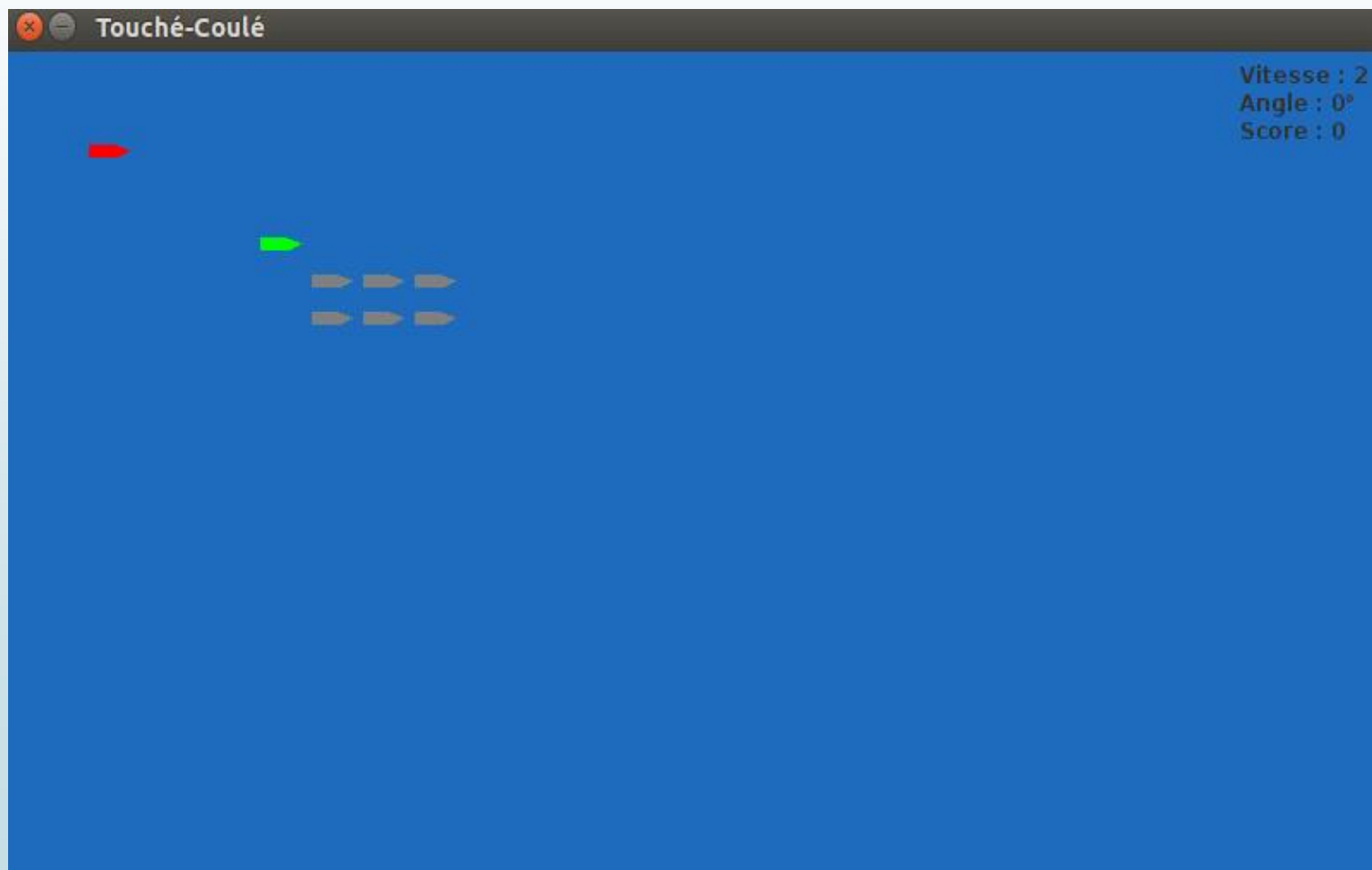
I. Présentation

1. Présentation général



I. Présentation

2. Déroulement



Sommaire

Introduction

I. Présentation

- 1. Présentation générale
- 2. Déroulement

II. Réalisation

- 1. Diagramme de classes
- 2. Classe Jeu
- 3. Classe Frame
- 4. Classe abstraite Bateau

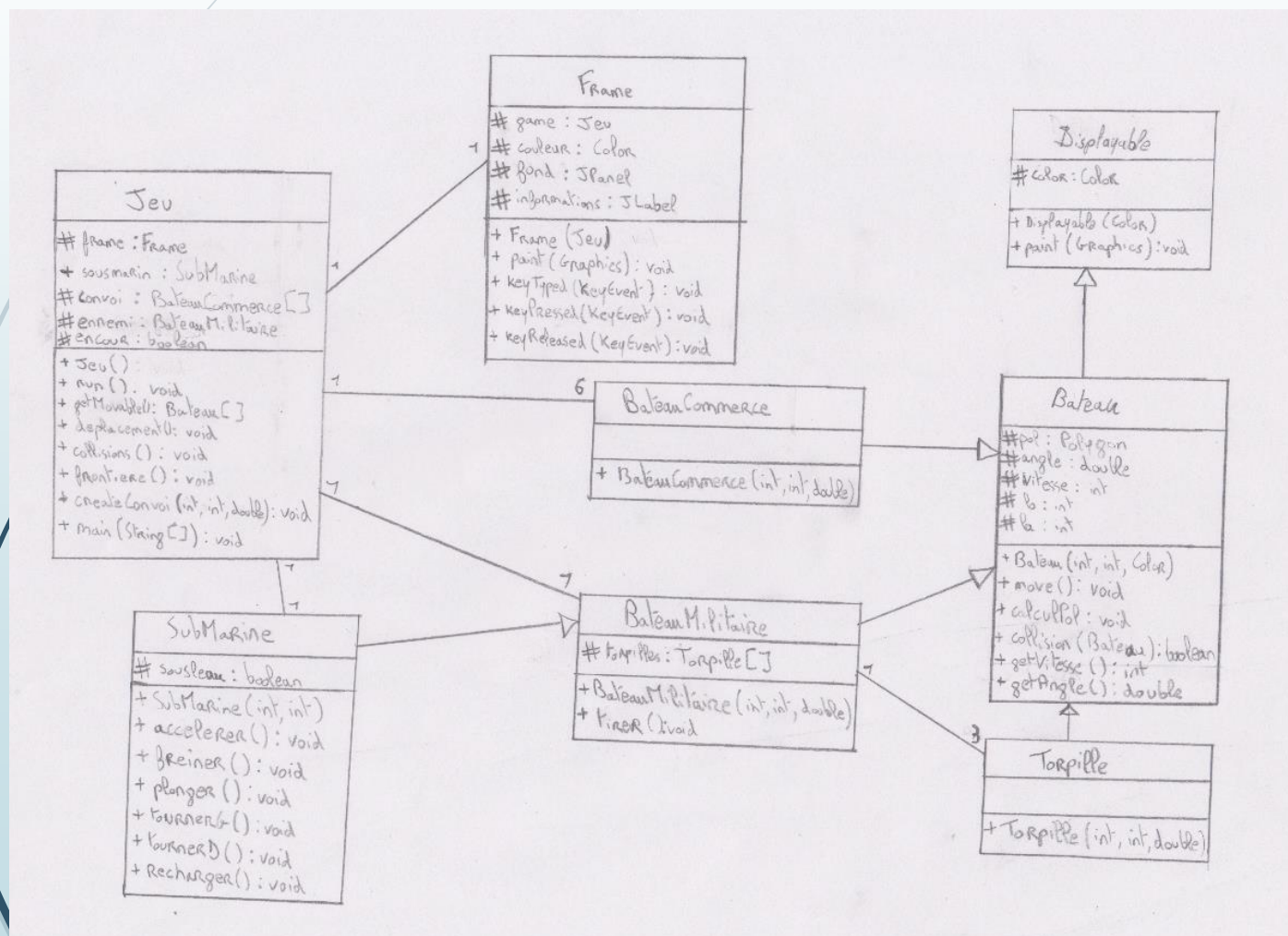
Problème majeur

Améliorations envisageables

Conclusion

II. Réalisation

1. Diagramme de classes



- 8 classes dont 2 abstraites
- 5 héritages
- Quelques compositions

II. Réalisation

2. Classe Jeu

```
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;
}
```

Méthode getMovable :

- Renvoi tout les objets qui peuvent se déplacer
- But : Pouvoir déplacer tout les objets
- Principe : Initialiser un tableau puis le remplir en fonction des objets existants

II. Réalisation

3. Classe Frame

```
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++;
    }
}
```

Méthode paint :

- But : Pouvoir afficher tout les objets sur la fenêtre
- Principe : Modifie les informations du sous-marin puis affiche le fond et ensuite affiche tout se qui existe

II. Réalisation

3. Classe Frame

```
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();  
    }  
}
```

Méthode keyPressed :

- But : Détecter les évènements du clavier
- Principe : Dès que le joueur appuie sur une touche, on lance certaine action

II. Réalisation

4. Classe abstraite Bateau

```
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;  
}
```

Méthode collision :

- Renvoi vrai en cas de collision
- But : Détecter une collision entre deux bateau
- Principe : Le premier bateau vérifie si dans sa forme, il contient l'un des points du second

Sommaire

Introduction

I. Présentation

- 1. Présentation générale
- 2. Déroulement

II. Réalisation

- 1. Diagramme de classes
- 2. Classe Jeu
- 3. Classe Frame
- 4. Classe abstraite Bateau

Problème majeur

Améliorations envisageables

Conclusion

Problème majeur



Détecter une collision :

- Fonctionne quand il n'y a aucun mouvement du convoi et selon la position de départ
- Non résolu

Sommaire

Introduction

I. Présentation

- 1. Présentation générale
- 2. Déroulement

II. Réalisation

- 1. Diagramme de classes
- 2. Classe Jeu
- 3. Classe Frame
- 4. Classe abstraite Bateau

Problème majeur

Améliorations envisageables

Conclusion

Améliorations envisageables



- Une classes modélisant les convois
- Des points d'apparitions
- Une barre de menu
- Enregistrer les meilleurs scores

Conclusion

Ce qu'il faut retenir :

- Objectifs atteints
- Beaucoup d'évolution possible