

PROJET : LES PIRATES

BEN JAAFAR CHEDLI / RIBEIRO DUARTE
BAGDATLI OKTAY / MUNOS ENZO
GUICHARD LUCAS / ESSENGUE MATIS

Vendredi 17 mai 2024

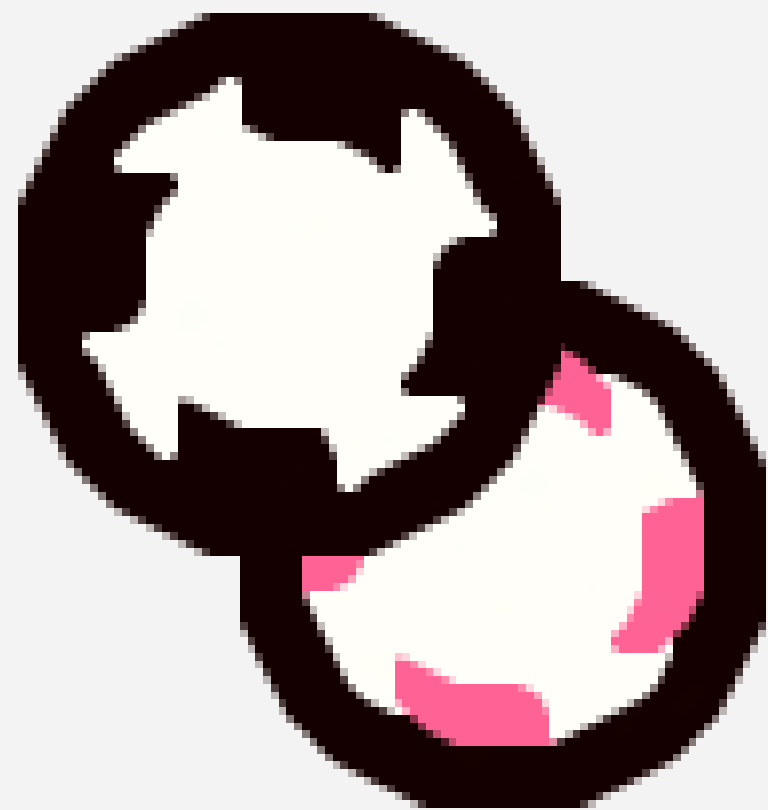


Université Paul Sabatier - 2024

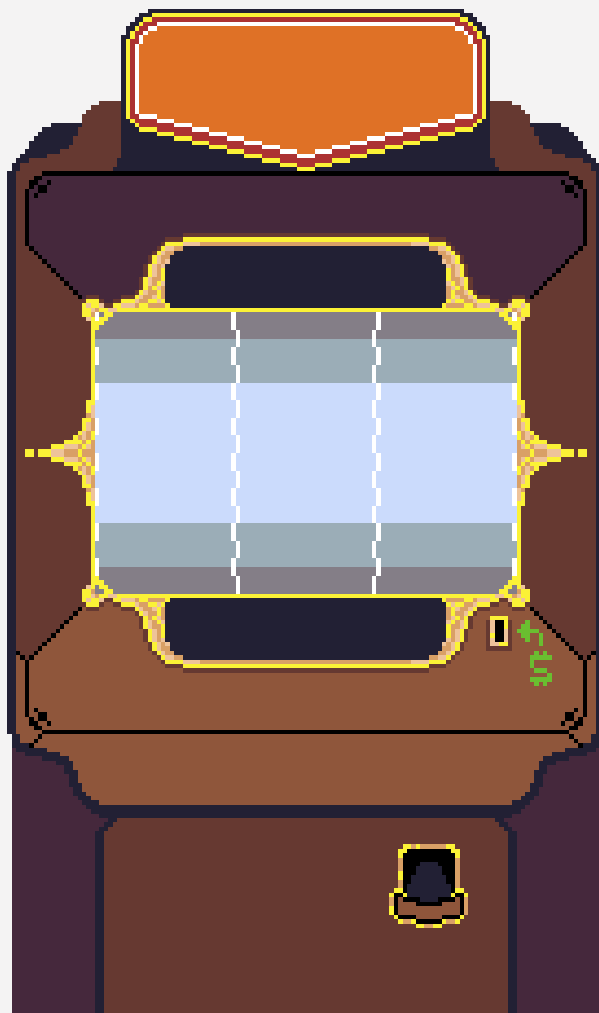
MAIS QUE FONT LES PIRATES AVEC LEUR ARGENT ?



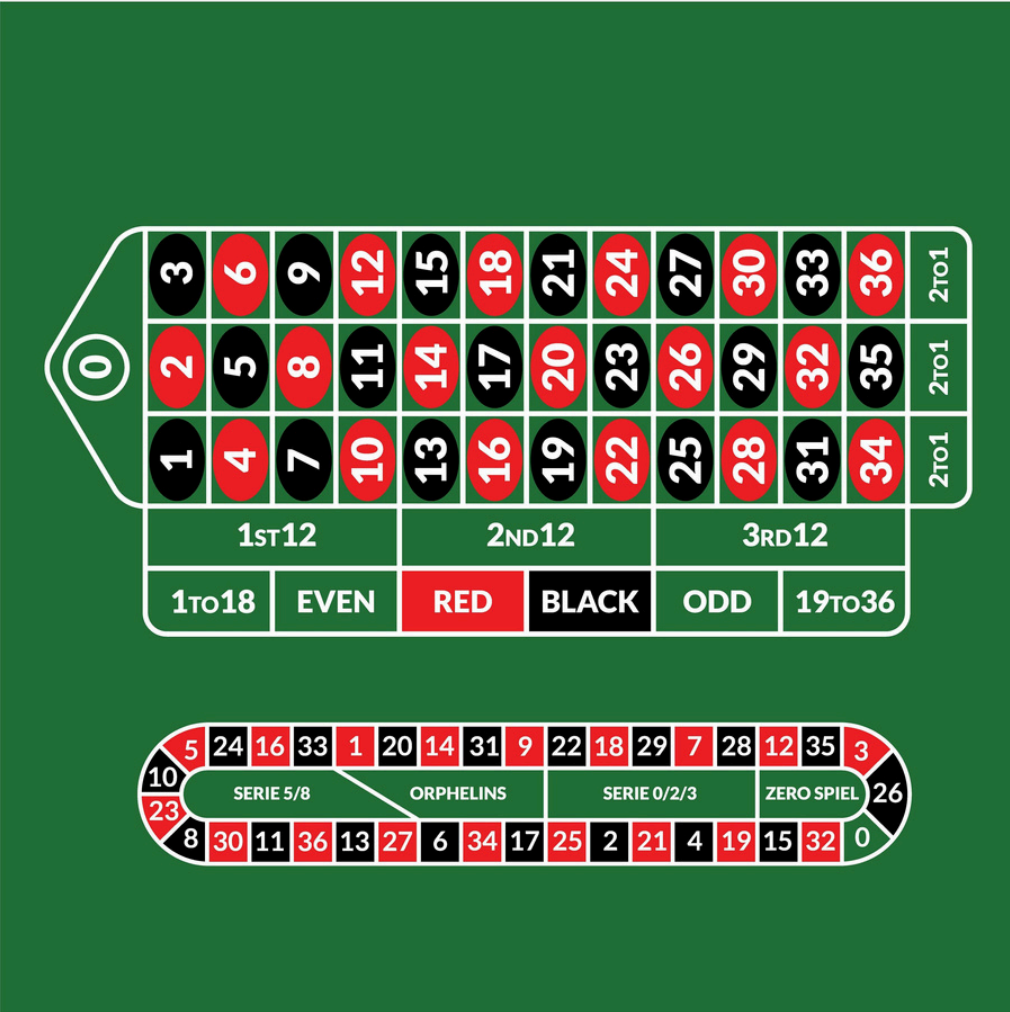
LE JEU



POINT DE VIE



VALEURS DE DEPLACEMENT

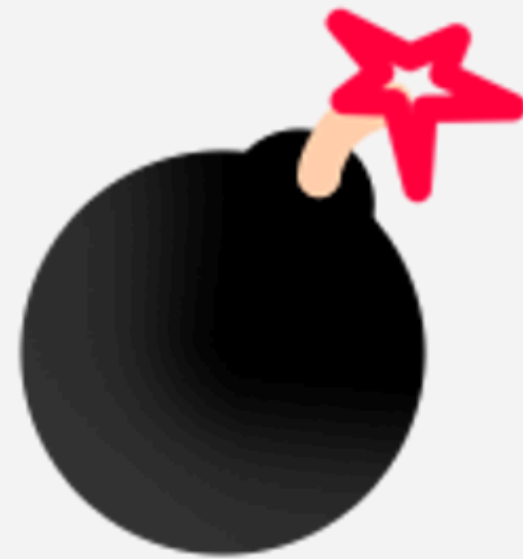


PLATEAU

LES CASES



REJOUER



DÊGATS

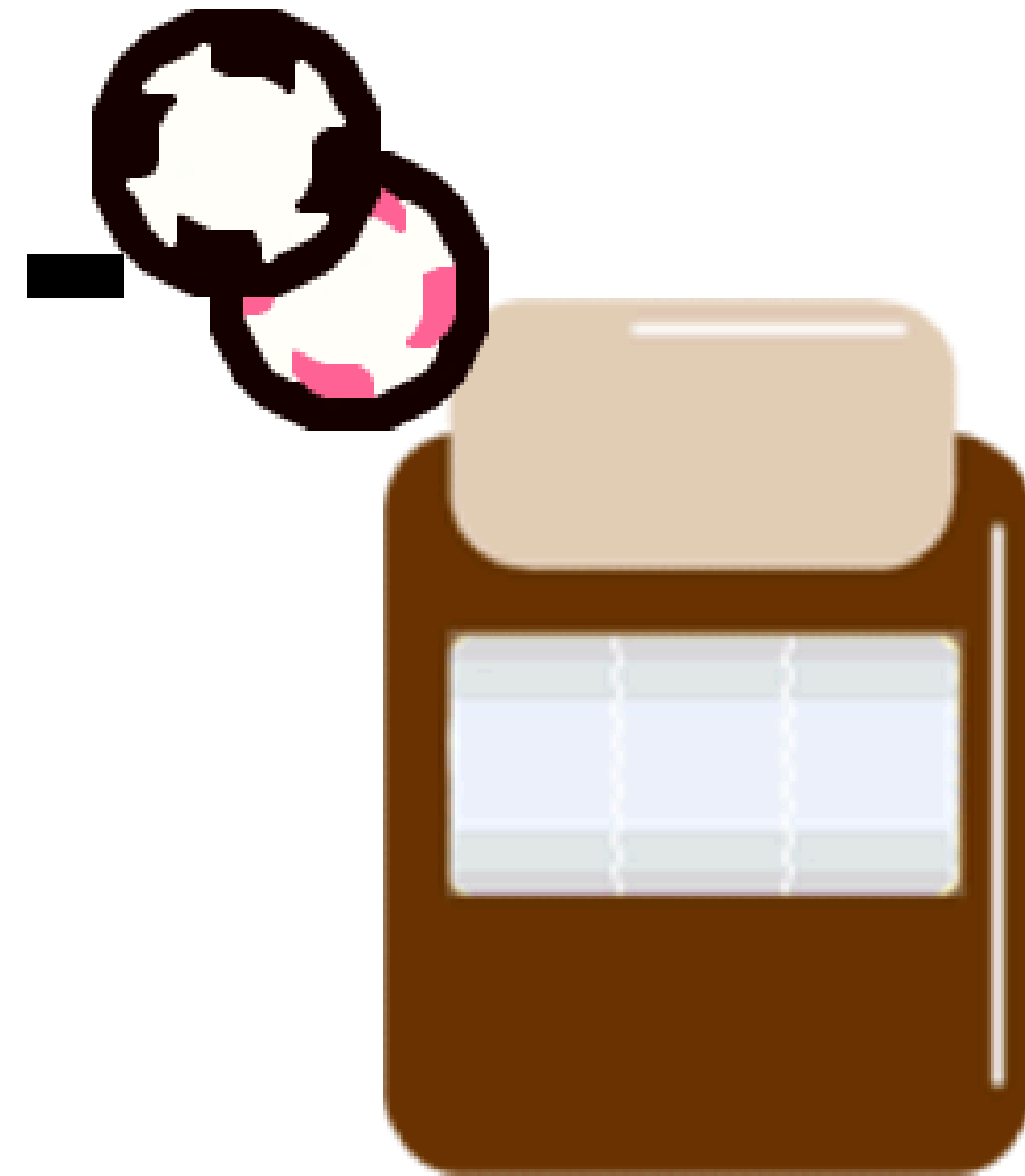


RECULER



DUEL ROULETTE

UNE AUTRE MÉCANIQUE



ORGANISATION

En 2 étapes

GROUPEMENT

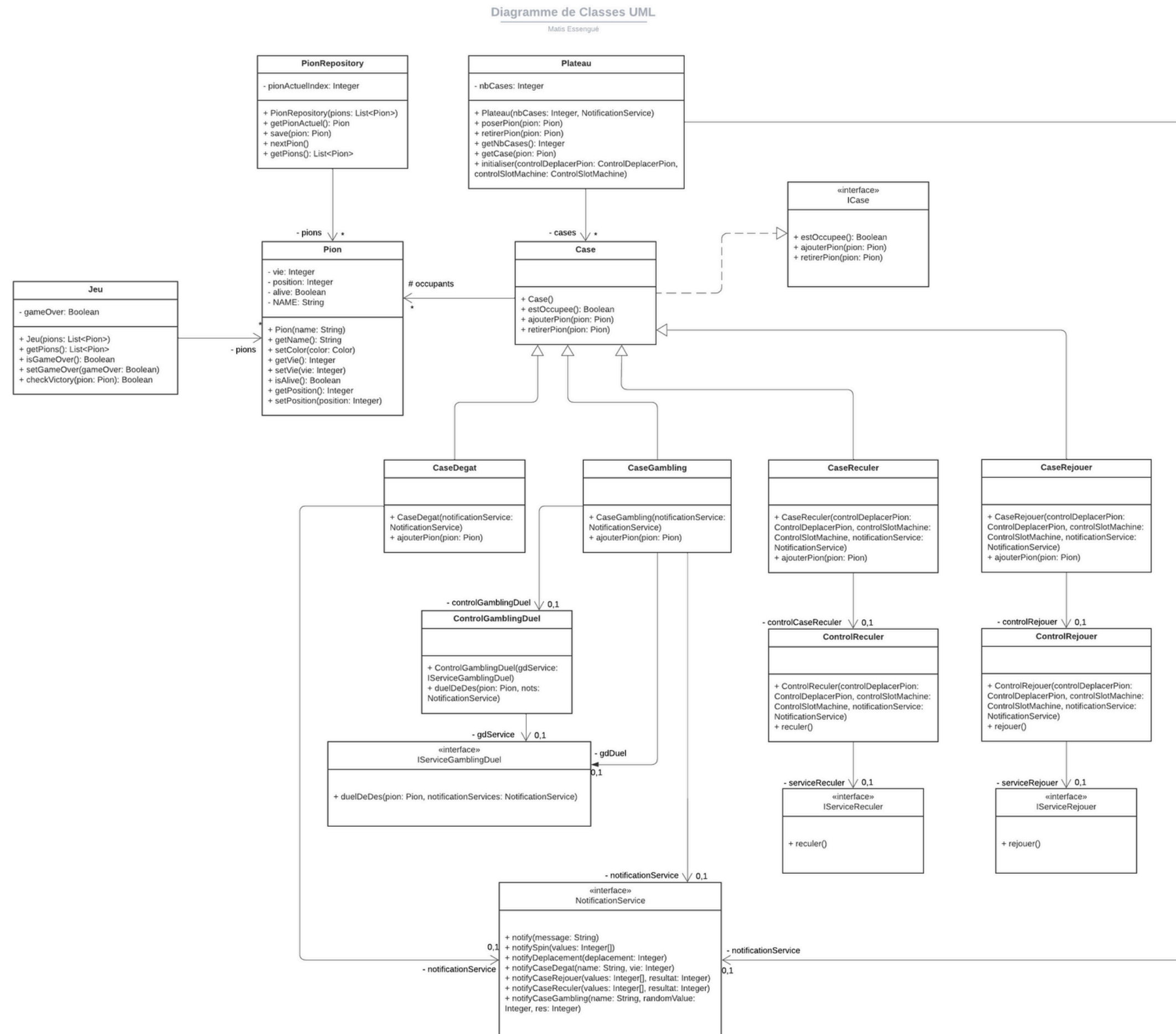
Refactorisation de l'architecture
ECB
Pause du developpment de la
partie IHM

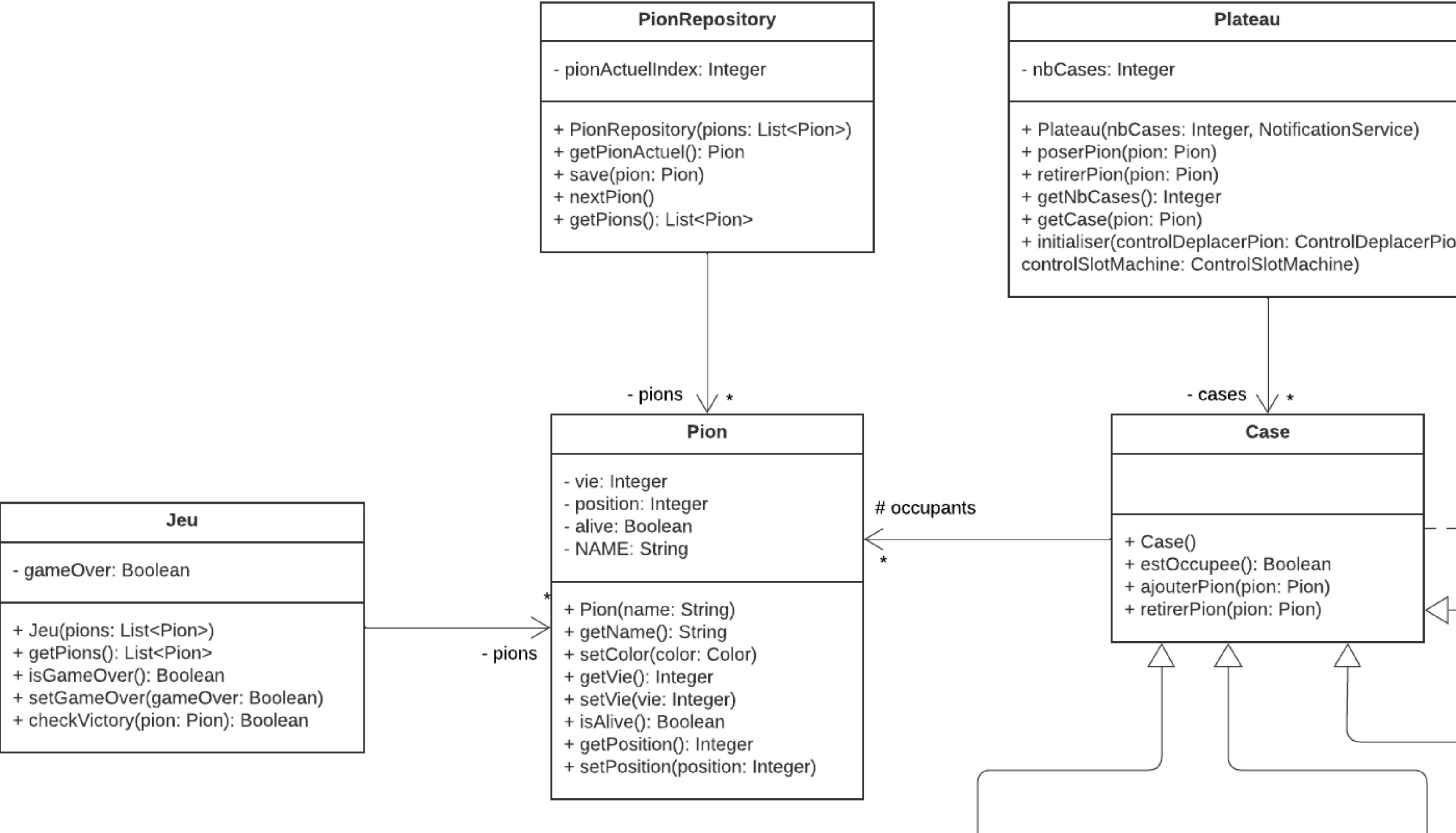
01

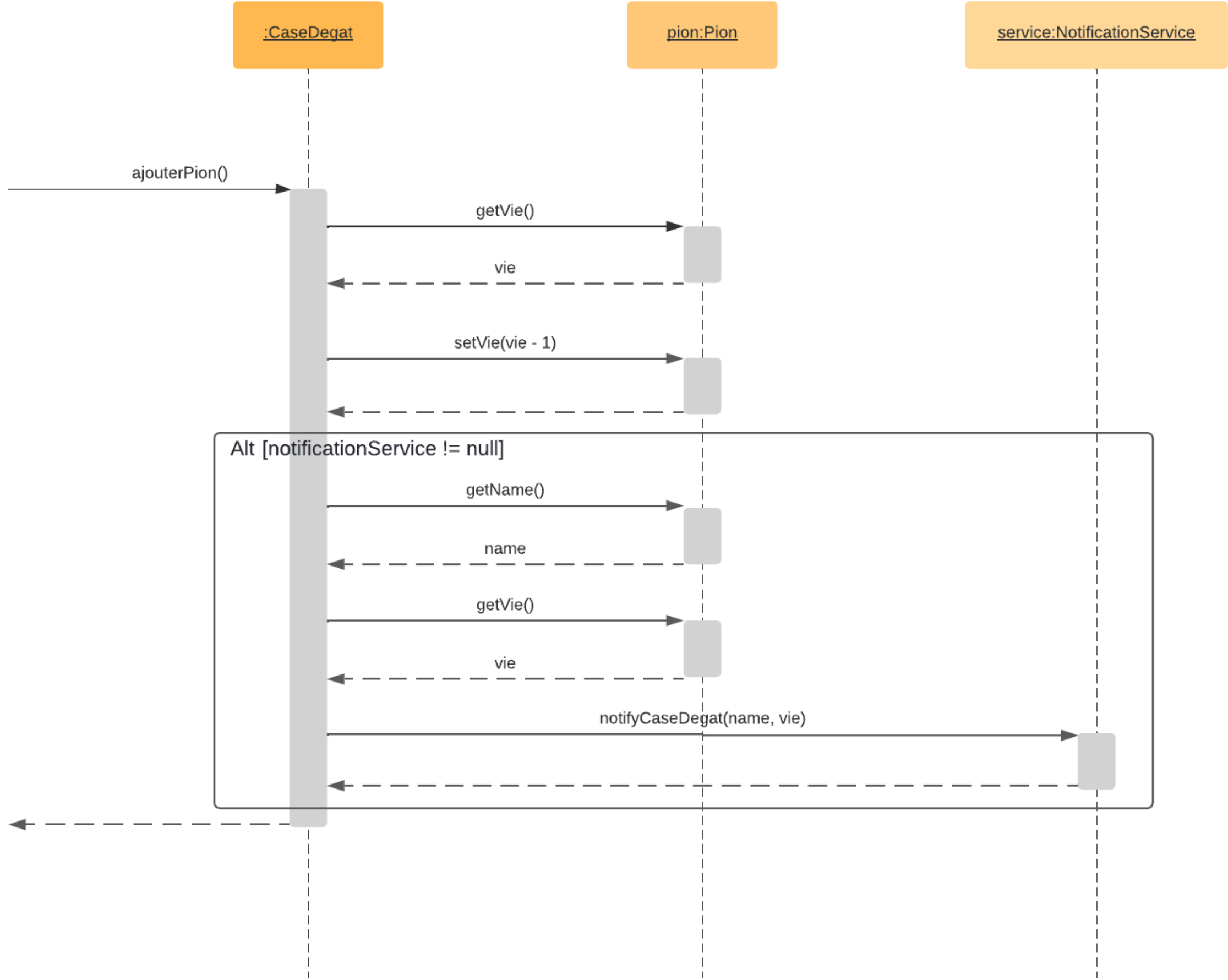
RÉPARTITION

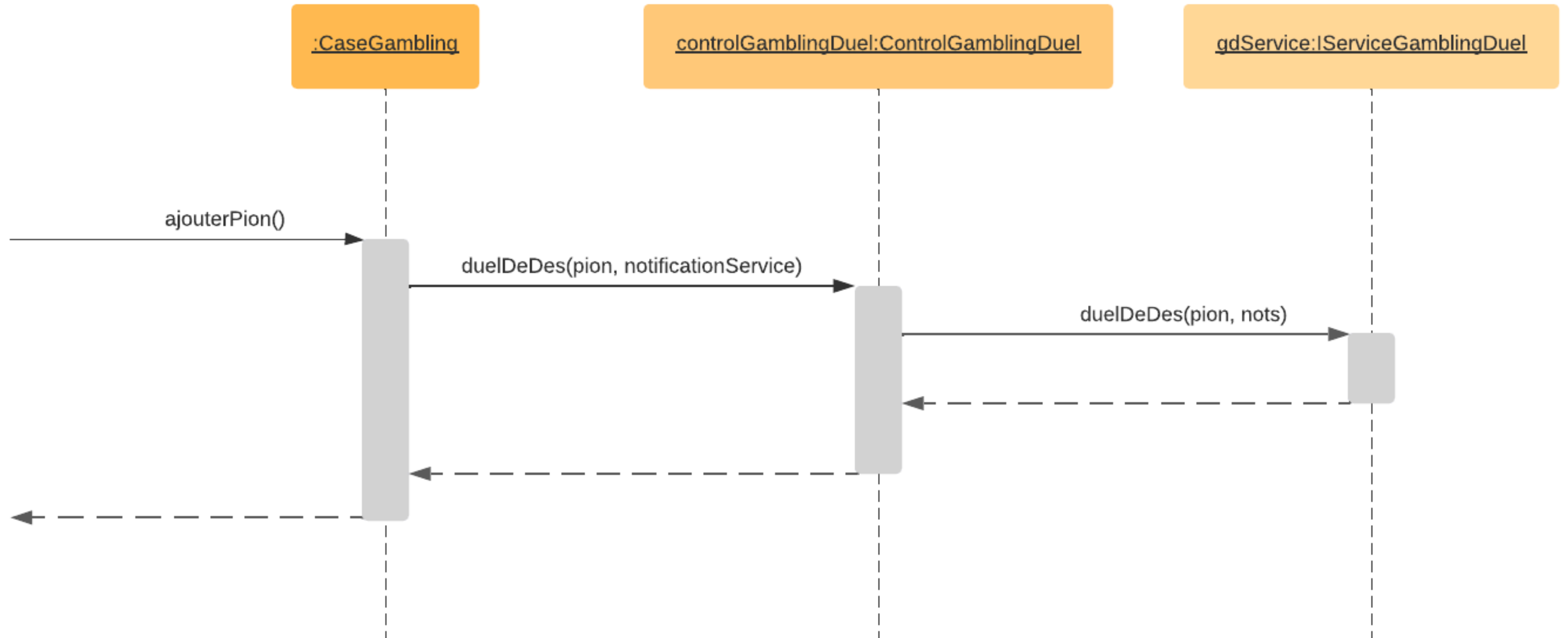
Les tests fonctionnelles
Le diagramme des classes et
l'UML
Mise à jour du code

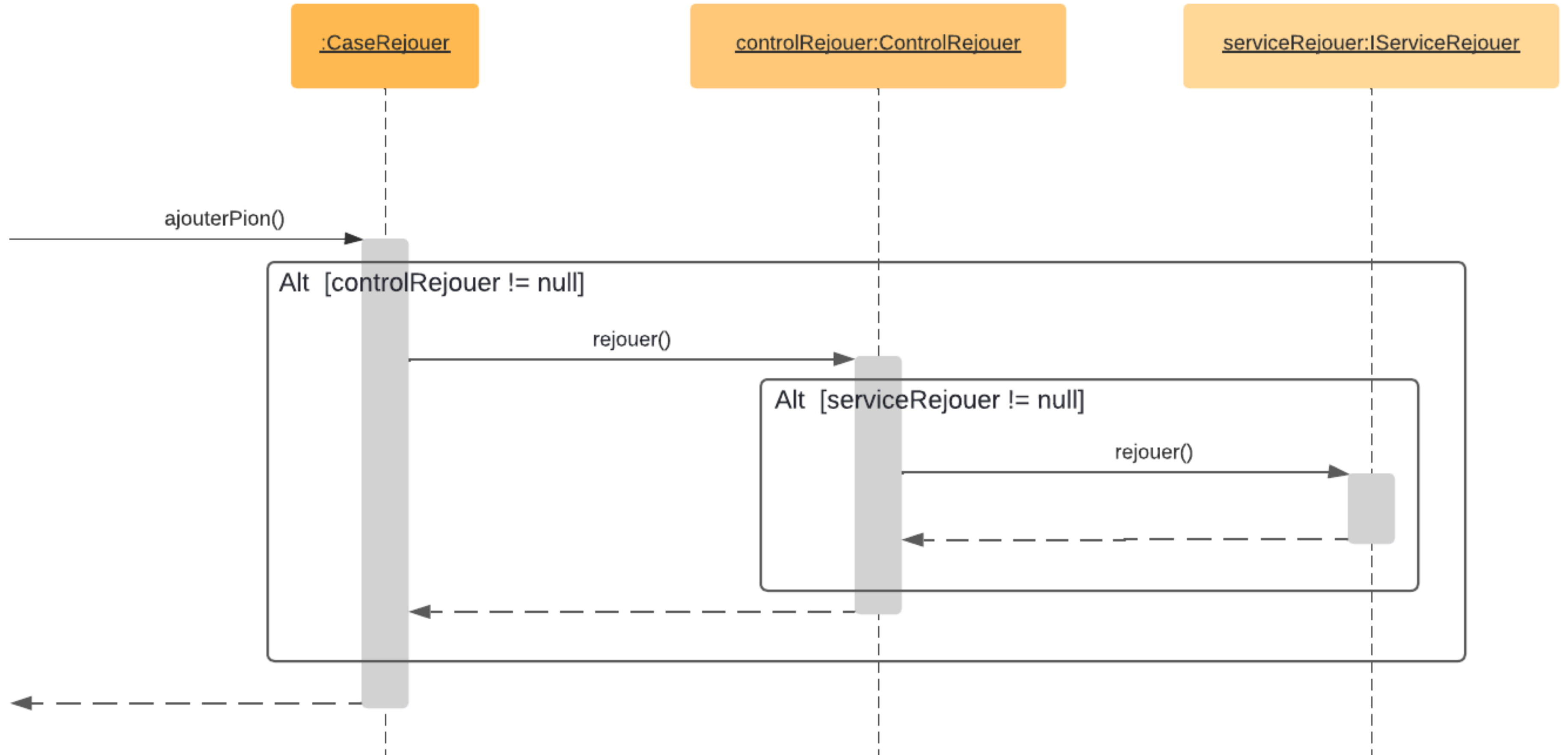
02

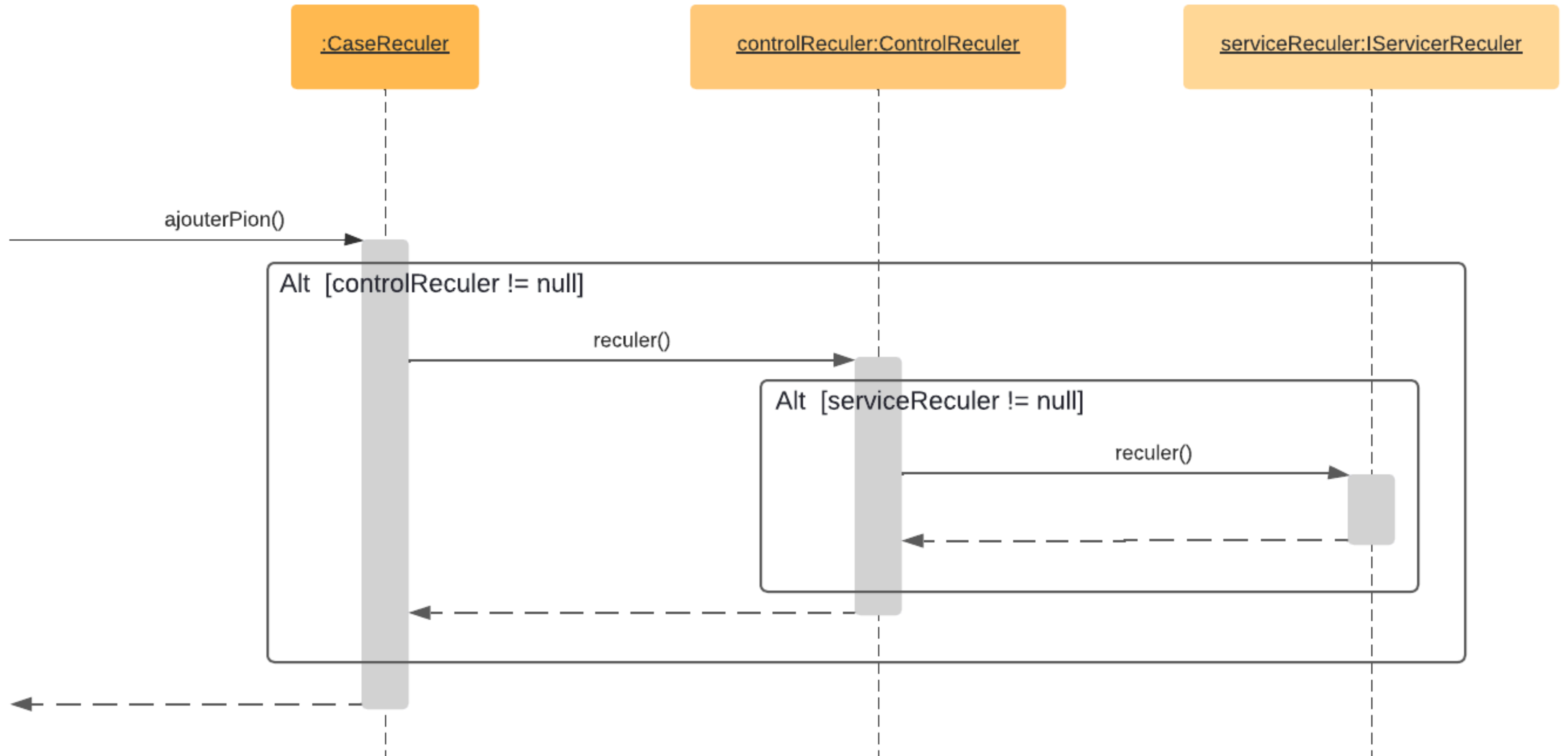




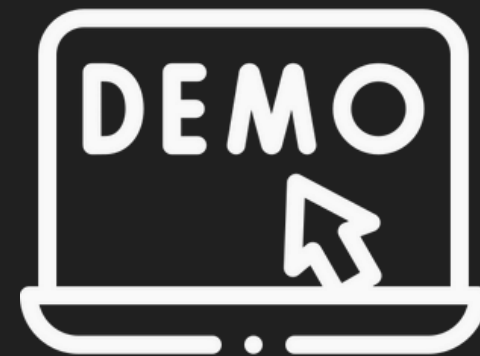


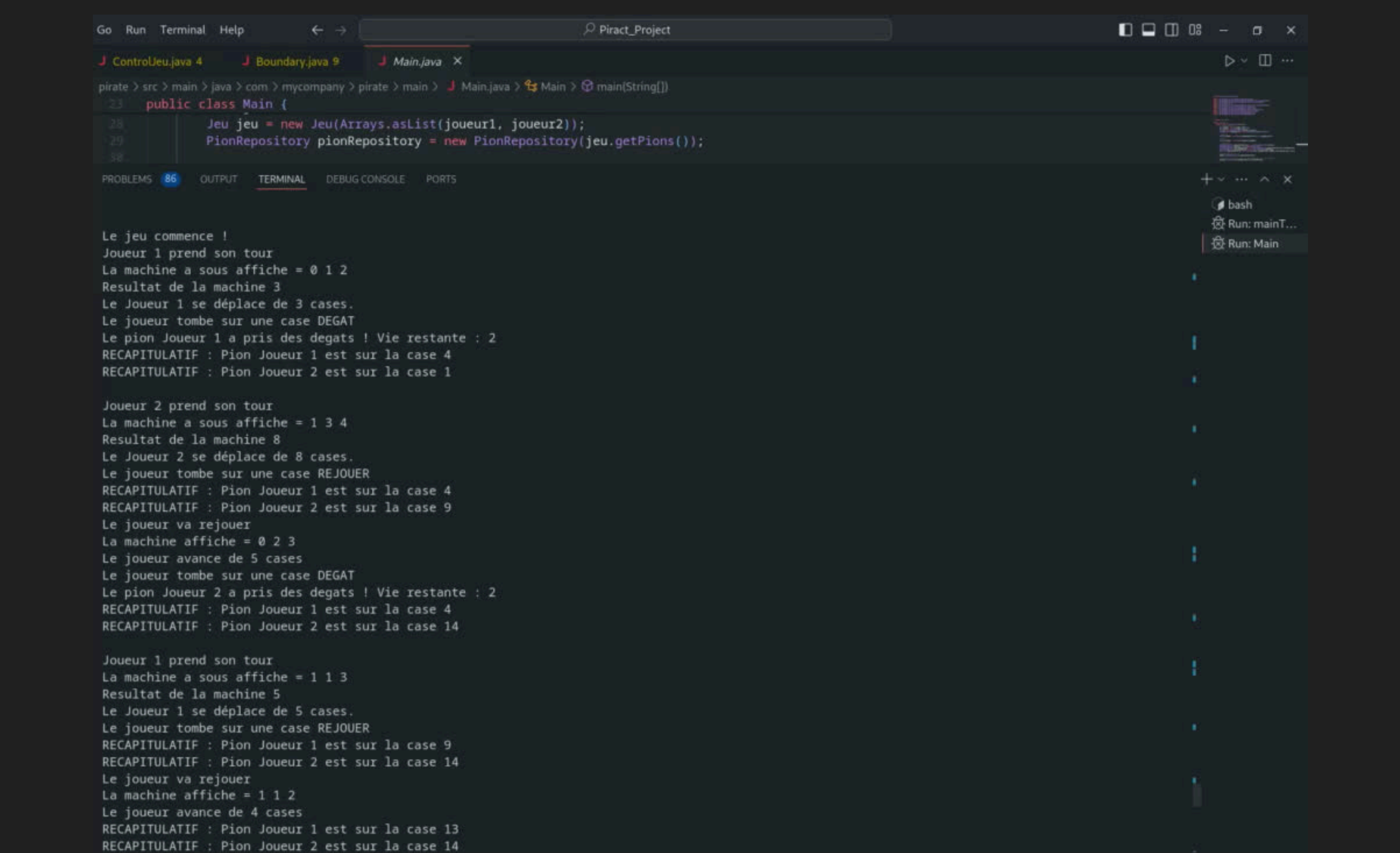


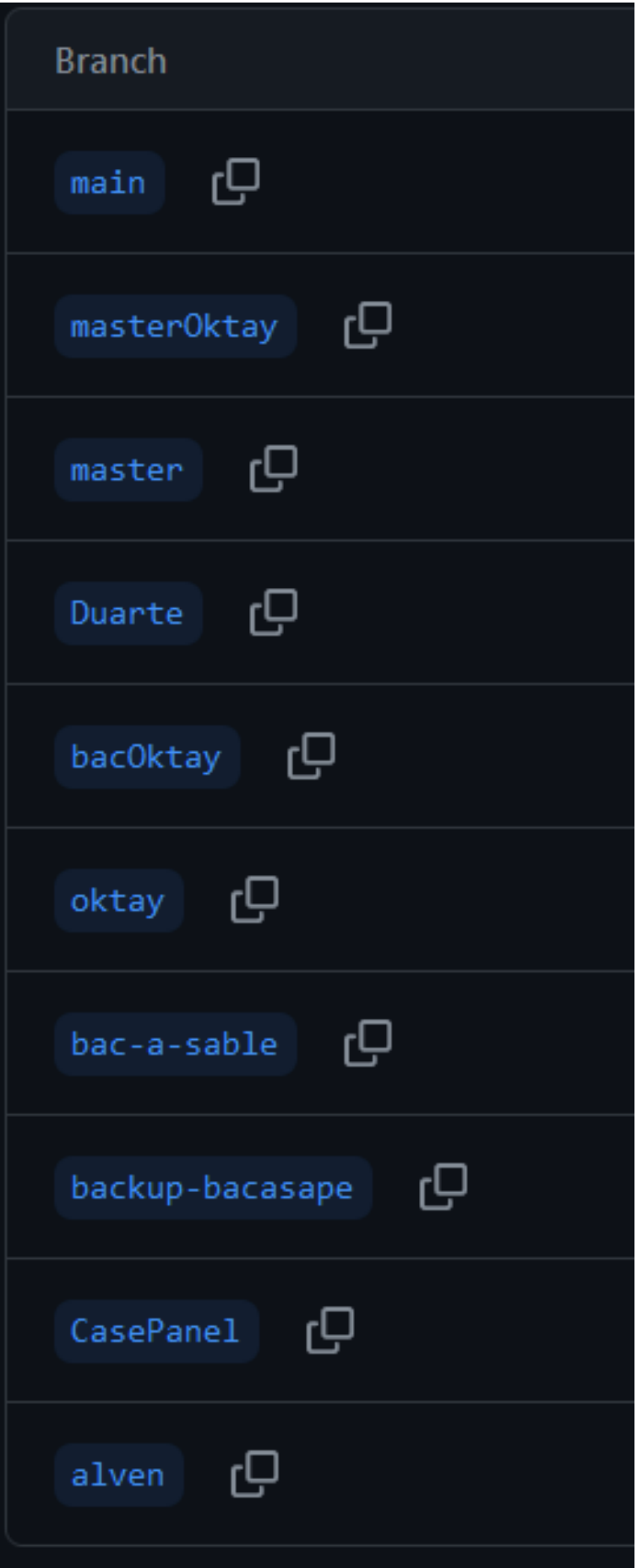
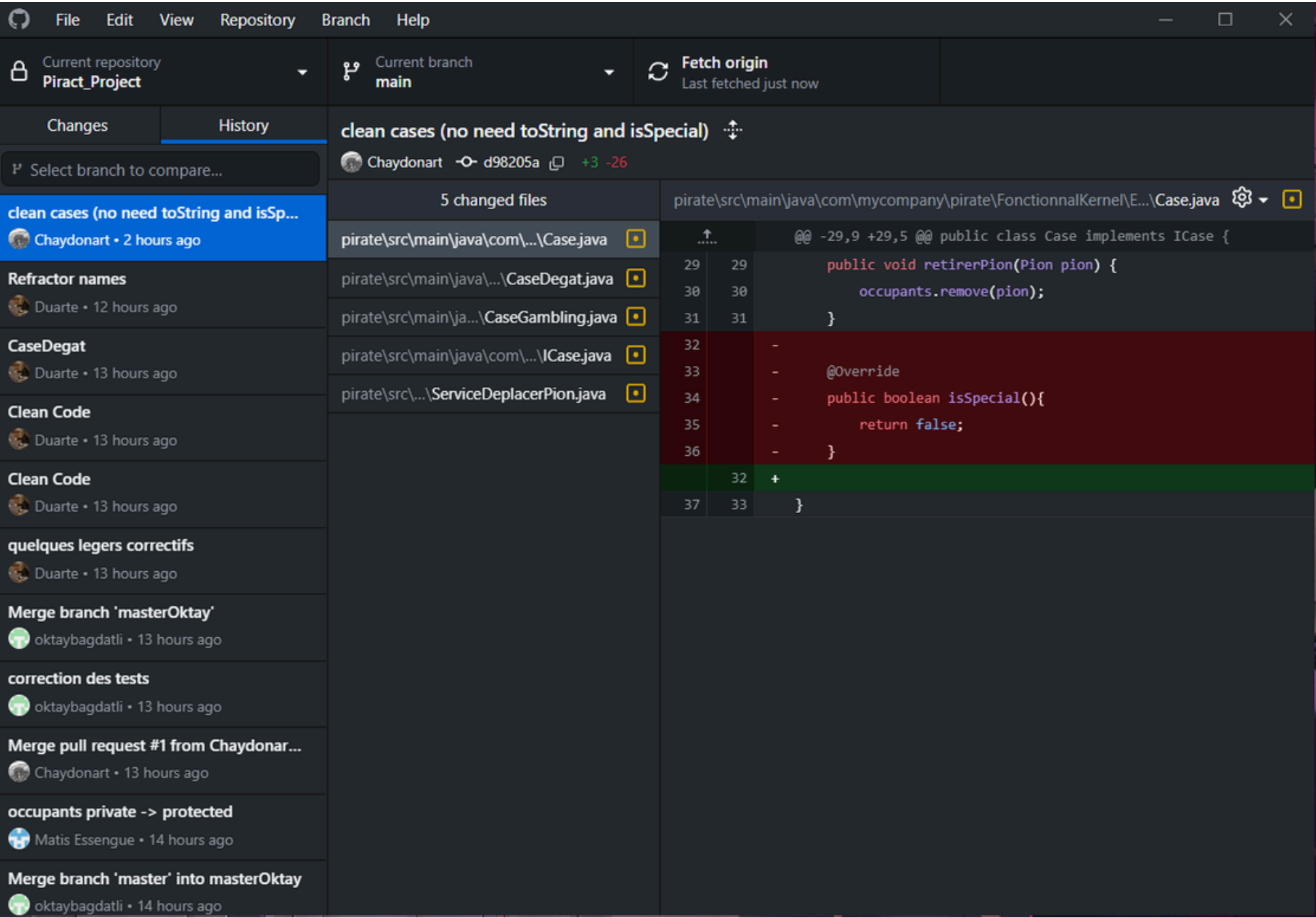


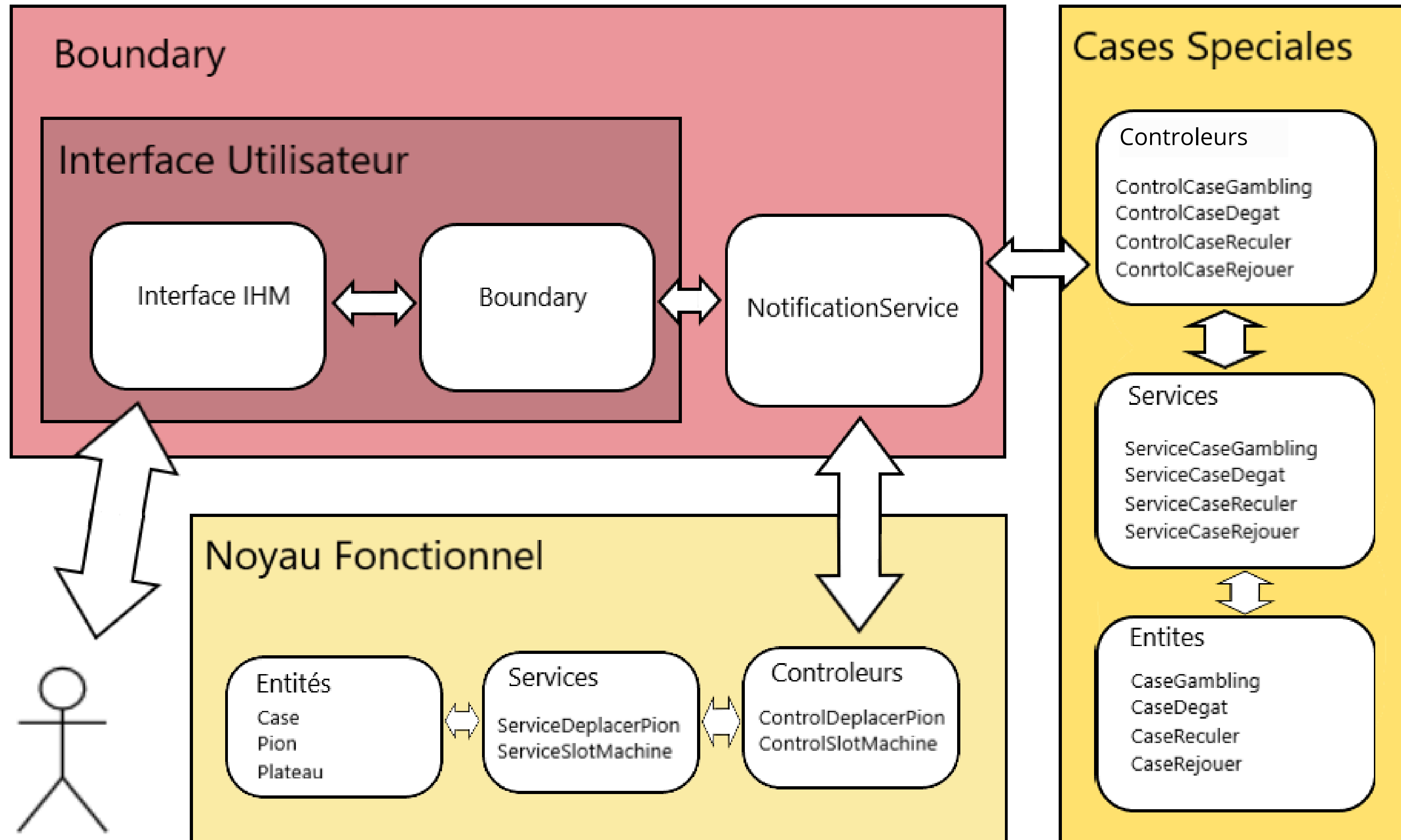



NOUS ALLONS MAINTENANT PROCÉDER A UNE DÉMO




















 com.mycompany.pirate.Boundary.UserInterface

-  AnimatedGIF.java
-  BackgroundPanel.java
-  Board.java
-  CartePanel.java
-  CasePanel.java
-  ImagePanelSlotmachine.java
-  LifePanel.java
-  PionPanel.java
-  Pirate.java
-  SlotMachinePanel.java
-  newTest.java

```

@Override
public void notifyDeplacement(int deplacement) {
    deplacerPion(deplacement);
}

@Override
public void notifyCaseDegat(String name, int vie) {
    afficherMessage("Le joueur tombe sur une case DEGATS");
    afficherMessage("Le pion " + name + " a pris des degats ! Vie restante : " + vie);
}

@Override
public void notifyCaseRejouer(int[] values, int resultat) {
    afficherMessage("Le joueur tombe sur une case REJOUER");
    notifyDeplacement(resultat);
    afficherMessage("Le joueur va rejouer");
    afficherMessage("La machine affiche = " + values[0] + " " + values[1] + " " + values[2]);
    afficherMessage("Le joueur avance de " + resultat + " cases");
}

@Override
public void notifyCaseReculer(int[] values, int resultat) {
    afficherMessage("Le joueur tombe sur une case RECULER");
    notifyDeplacement(resultat);
    afficherMessage("La machine affiche = " + values[0] + " " + values[1] + " " + values[2]);
    afficherMessage("Le joueur recule de " + (-resultat) + " cases");
}

@Override
public void notifyCaseGambling(String name, int randomValue, int res) {
    afficherMessage("Le " + name + " tombe sur une case GAMBLING");
    afficherMessage("Le " + name + " va donc proceder a un duel contre Gambi le robot !");
    afficherMessage("Duel gambling ! Le joueur doit faire une valeur superieure a "+ randomValue);
    afficherMessage("La roulette affiche... " + res + " !");
}

```

La classe Boundary :

Start ()

Spin()

deplacerPion()

etatJeu()

INotificationService

- Preconditon
- Cas à tester
- Exigences
- Postcondition
- Adapter/Changer
- Recommencer

```

package com.mycompany.pirate.test;

import java.util.HashMap;
import java.util.Map;
import java.util.function.Supplier;

public abstract class Tester {
    private Map<String, Supplier<Boolean>> funcRes = new HashMap<>();

    public Tester() {
        TestRes.gameUI.setGameLoopController(TestRes.gameLoopController);
        TestRes.plateau.initialiser(TestRes.controlDeplacerPion, TestRes.controlSlotMachine);
    }

    public Map<String, Supplier<Boolean>> getFuncRes() {
        return funcRes;
    }

    public void setFuncRes(Map<String, Supplier<Boolean>> funcRes) {
        this.funcRes = funcRes;
    }

    public void test(){
        System.out.println("execution des tests pour " + this.getClass().getSimpleName() + ":");
        System.out.println();
        this.funcRes.entrySet().stream().forEach( e -> System.out.println(
            e.getValue().get() ? e.getKey() + " passed\n" : e.getKey() + " failed\n"
        ));
        System.out.println(x:"-----");
    }
}

package com.mycompany.pirate.test;

import java.util.ArrayList;
import java.util.List;

public class TestAll {
    private List<Tester> tests = new ArrayList<>();

    public TestAll() {
        this.tests.add(new TestDeplacerPion());
        this.tests.add(new TestGamblingDuel());
        this.tests.add(new TestRejouer());
        this.tests.add(new TestSlotMachine());
        this.tests.add(new TestReculer());

        this.tests.stream().forEach( t -> t.test());
    }

    public static void main(String[] args) {
        TestAll testAll = new TestAll();
    }
}

public class TestRes {
    public static Pion joueur1 = new Pion(name:"Joueur 1");
    public static Pion joueur2 = new Pion(name:"Joueur 2");
    public static Jeu jeu = new Jeu(Arrays.asList(joueur1, joueur2));
    public static PionRepository pionRepository = new PionRepository(jeu.getPions());
    public static Boundary gameUI = new Boundary(gameLoopController:null, pionRepository);
    public static Plateau plateau = new Plateau(nbCases:36,gameUI);
    public static ServiceDeplacerPion deplacerPionService = new ServiceDeplacerPion(plateau);
    public static ServiceSlotMachine smService = new ServiceSlotMachine();
    public static ControlDeplacerPion controlDeplacerPion = new ControlDeplacerPion(new ServiceDeplacerPi
    public static ControlSlotMachine controlSlotMachine = new ControlSlotMachine(smService);
    public static ControlJeu gameLoopController = new ControlJeu(jeu, pionRepository, notificationService
}
    
```

```
execution des tests pour TestDeplacerPion:

position initial -> 1, deplacement -> 36, destination finale ->36
testDeplacementSurpasse passed
position initial -> 1, deplacement -> 5, destination finale ->6
testDeplacementValid passed
-----

execution des tests pour TestGamblingDuel:

Duel gambling ! Le joueur doit faire une valeur superieure a 7
La roulette affiche... 5 !
Joueur 1 a perdu le gambling ! Vie restante : 2
TestDuelDeDes passed
-----

execution des tests pour TestRejouer:

Le joueur tombe sur une case REJOUER
Le joueur va rejouer
La machine affiche = 0 3 4
Le joueur avance de 7 cases
1
TestRejouerPion passed
-----

execution des tests pour TestSlotMachine:

interval -> [0,3[| valeur -> 1
interval -> [1,4]| valeur -> 4
interval -> [1,4]| valeur -> 4
TestDansInterval passed
value 0 => 0
value 1 => 1
value 2 => 3
TestFormatCorrect passed
-----

execution des tests pour TestReculer:

Le joueur tombe sur une case RECULER
La machine affiche = 1 1 4
Le joueur recule de 6 cases
position initial etait -> 13,position final -> 7
TestPosition passed
-----
```

Test Déplacement

Précondition	Initialisation de pion et contrôleur	
Déplacement valide	Déplacer le pion	Pos : 1, Dep : 2 -> PosFin : 3
Déplacement qui surpasse la taille de plateau	Déplacer le pion à la limite de taille de plateau	Pos : 25, Dep : 10 -> PosFin :36
Postcondition	Pion avec une position avancé et dans la limite de plateau	

Test Gambling

Précondition	Initialisation de pion et contrôleur	
Duel effectué	Effectuer le duel contre la machine et voir si le pion a perdu son point de vie ou gagné le duel	DeMachine : 4 , DePion : 7 -> PointVie = cst DeMachine :5, DePion : 2 -> PointVie -=1
Postcondition	Le pion qui a perdu ou pas perdu son point de vie	

Test Reculer

Précondition	Initialisation de pion et contrôleur	
Effectuer recule	Change la position du pion	Pos : 5 , Dep : 3 -> Pos Fin : 2
Postcondition	Le pion avec la position reculé	

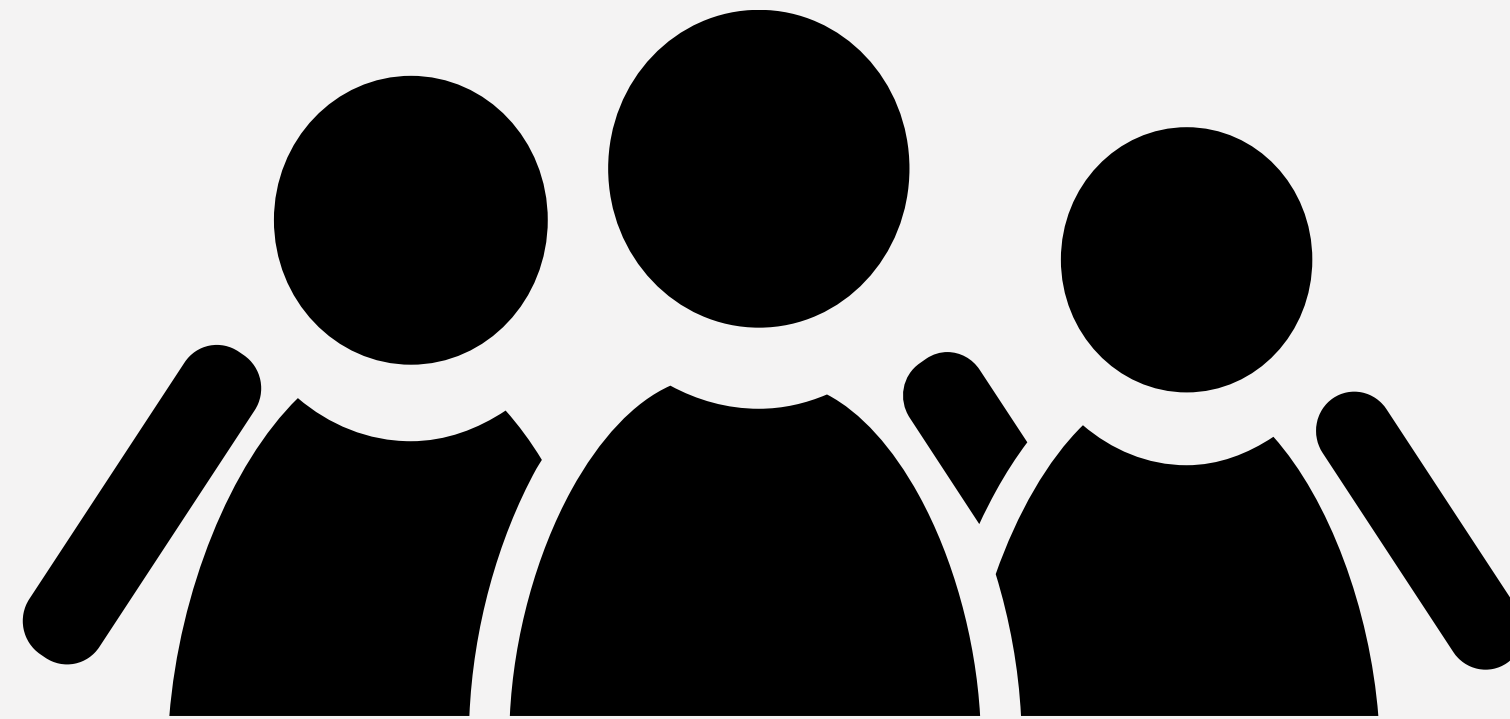
Test Rejouer

Précondition	Initialisation des attribut	
Rejoue	Slot machine doit encore faire un spin pour décider le déplacement	Apres caseRejouer compteur+=1(peut être plusieurs fois)
Postcondition	Compteur de spin incrémenté	

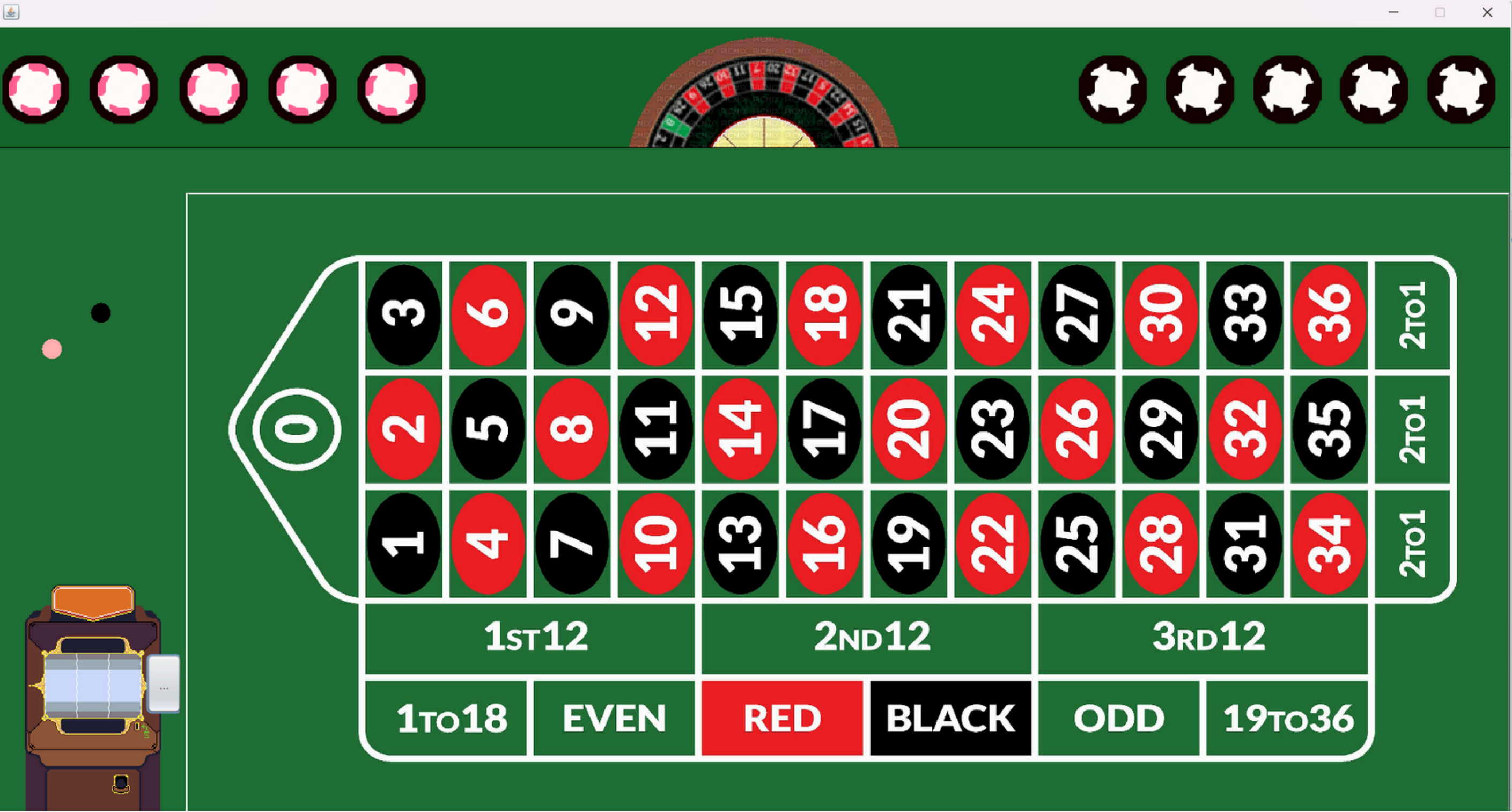
Test Slot Machine

Précondition	Initialisation de Slot Machine	
Dans bon format	Faut que fonction spin retourne 3 Integer	Values.length == 3 -> True
Dans Intervalle	Faut que les valeurs retournées par le spin aient les valeurs propres	1 -> [0,3[; 2,3 ->[1,4] ⇒ True
Post Condition		

RETOUR SUR L'EXPÉRIENCE DU TRAVAIL D'ÉQUIPE ET LE PROJET



Mockup d'écran



Merci.

PROJET PIRATES - 2024 - UNIVERSITE PAUL SABATIER



BEN JAAFAR CHEDLI / RIBEIRO DUARTE / BAGDATLI OKTAY / MUNOS ENZO / GUICHARD LUCAS / ESSENGUE MATIS