El Presidente

Technical documentation

Team 8
Nori Tran Nguyen
Nacim Messaci
Kevin Cheng

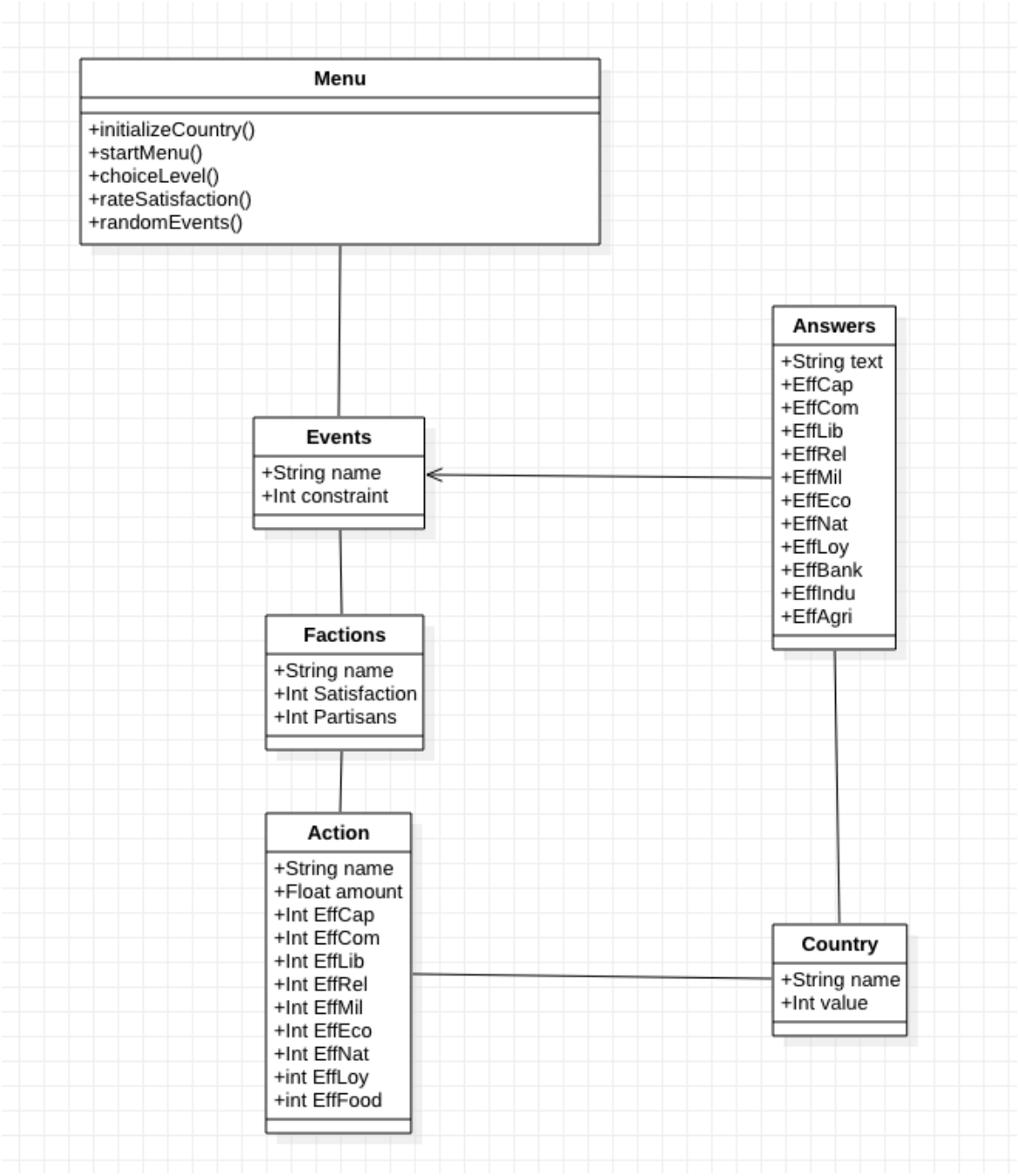
I/ Introduction

The game El Presidente consists of managing an island where 8 competing factions are present. Each faction is defined by a name (eg: capitalists, communists, ...) and has a number of supporters for each. Various events will happen through the game.
Depending on the event, the player may have the possibility to chose an action (between 1 to 3) and react to the event.

Depending on the event and/or the choices, the number of supporters and their satisfaction will be modified. The President's territory will be divided into two parts: Agriculture and Industry. Agriculture will be used to provide food for the government while industry will provide money. The object of this game will be to have enough money to face the various events while having enough food to provide needs for the people.

In fact, the overall satisfaction of the entire population must remain above a certain value, depending on the difficulty chosen by the player.

II/ Presentation of design pattern.



III/ Presentation of object.

Our project El Presidente is compose by 4 objects and 2 enums.

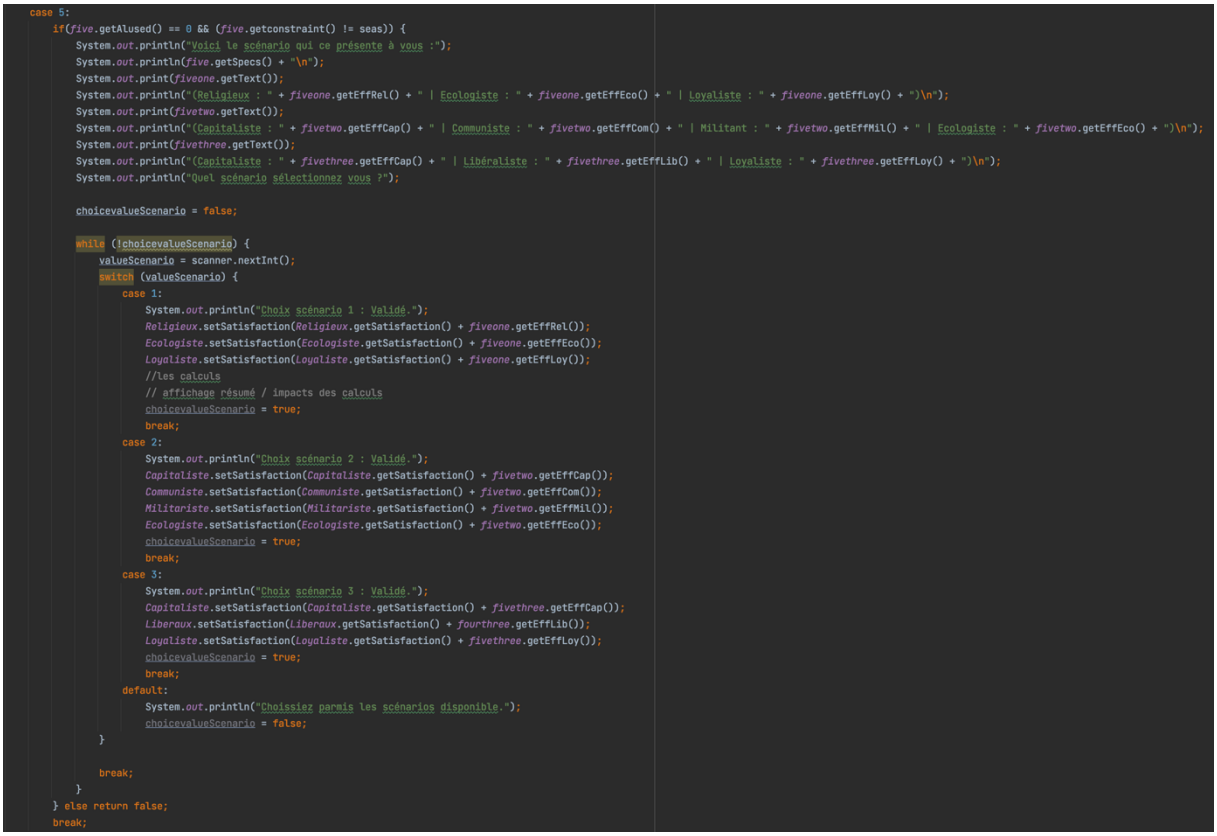
Main : object main is use to call all the functions.

```
package src;

import java.util.Scanner;

public class Main {

    public static void main(String[] args){
            Scanner scanner = new Scanner (System.in);
            int seasons =1, EndGame =0;
            Menu.startMenu();
            String president = Menu.initializePresident();
            String country = Menu.initializeCountry();
            float difficulty = Menu.choiceLevel();
            while(Menu.loosecondition(difficulty) && EndGame <20) {
                if(Menu.randomEvents(seasons)){
                    Menu.viewUpdate(president, country);
                    seasons = Menu.season(seasons, president, country);
                    Menu.repartition();
                    Menu.checkValueSat();
                    Menu.getdown();
                    EndGame++;
                }
            }
    }
}
```

Menu : this object menu manage all the rules of the game. Contains all procedures of the program like :

- menuStart(): which initialize the start menu of the game. In this step, we have the choice between check the rules of the game or go directly inside it.
- choiceLevel() : which give us the possibility to choose the difficulty of our game. (Easy / Medium or Hard)
- randomEvents() : this procedure generate randomly an event and propose the answers which are linked with.

Example of case 5 in randomEvent()



```
case 5:
    if(five.getAlused() == 0 && (five.getconstraint() != seas)) {
        System.out.println("Voici le scénario qui ce présente à vous :");
        System.out.println(five.getSpecs() + "\n");
        System.out.print(fiveone.getText());
        System.out.println("(Religieux : " + fiveone.getEffRel() + " | Ecologiste : " + fiveone.getEffEco() + " | Loyaliste : " + fiveone.getEffLoy() + ")\n");
        System.out.print(fivetwo.getText());
        System.out.println("(Capitaliste : " + fivetwo.getEffCap() + " | Communiste : " + fivetwo.getEffCom() + " | Militant : " + fivetwo.getEffMil() + " | Ecologiste : " + fivetwo.getEffEco() + ")\n");
        System.out.print(fivethree.getText());
        System.out.println("(Capitaliste : " + fivethree.getEffCap() + " | Libéraliste : " + fivethree.getEffLib() + " | Loyaliste : " + fivethree.getEffLoy() + ")\n");
        System.out.println("Quel scénario sélectionnez vous ?");

        choicevalueScenario = false;

        while (!choicevalueScenario) {
            valueScenario = scanner.nextInt();
            switch (valueScenario) {
                case 1:
                    System.out.println("Choix scénario 1 : Validé.");
                    Religieux.setSatisfaction(Religieux.getSatisfaction() + fiveone.getEffRel());
                    Ecologiste.setSatisfaction(Ecologiste.getSatisfaction() + fiveone.getEffEco());
                    Loyaliste.setSatisfaction(Loyaliste.getSatisfaction() + fiveone.getEffLoy());
                    //les calculs
                    // affichage résumé / impacts des calculs
                    choicevalueScenario = true;
                    break;
                case 2:
                    System.out.println("Choix scénario 2 : Validé.");
                    Capitaliste.setSatisfaction(Capitaliste.getSatisfaction() + fivetwo.getEffCap());
                    Communiste.setSatisfaction(Communiste.getSatisfaction() + fivetwo.getEffCom());
                    Militariste.setSatisfaction(Militariste.getSatisfaction() + fivetwo.getEffMil());
                    Ecologiste.setSatisfaction(Ecologiste.getSatisfaction() + fivetwo.getEffEco());
                    choicevalueScenario = true;
                    break;
                case 3:
                    System.out.println("Choix scénario 3 : Validé.");
                    Capitaliste.setSatisfaction(Capitaliste.getSatisfaction() + fivethree.getEffCap());
                    Liberaux.setSatisfaction(Liberaux.getSatisfaction() + fourthree.getEffLib());
                    Loyaliste.setSatisfaction(Loyaliste.getSatisfaction() + fivethree.getEffLoy());
                    choicevalueScenario = true;
                    break;
                default:
                    System.out.println("Choissiez parmis les scénarios disponible.");
                    choicevalueScenario = false;
            }

            break;
        }
    } else return false;
    break;
```

Country : this object instantiate the parameters of the country : the name as well as values like industrialization, agriculture, treasury and also the country's food stock.
Getters et setters were also instantiate in order to manage data in the program.

Attributes :

- String name;
- Int value;

```
package src;

public class Country {
    private String name;
    private int value;

    public static Country Indu = new Country( name: "Industrialisation", value: 15);
    public static Country Agri = new Country( name: "Agriculture" , value: 15);
    public static Country Tresor = new Country ( name: "Trésorerie", value: 2000);
    public static Country Food = new Country ( name: "Nourriture" , value: 600);

    public Country(String name, int value){
        this.name = name;
        this.value = value;
    }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    public int getValue() { return value; }

    public int setValue(int value) {
        this.value = value;
        return value;
    }

    public static Country getIndu() { return Indu; }

    public static void setIndu(Country indu) { Indu = indu; }

    public static Country getAgri() { return Agri; }

    public static void setAgri(Country agri) { Agri = agri; }

    public static Country getTresor() { return Tresor; }

    public static void setTresor(Country tresor) { Tresor = tresor; }

    public static Country getFood() { return Food; }

    public static void setFood(Country food) { Food = food; }
}
```

Factions : this object instantiate all the factions existing in El Presidente. It specifies also values such as the satisfaction rate and the number of supporters for each “teams” when we start the game.

Attributes :

- String name;
- Int satisfaction;
- Int partisans;
- Int down;

```java
package src;

public class Factions {

    private String name;
    private int satisfaction;
    private int partisans;
    private int down;
    public static Factions capitaliste = new Factions( name: "Capitaliste", satisfaction: 50,  partisans: 15,  down: 0);
    public static Factions communiste = new Factions( name: "Communiste", satisfaction: 50,  partisans: 15, down: 0);
    public static Factions liberaux = new Factions( name: "Libéraux", satisfaction: 50,  partisans: 15, down: 0);
    public static Factions religieux = new Factions( name: "Religieux", satisfaction: 50,  partisans: 15, down: 0);
    public static Factions militariste = new Factions( name: "Militariste", satisfaction: 50,  partisans: 15, down: 0);
    public static Factions ecologiste = new Factions( name: "Ecologiste", satisfaction: 50,  partisans: 15, down: 0);
    public static Factions nationaliste = new Factions( name: "Nationaliste", satisfaction: 50,  partisans: 15, down: 0);
    public static Factions loyaliste = new Factions( name: "Loyaliste", satisfaction: 100,  partisans: 15, down: 0);

    public Factions(String name, int satisfaction, int partisans, int down) {
        this.name = name;
        this.satisfaction = satisfaction;
        this.partisans = partisans;
        this.down = down;
    }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    public int getSatisfaction() { return satisfaction; }

    public void setSatisfaction(int satisfaction) { this.satisfaction = satisfaction; }

    public int getPartisans() { return partisans; }

    public void setPartisans(int partisans) {
        this.partisans = partisans;
    }
    public int getDown(){return down;}
    public void setDown(int down) { this.down = down; }
}
```

Actions : this one manage the actions wich the players can do in the game. He has the possibility to bride a faction and he can also buy some food rations for his people.

Some attributes of Actions :

- String name ;
- Float amount ;
- EffEco ;
- EffNat ;
- EffLoy ;

```
package src;

public class Actions {

    private String name;
    private float amount;
    private int EffCap;
    private int EffCom;
    private int EffLib;
    private int EffRel;
    private int EffMil;
    private int EffEco;
    private int EffNat;
    private int EffLoy;
    private int EffFood;
    public static Actions goldCapitaliste = new Actions( name: "Pot de vin : Capitaliste", amount: 15, effCap: 10, effCom: 0, effLib: 0, effRel: 0, effMil: 0, effEco: 0, effNat: 0, effLoy: 0, EffFood: 0);
    public static Actions goldCommuniste = new Actions( name: "Pot de vin : Communiste", amount: 15, effCap: 0, effCom: 10, effLib: 0, effRel: 0, effMil: 0, effEco: 0, effNat: 0, effLoy: 0, EffFood: 0);
    public static Actions goldLiberaux = new Actions( name: "Pot de vin : Liberaux", amount: 15, effCap: 0, effCom: 0, effLib: 10, effRel: 0, effMil: 0, effEco: 0, effNat: 0, effLoy: 0, EffFood: 0);
    public static Actions goldReligieux = new Actions( name: "Pot de vin : Religieux", amount: 15, effCap: 0, effCom: 0, effLib: 0, effRel: 10, effMil: 0, effEco: 0, effNat: 0, effLoy: 0, EffFood: 0);
    public static Actions goldMilitariste = new Actions( name: "Pot de vin : Militariste", amount: 15, effCap: 0, effCom: 0, effLib: 0, effRel: 0, effMil: 10, effEco: 0, effNat: 0, effLoy: 0, EffFood: 0);
    public static Actions goldEcologiste = new Actions( name: "Pot de vin : Ecologiste", amount: 15, effCap: 0, effCom: 0, effLib: 0, effRel: 0, effMil: 0, effEco: 10, effNat: 0, effLoy: 0, EffFood: 0);
    public static Actions goldNationaliste = new Actions( name: "Pot de vin : Nationaliste", amount: 15, effCap: 0, effCom: 0, effLib: 0, effRel: 0, effMil: 0, effEco: 0, effNat: 10, effLoy: 0, EffFood: 0);
    public static Actions foodMarket = new Actions ( name: "Acheter une unité de nourriture", amount: 8, effCap: 0, effCom: 0, effLib: 0, effRel: 0, effMil: 0, effEco: 0, effNat: 0, effLoy: 0, EffFood: 1);


    public Actions(String name, float amount, int effCap, int effCom, int effLib, int effRel, int effMil, int effEco, int effNat, int effLoy, int EffFood) {
        this.name = name;
        this.amount = amount;
        this.EffCap = effCap;
        this.EffCom = effCom;
        this.EffLib = effLib;
        this.EffRel = effRel;
        this.EffMil = effMil;
        this.EffEco = effEco;
        this.EffNat = effNat;
        this.EffLoy = effLoy;
        this.EffFood = EffFood;
    }
}
```

Events: Events is an enum, which is the list of the different event that can happen during the game randomly.
We have described the event, their specification and constraint.

All the attributes:

- String specs ;
- Int constraint;
- Int alused;

```java
package src;

public enum Events {
    ONE( Specs: "Des extra-terrestes apparaissent et prennent contact, ils souhaitent s'adresser au chef terrien :",  constraint: 0,  alused: 0),
    TWO( Specs: "Des hackers ont piraté des secrets d'état et réclament une rançon de 250 millions d'euro :",  constraint: 0,  alused: 0),
    THREE( Specs: "Une pandémie touche mondialement :",  constraint: 1,  alused: 0),
    FOUR( Specs: "Une nouvelle source d'énergie verte illimité est découverte :",  constraint: 0,  alused: 0),
    FIVE( Specs: "Suite à un incendie ayant dévaster une grand partie de la faune et la flore du pays :",  constraint: 2,  alused: 0),
    SIX( Specs: "Votre fille vient d'être enlevée par des terroristes, ils vous demandent une rançon de 10millions d'euros :",  constraint: 0,  alused: 0),
    SEVEN( Specs: "Une grève des travailleurs du bâtiment dure depuis 2 semaines :",  constraint: 0,  alused: 0),
    EIGHT( Specs: "Une forte canicule impacte les agriculteurs, ainsi que les personnes agées :",  constraint: 3,  alused: 0),
    NINE( Specs: "Des réfugiés de guerre provenant du pays voisins demandent votre protection :",  constraint: 0,  alused: 0),
    TEN( Specs: "Une météorité va s'écraser très prochainement dans le territoire du pays :",  constraint: 0,  alused: 0),
    ELEVEN( Specs: "Un traité concernant l'émission de CO2 est proposé à échelle mondiale :",  constraint: 0,  alused: 0),
    TWELVE( Specs: "Vous observez une augmentation de présence de punaise de lit à échelle nationale :",  constraint: 0,  alused: 0),
    THIRTEEN( Specs: "Un célèbre et brillant scientifique vous demande des subventions pour ses recherches spatiales :",  constraint: 0,  alused: 0),
    FOURTEEN( Specs: "Une montée alarmante du niveau du fleuve national menace de déborder :",  constraint: 0,  alused: 0),
    FIFTEEN( Specs: "Un coup d'état vous menace, il est orchestré par votre premier ministre :",  constraint: 0,  alused: 0),
    SIXTEEN( Specs: "Un séisme magnitude 7.5 a ravagé une bonne partie de la capitale :",  constraint: 0,  alused: 0),
    SEVENTEEN( Specs: "Une guerre religieuse naît entre les 8 différentes factions présente au sein du pays :",  constraint: 0,  alused: 0),
    EIGHTEEN( Specs: "Les nouvelles élections présidentielles approchent :",  constraint: 0,  alused: 0),
    NINETEEN( Specs: "Votre premier ministre est accusé de détournement de fond public, le peuple n'a pas encore écho de cette information :",  constraint: 0,  alused: 0),
    TWENTY( Specs: "Votre pays est sélectionné pour organiser les prochains Jeux Olympiques :",  constraint: 0,  alused: 0);

    private final String Specs;
    private final int constraint;
    private int alused;
    Events(String Specs, int constraint, int alused){
        this.Specs = Specs;
        this.constraint = constraint;
        this.alused = alused;
    }
    public String getSpecs(){return this.Specs;}
    public int getconstraint(){return this.constraint;}
    public int getAlused(){return this.alused;}
    public void setAlused(int alused){
        this.alused = alused;
    }
}
```

Answers : Answers is an enum, which is the list of the different scenario possible that can happen during the game in response of the event .
We have describe the event, their specification and constraint.

Some attributes of Answers :

- EffCap;
- EffLib;
- EffRel;
- EffCom;