Documentation El Presidente

Table des matières

[Choix d’implémentation 2](#_Toc65410623)

[Structure de l’application 2](#_Toc65410624)

[Application 2](#_Toc65410625)

[ConsoleInput 2](#_Toc65410626)

[ConsoleOutput 2](#_Toc65410627)

[Core 2](#_Toc65410628)

[IndexOfFactionRecuperationMethod 2](#_Toc65410629)

[JsonEventsRepository 2](#_Toc65410630)

[JsonScenariosRepository 2](#_Toc65410631)

[Implémentation des bonus 2](#_Toc65410632)

[Description algorithmes 2](#_Toc65410633)

[Application 2](#_Toc65410634)

[ConsoleInput 2](#_Toc65410635)

[ConsoleOutput 3](#_Toc65410636)

[Core 3](#_Toc65410637)

[IndexOfFactionRecuperationMethod 3](#_Toc65410638)

[JsonEventsRepository 3](#_Toc65410639)

[JsonScenariosRepository 3](#_Toc65410640)

[Description Design patterns 3](#_Toc65410641)

[Schémas 3](#_Toc65410642)

[Exemples pour appréhender la solution 3](#_Toc65410643)

# Choix d’implémentation

## Structure de l’application

Notre application El-Presidente est découpée en plusieurs packages ayant chacun un rôle précis.

### Application

Ce package contient la classe Game s’occupant du déroulement principal du jeu (classe main).

### ConsoleInput

Ce package s’occupe de la vérification des valeurs saisies par l’utilisateur.

### ConsoleOutput

Ce package s’occupe de la construction des affichages console.

### Core

Ce package contient les différentes classes « métier » servant au jeu de façon directe ou indirecte.

Il contient lui-même différents packages pour plus de clarté :

* Activity
* Agriculture
* Enum (contient toutes les enum)
* EventParsers (contient toutes les classes servant à parser les événements json)
* Faction
* Industry
* Input
* Isle
* Output
* ScenarioParsers (contient toutes les classes servant à parser les scénarios json)

### IndexOfFactionRecuperationMethod

Ce package contient les différentes méthodes de récupération d’id de faction au moment de l’ajout/suppression de partisans

### JsonEventsRepository

Ce package s’occupe du parsing des événements json

### JsonScenariosRepository

Ce package s’occupe du parsing des scénarios json

## Implémentation des bonus

### Choix d’événements différents selon la difficulté de la partie

Les choix sont différents selon la difficulté du jeu grâce à des fichiers de configuration associés à chaque difficulté et à chaque scénario. Certains choix avantageux ne sont pas disponibles lors d’une partie difficile et certains choix négatifs ne sont pas disponibles en partie facile.

### Répartition des partisans selon méthodes/algorithmes différents

Les partisans peuvent être répartis selon l’algorithme appliqué au jeu parmi un algorithme aléatoire, un algorithme récupérant la faction avec le plus de partisans et un récupérant la faction avec le moins de partisans.

Cet algorithme est renseigné dans la classe Isle au niveau de l’attribut indexOfFactionRecuperationMethod étant du type de la méthode choisie parmi une des classes implémentées dans le package IndexOfFactionRecuperationMethod.

# Description algorithmes

## Application

Ce package contient la classe Game s’occupant du déroulement principal du jeu (classe main).

## ConsoleInput

Ce package s’occupe de la vérification des valeurs saisies par l’utilisateur.

## ConsoleOutput

Ce package s’occupe de la construction des affichages console.

## Core

Ce package contient les différentes classes « métier » servant au jeu de façon directe ou indirecte.

Il contient lui-même différents packages pour plus de clarté :

* Activity
* Agriculture
* Enum (contient toutes les enum)
* EventParsers (contient toutes les classes servant à parser les événements json)
* Faction
* Industry
* Input
* Isle
* Output
* ScenarioParsers (contient toutes les classes servant à parser les scénarios json)

## IndexOfFactionRecuperationMethod

Ce package contient les différentes méthodes de récupération d’id de faction au moment de l’ajout/suppression de partisans

## JsonEventsRepository

Ce package s’occupe du parsing des événements json

## JsonScenariosRepository

Ce package s’occupe du parsing des scénarios json

# Description Design patterns

# Schémas

# Exemples pour appréhender la solution