Documentation El Presidente

Table des matières

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

Game : déroulement du jeu

- Sélection de la difficulté
- Récupération de la satisfaction minimale (selon la difficulté)
- Sélection et récupération du scénario choisi (fin de l'exécution si les données n'ont pas pu être récupérées)
- Création de l'île
- Déroulement de chaque tour :
    - Passage au tour suivant en début de tour (départ à 0 donc premier à 1)
    - Affichage des informations de l'île
    - Evénement scénario :
        - Récupération de l'événement à la suite ou de l'événement dépendant si l'événement principal n'a pas de nom
        - Affichage des choix et choix de l'utilisateur
        - Application des effets
        - Suppression de l'événement s'il n'a pas d'événement dépendant, suppression de son nom et de tous les choix sauf celui de l'utilisateur s'il y a des événements dépendants afin de garder ces derniers situés dans le choix
    - Evénement bac à sable :
        - Récupération d'un événement aléatoire si le premier événement de la liste n'est pas un événement sans nom
        - Vérification de la validité de la saison
            - Si elle n'est pas valide, on récupère un événement précédent dans la liste
            - Si aucun ne correspondait, on récupère un événement suivant dans la liste
            - Si aucun ne correspondait, on remplit à nouveau la liste d'événements et on réitère
        - Affichage des choix et choix de l'utilisateur
        - Application des effets
        - Suppression de l'événement s'il n'a pas d'événement dépendant, suppression de son nom et de tous les choix sauf celui de l'utilisateur et placement de l'événement en début de liste s'il y a des événements dépendants afin de garder ces derniers situés dans le choix
    - Traitement de fin d'année si c'est l'hiver :

        - Affichage des actions possibles (pot de vin, marché alimentaire, bilan)
        - Fin de l'année lors du bilan
    - Passage à la saison suivante en fin de tour
    - Fin de la partie lors d'un coup d'état :
        - Affichage du score

Core

Pour chaque méthode où il faut diminuer la valeur d'un attribut, si le paramètre a une valeur négative, on le repasse en valeur positive pour ne pas décroitre une valeur négative ce qui reviendrait à augmenter la valeur.

Isle

Score

- 100 points par tour
- 1 point par trésorerie
- 1 point par unité de nourriture
- 1 point par pourcentage d'industrie
- 1 point par pourcentage d'agriculture
- Score x2 en mode normal
- Score x3 en mode difficile

Pot de vin

- Calcul du montant
- Si le montant est suffisant
    - Diminution de la trésorerie
    - Augmentation du pourcentage d'approbation de la faction de 10%
    - Diminution de la satisfaction des LOYALISTS du montant divisé par 10

Marché alimentaire

- Si la trésorerie est suffisante pour acheter toutes les unités de nourriture passées en paramètre
    - Nombre réel d'unités de nourritures achetées augmenté au nombre demandé
    - Diminution de la trésorerie
- Sinon
    - Achat du nombre maximum d'unités de nourriture selon la trésorerie

Bilan de fin d'année

- Génération de la trésorerie grâce à l'industrie
- Génération des unités de nourriture grâce à l'agriculture
- S'il n'y a pas assez de nourriture pour nourrir tous les partisans
    - Elimination des partisans selon la méthode mise en place jusqu'à ce qu'il y ait assez de nourriture pour tous les partisans :
        - Diminution de l'approbation de toutes les factions de 2%
- Sinon si l'agriculture seule est en excédent
    - Diminution de la nourriture servant à nourrir les partisans

    - Natalité entre 1 et 10 % → répartition dans les factions selon la méthode mise en place
- Sinon : diminution de la nourriture servant à nourrir les partisans

Pot de vin est possible ?

- Sans paramètre :
    - Vérification qu'au moins une faction peut bénéficier d'un pot de vin avec la trésorerie actuelle
- Avec paramètre :
    - Vérification que la faction peut bénéficier du pot de vin avec la trésorerie actuelle

Marché alimentaire est possible ?

- Vérification que la trésorerie est suffisante pour payer au moins 1 unité de nourriture

Incrémentation des partisans

- Récupération d'un id de faction selon la méthode mise en place jusqu'à en avoir un correspondant à une faction n'ayant pas aucuns partisans
- Augmentation des partisans de la faction

Décrémentation des partisans

- Récupération d'un id de faction selon la méthode mise en place jusqu'à en avoir un correspondant à une faction n'ayant pas aucuns partisans
- Diminution des partisans de la faction

Incrémentation de l'industrie

- Augmentation du pourcentage de l'industrie
- Si l'industrie et l'agriculture dépassent 100%
    - Diminution de l'agriculture pour avoir un cumul à 100%

Incrémentation de l'agriculture

- Augmentation du pourcentage de l'agriculture
- Si l'industrie et l'agriculture dépassent 100%
    - Diminution de l'industrie pour avoir un cumul à 100%

IndexOfFactionRecuperationMethod

RandomRepartitionMethod

- Retourne un indice aléatoire entre les indices des factions

MostPartisansRepartitionMethod

- Retourne l'indice de la faction avec le plus de partisans

LessPartisansRepartitionMethod

- Retourne l'indice de la faction avec le moins de partisans en dehors des factions avec 0 partisans

JsonScenariosRepository

- Récupération de tous les scénarios dans le fichier json pour l'affichage menu :
    - Récupération de l'id, du nom et de l'histoire de chaque scénario uniquement

- Récupération de tous les id des scénarios dans le fichier json :
    - Récupération de l'id de chaque scénario

- Récupération du scénario intégralement selon l'id et la difficulté
    - Récupération des informations principales
    - Parsing des paramètres de début de jeu :
        - Récupération des différents paramètres
        - Parsing des paramètres des factions
            - Récupération des paramètres pour chaque faction
    - Parsing des événements du scénario :
        - Cf partie suivante
    - Parsing des événements neutres :
        - Cf partie suivante
    - Création du scénario

JsonEventsRepository

- Récupération de tous les événements selon l'id du scénario et la difficulté de la partie (exclusion du bac à sable)
    - Récupération du bon fichier json selon l'id et la difficulté
    - Parsing des événements en une liste :
        - Parsing des choix :
            - Récupération du libellé
            - Parsing des effets :
                - Récupération des effets sur les factions
                - Récupération des effets sur les facteurs
                - Récupération de l'effet sur les partisans
            - Parsing des événements dépendants s'il y en a :
                - Même fonctionnement que le parsing des événements
- Récupération de tous les événements neutres (bac à sable) :
    - Récupération du bon fichier json selon la difficulté
    - Parsing des événements en une liste :
        - Récupération des saisons
        - Parsing des choix :
            - Récupération du libellé
            - Parsing des effets :
                - Récupération des effets sur les factions
                - Récupération des effets sur les facteurs
                - Récupération de l'effet sur les partisans
            - Parsing des événements dépendants s'il y en a :
                - Même fonctionnement que le parsing des événements