**Rapport**

L2 Info Groupe 2

Yeqian HE Jincheng YE

Objectif : on construit le jeu l’île interdite.

**Artefacts :**

* 4 artefacts, chacun lié à un élément (air, eau, terre, feu).

**Zone :**

* L’île est formée par un ensemble de zones disposées en grille.

Les zones particuliers :

1.un héliport

2.des zones associées à chacun des éléments, qui permettent de récupérer l’artefact correspondant.

Chaque zone est dans l’une des situations suivantes : « normale » , « inondée » ou « submergée » .

* Inonder zones :

**Clé :**

* Chercher une clé :

**Personnages particuliers :**

1.Pilote 2.Ingénieur 3. Explorateur

4.Navigateur 5. Plongeur 6.Messager

**Le tour d’un joueur :**

* Effectuer les actions normales :

1.se déplacer ( vers une zone adjacente non submergée ) :

2.assécher la zone ( pas submergée ) :

3.récupérer un artefact ( zone d’artefact & clé correspondante) :

4.Echanges de clés (les joueurs présent dans la même zone) :

* Actions spéciales :

1.Sac de sable :

2.Hélicoptère :

**Succès :**

* 4 artefacts sont récupérés & tous les joueurs sont regroupés à l’héliport.

**Vues :**

1. Afficher les zones de l’île et leur état (class Game)
2. Un bouton « fin de tour » qui peut procéder à l’inondation de trois zones (non encore submergées ) aléatoires.(class Game -> start())
3. Le modèle avec un joueur (class Joueur)
4. La vue représente le joueur sur sa zone (class Game -> draw())
5. Étendre la vue et le contrôleur avec des moyens d’ordonner le mouvement du joueur. On choisit clics de boutons (class Game -> direction())
6. Étendre le contrôleur pour que le joueur puisse réaliser au maximum trois actions avant de cliquer sur le bouton « fin de tour ». (class Game -> direction(), on définit « click » pour limiter les étapes.)
7. Étendre l’ensemble pour permettre plusieurs joueurs. Le contrôleur doit faire agir les joueurs à tour de rôle, en insérant une phase d’inondation (« fin de tour ») à la fin du tour de chaque joueur. (class Game -> Joueur[] permet d’ajouter plusieurs joueurs)
8. Étendre l’ensemble pour ajouter aux coups du joueur l’action d’assécher la zone sur laquelle il se trouve ou une zone adjacente, en choisissant une interface utilisateur cohérente avec celle permettant le déplacement.(class Game -> direction(), Button « assécher »)
9. La vue indique le joueur dont c’est le tour et le nombre d’actions auxquelles il a encore droit. (class Game -> direction(), on utilise « label » pour stocker)
10. Étendre le modèle pour inclure les zones spéciales, les clés que peut posséder un joueur, et l’action de récupérer un artefact. Étendre l’effet du bouton «fin de tour» pour que le joueur dont c’était le tour reçoive, aléatoirement, une cléou rien. (class Game -> getCle() pour obtenir cle dans le zone spécial)
11. La vue d’un panneau représentant les joueurs, et y indiquer pour chacun les clés et artefacts qu’il possède. (class Game -> cle(), affiche les clés possède de chaque joueur.)
12. Étendre le contrôleur pour inclure l’action de récupérer un artefact et la fin de partie gagnante. (class Game -> isWinning())
13. Les critères caractérisant les parties perdues et arrêter la partie lorsqu’ils se réalisent. (class Game -> isFail())

**Simulation d’un vrai paquet de cartes :**

1. Deux paquets dédiés (un paquet contenant une carte par zone du jeu, et un paquet contenant un certain nombre de cartes pour chaque issue de la rechecherche de clé)
2. Les opérations :

mélanger le paquet

tirer la première carte

poser une carte dans la défausse

mélanger la défausse

la replacer au sommet du paquet ( réalisée automatiquement lorsque la

dernière carte du paquet a été tirée)

3.

**Difficultés :**

1. Penser l’ordre de programmation des différents éléments.
2. Visualisation des cartes et des joueurs.
3. Deplacer les joueurs partager.
4. Renouveler l’image chaque tour.
5. File de images est toujours incorrect.