

THE ISLAND

**CAHIER
DES
CHARGES**



I - Contexte

The Island est un jeu de société familial et dynamique. The Island est un jeu qui mélange stratégie, hasard et coups bas, dans une ambiance survoltée. On se prend vite au jeu de tenter de sauver ses explorateurs tout en gênant ceux des autres. Mais ce jeu n'existe que sous forme de jeu plateau, alors pour y jouer il faudra l'avoir acheter. Après cela il faudra mettre en place ce jeu qui prendra de la place et le ranger quand la partie sera finie. Ce qui peut s'avérer pénible.

II - Description de la demande

a - Objectifs

Pour palier à ses contraintes, une adaptation sur pc de ce jeu de plateau sera créée par nos soins en java, cela qui va permettre des expériences plus accessibles sans rien payer. Et qui enlève les soucis de configuration et de rangement du jeu de plateau. De plus cela offrira plus d'espace, ce qui permettra de jouer plus confortablement.

b - Les fonctions du produits

- Aider les joueurs débutants à assimiler les règles avec un onglet help qui explicitera tout cela
- Générer le plateau sur lequel la partie va se jouer
- Générer les 40 tuiles de l'île, les 40 pions explorateurs, 5 pions Serpent de Mer, 6 pions Requin, 5 pions Baleine
- Chaque tuile et pion doit pouvoir faire ce pourquoi il a été fait

II - Description de la demande

b - Les fonctions du produits

- Donner un ordre de tour de façon aléatoire
- Faire une action: cela consiste à jouer une tuile qui nous a rapporter si on le veut et qu'on en possède une et de jouer ses 3 points de déplacement
- Faire une destruction: cela consiste à retirer une tuile de l'île et jouer l'évènement se trouvant à la face caché ou récupérer son bonus
- Monstre: cela consiste à jouer le dé monstre et à déplacer le monstre obtenu, et faire agir le monstre si possible.
- Faire tourner la main selon l'ordre donné au départ
- onglet help pour aider les débutants à assimiler les règles du jeu
- Finir le jeu avec l'évènement éruption volcanique
- Compter les points de chacun et afficher le vainqueur

II - Description de la demande

c - Critères d'acceptabilité et de réception

Notre version du jeu The Island implémenté en java doit pouvoir permettre de vivre la même expérience de jeu, en ayant bien évidemment aucun bugs. L'interface graphique doit être la plus ergonomique possible. Le jeu doit pouvoir faire jouer 4 joueurs grand max qui joueront l'un après l'autre et pas en réseau.

III - Contraintes

a - Contrainte de délai

Dates importantes	Tâche à accomplir
29 Mai 2024	Remise du rapport final, du code source du projet, des binaires et du support de la présentation
30 Mai 2024	soutenances

III - Contraintes

b - Contraintes techniques

Langage d'implémentation	Java
Utilisation optimale et maximale de java	Les collections, exceptions devront être utilisées au mieux
Bibliothèque de l'interface graphique	Swing
Convention de codage	SUN
Outil de documentation du code	javadoc

III - Contraintes

b - Contraintes techniques

**Compilation facile
du livrable**

Le projet final devra
fonctionner et
se compiler en
dehors de tout IDE

Format du livrable

format jar et «
exécutable »

IV - Déroulement du projet

a - Tâches à accomplir

? optionnel tant que les tâches plus primordiales n'ont pas été achevées

Il y aura environ 20 tâches ou plus à faire si ne compte pas la menu du jeu.

? Créer le menu de notre jeu

- Créer l'onglet help que l'on retrouvera dans le menu et même en pleine partie;
- Créer le plateau de jeu qui aura:
 - 4 plages au coin du plateau
 - la mer
 - 5 pions serpent de mer, 1 au milieu et chacun des 4 autres en face d'une plage
 - la zone au milieu délimitée en noir qui contiendra l'île;

IV - Déroulement du projet

a - Tâches à accomplir

- Créer une class position qui va garder les coordonnées de chaque élément sur le plateau
 - class Position:
 - Attributs
 - X: int
 - Y: int
 - Méthodes
 - getCoordonnée()
 - setCoordonnée()

IV - Déroulement du projet

a - Tâches à accomplir

- Créer les tuiles de l'île:
 - class Tuile extends Position:
 - Attributs
 - - niveau engloutissement: int
 - - face caché: Face caché
 - super(X, Y)
 - Méthodes
 - + get niveau engloutissement(): int
 - + get tuile(): Tuile
 - + get face caché (): Face caché
 - //On réécrit la fonction qui change les coordonnées en une fonction qui ne fait rien car cet élément ne se déplace pas
 - @override
 - set coordonnée(){
 - }
 -

IV - Déroulement du projet

a - Tâches à accomplir

- class Plage extends Tuile:
 - Attributs
 - super(1, face caché, X, Y)
- class Forêt extends tuile:
 - Attributs
 - super(2, face caché, X, Y)
- class Montagne extends tuile:
 - Attributs
 - super(3, face caché, X, Y)

IV - Déroulement du projet

a - Tâches à accomplir

- class Face caché extend Position
 - Attributs
 - type: bonus ou évènements : String
 - super(X, Y)
 - methodes
 - @classmethod bonus()
 - @classmethod évènement()
 - action du bonus ou de l'évènement
 - //On réécrit la fonction qui change les coordonnées en une fonction qui ne fait rien car cet élément ne se déplace pas
 - @override
 - set coordonnée(){
 - }
- Créer le dé monstre:
 - Avec équiprobabilité donc 1 chance sur 6 de tomber sur n'importe quelle face
 - deux faces opposées serpent de mer
 - deux faces opposées baleine
 - deux faces opposées requin

IV - Déroulement du projet

a - Tâches à accomplir

- Créer les pions monstres
 - class Monstre extend Position
 - Attributs
 - - nom du monstre: String
 - super(X, Y)
 - Methodes
 - + get nom monstre(): String
 - 5 serpents de mers
 - class Serpent de mer extend Monstre
 - Attributs
 - super("serpent de mer" , X, Y)
 - Methodes
 - + dévorer explorateurs et bateaux(): void

IV - Déroulement du projet

a - Tâches à accomplir

- 6 requins
 - class Requin extend Monstre
 - Attributs
 - super(“requin” , X, Y)
 - Methodes
 - + dévorer explorateurs(): void
- 6 baleines
 - class Baleine extend Monstre
 - Attributs
 - super(“baleine” , X, Y)
 - Methodes
 - + couler bateau(): void

IV - Déroulement du projet

a - Tâches à accomplir

- Créer les 12 pions bateaux
 - class Bateau extend Position
 - Attributs
 - - capacité_bateau = 3
 - Méthodes
 - + get capacité bateau(): int
 - //On va diminuer la capacité du bateau quand un explorateur monte dans celui ci
 - + diminuer capacité bateau(): void
 - Créer les 40 pions explorateurs
 - class Explorateur extend Position:
 - Attributs:
 - - points de trésor: Int
 - super(X, Y)
 - Méthodes
 - + get points de trésor()

IV - Déroulement du projet

a - Tâches à accomplir

- - class Explorateurs extend hashmap:
 - //Ce sera un dictionnaire, qui à qui le pseudo d'un joueur retourne ses explorateurs
 - //Cela va contenir les 10 explorateurs avec chacun leur point de trésor
 - Faire le code qui génère l'île avec ses tuiles et ses pions
 - Faire le code qui va donner l'ordre de tour de la partie aléatoirement
 - Faire le code qui va permettre de faire de jouer son bonus si on en a un et de jouer ses 3 points de déplacements
 - Faire le code qui permette de prendre une tuile par ordre d'engloutissement
 - Faire le code qui permettra de garder son bonus ou de jouer l'évènement de la face caché de la tuile

IV - Déroulement du projet

a - Tâches à accomplir

- Faire le code qui permette de faire jouer le dé monstre
- Faire le code qui permettra de ne faire bouger le monstre indiqué par le dé monstre
- Faire le code qui permette de faire agir le monstre s'il est sur une case occupé par un ou plusieurs explorateurs et/ou un bateau
- Faire le code qui va faire tourner la main selon l'ordre donné au départ
- Faire le code qui fera que l'évènement éruption volcanique mette fin à la partie
- Faire le code qui va compter les points des survivants de chaque joueur, les afficher et montrer le gagnant

