

RAPPORT POUR LE MINI PROJET D'IPO

CHARGEÉE DE TP : THI THUONG HUYEN NGUYEN

ETUDIANTES : NOEMIE HANUS ET FATIMA-ZAHRA MERIMI

GROUPE N° 2

Ce rapport résume le travail effectué ; les points de difficulté rencontrés et les solutions envisagées/réalisées.

Ce rapport résume le travail effectué ; les points de difficulté rencontrés et les solutions envisagées/réalisées.

Difficultés générales:

L'impossibilité de se voir et de discuter de vive-voix a bien sûr ajouté une difficulté supplémentaire. De plus, nous avons mis un certain temps avant de bien comprendre le fonctionnement de Git.

PARTIE 1 : Frog

Fichier(s) : Frog; Game

Le travail effectué: nous avons implémenté la classe Frog, à partir de l'interface IFrog, et nous avons implémenté les fonctions testWin() et testLose() dans la classe Game.

Les points de difficulté rencontrés et les solutions envisagées/réalisées: Cette partie était relativement simple cependant j'ai eu un peu de mal à réaliser qu'il fallait simplement coder les méthodes décrites dans IFrog.

PARTIE 2 : Environnement

Fichier(s) : Car ; Lane ; Environment

Le travail effectué: Nous avons implémenté les classes Car, Lane et Environment, à partir des squelettes fournis et indications sur le sujet. Nous avons changé la visibilité de addToGraphics() à protected pour pouvoir l'utiliser dans la classe Lane.

Les points de difficulté rencontrés et les solutions envisagées/réalisées: Dans cette partie, on a essayé deux types de raisonnement, l'un ascendant et l'autre descendant. On a finalement opté pour l'ascendant.

Les difficultés rencontrées étaient liées à la méthode isSafe(), on a essayé de changer d'algorithme puis finalement il fallait considérer tous les cas de figures.

Nous avons eu des problèmes de voitures qui se chevauchaient ou se créaient côte à côte, donnant l'illusion de voiture trop grande, nous les avons réglées en modifiant la fonction mayAddCar().

Nous avons eu des problèmes d'affichages, avec des voitures qui "clignotaient", dû à un appel de addToGraphics() au mauvais endroit.

PARTIE 3 : Jeu infinie

Fichier(s) : EnvInf ; FrogInf ; Game

Le travail effectué: Dans la classe Case, nous avons changé la visibilité l'attribut ord de private final à private. Nous avons considéré EnvInf comme classe fille de Environment. Cependant, nous avons recodé update de EnvInf et supprimer le isWinning() dans l'interface IEnvironment, et ajouté dans Game la position de frog dans testLose()

Les points de difficulté rencontrés et les solutions envisagées/réalisées: Nous avons rencontré des problèmes dès la conception de l'algorithme, nous avons pensé que l'utilisation d'une très grande arraylist allait prendre beaucoup de temps à s'afficher. Après ça, on a essayé plusieurs façons pour update l'affichage, finalement on s'est résigné à update le premier affichage au début de update() et après chaque mouvement vers le haut ou le bas de frog, on update l'affichage dans le update de la classe Lane.

PARTIE 4 :

Fichier(s) : Game

Le travail effectué: On a essayé de trouver une classe qui nous aide à calculer la durée de la partie; par exemple Duration ou Time ou Date. Finalement, on a utilisé System.currentTimeMillis().

Les points de difficulté rencontrés et les solutions envisagées/réalisées: Nous avons essayé d'utiliser Time, il fallait définir une classe TimerTask et on a eu des erreurs de répétition du coup il fallait utiliser cancel().