

CPS TME6 - Spécification du projet Lemmings

Le but du projet est de réaliser une version sûre du jeu de réflexion *Lemmings*, développé par *DMA Design* et édité par *Psygnosis* en 1991 en utilisant les approches *Design by Contract* et *Model-Based Testing*.

Le but du **TME6** est de rendre un squelette de projet comprenant une spécification du projet couvrant au moins **les questions 1, 2, 3 et 4 de l'examen réparti et le service Joueur** (sans implémentation, sans contrat, sans test).

Consignes (projet)

- L'objectif du projet est de réaliser un jeu sûr en écrivant des contrats cohérents, complets et non-redondants et des jeux de test pertinents.
- Une interface minimum (permettant de jouer) doit être implémentée. Une interface graphique esthétique et ergonomique (par exemple, gestion de la souris) est bienvenue, mais pas obligatoire.
- Le projet doit être écrit en Java et doit suivre l'architecture des TME 3, 4, 5: une archive `.jar` contenant (entre autres) un `build.xml` pour la construction, l'exécution et le test du projet avec Ant, une répartition du code en services, décorateurs, contrats, implémentations et tests.
- Au moins deux implémentations doivent être fournies dans le projet: une implémentation correcte (qui valide tous les contrats) et une implémentation avec des erreurs (mettant en évidence la pertinence des contrats).

Attentes (projet)

Projet: Le projet doit couvrir au moins l'intégralité du sujet de l'examen réparti (dernière question *Lemmings spéciaux* incluse) et le service Joueur décrit ci-dessous. Les extensions décrites ci-dessous seront, bien entendu, valorisées.

Rapport: Le rapport (d'une dizaine de page) doit comporter un manuel d'utilisation et un bilan qui doit décrire la structure du projet et les choix faits lors de la réalisation, en insistant sur les choix de spécification, la mise en oeuvre des contrats et la stratégie de test. Le rapport, doit en outre, contenir des exemples commentés de test.

Soutenance: La soutenance doit résumer les choix de spécification, d'implémentation et de test et doit être accompagnée d'une démonstration (en direct) du projet.

Service: Joueur (projet et TME6)

Le service joueur doit implémenter le contrôle qu'a le joueur sur une partie. Au début, le joueur dispose d'un certains nombre de **jetons** pour chaque classe (à l'exception de tombeur).

A tout moment, le joueur peut choisir un lemmings (il faut choisir un type d'interface pour la sélection) et dépenser un jeton d'une classe pour transformer ce lemmings dans cette classe.

Le joueur doit disposer d'un moyen de remettre à zéro la partie à tout moment.

Extensions (projet)

Des extensions possibles sont proposées ici (la liste n'est évidemment pas exhaustive):

- **Vitesse de création:** Le joueur peut influencer dynamiquement sur la vitesse d'entrée `spawnspeed` des lemmings.
- **Annihilation:** Le joueur peut décider de transformer tous les lemmings actifs en explosif et stopper l'entrée des lemmings, afin de relancer la partie (et voir la colonie exploser plutôt que de relancer directement la partie).
- **Case speciale: Eau:** ajout de cases d'eau qui subit la gravité (coule). Par exemple si on détruit la case en dessous d'une case d'eau, l'eau s'écoule par le "trou". La quantité d'eau d'un niveau reste donc constante. Un lemming meurt si la case au dessus de sa position (celle où graphiquement se trouve sa tête) est dans l'eau pendant plus de 8 tours successifs.
- **Case speciale: Lave:** ajout de cases de lave se comportent comme de l'eau à l'exception qu'un lemming meurt si la case où il se trouve, ou la case juste au dessus contiennent de la lave.
- **Case speciale: Unidirectionnelle:** ajoute de case que l'on ne peut creuser que dans une seule direction (gauche / droite / bas).
- **Nouvelles classes:** ajout de nouvelles classes (cf. documentation de *Lemmings 2: the Tribes* sur le Web pour des exemples).
- **Combat:** ajout d'ennemis qui se déplacent, tuent les lemmings au contact et peuvent être tués par de nouvelles classes de lemmings.

Soumission (TME6)

Soumettre à l'adresse (et aux dates) habituelle(s) une archive `.jar` aux noms du binôme contenant:

- (obligatoire) un squelette de projet pour les différents services et les différentes couches (services, implémentations, contrats, tests) inspirés des TME précédents,
- (obligatoire) la définition avec spécification (en commentaire) des services liés aux questions 1, 2, 3 et 4 du partiel,
- (facultatif) la définition des autres services,
- (facultatif) un diagramme de classe et un diagramme de composants du système.