Lasers and Mirrors
Programación Orientada a Objetos

Trabajo Práctico

Federico Bond
ndelegajo

Kevin J. Hanna
ndelegajo

Fecha de Entrega: 4 de Noviembre, 2011

Resumen

Índice

1. Diseño

La implementación del juego Lasers and Mirrors intenta recrear una arquitectura MVC (Model, View y Controller). En el modelo (Game.java), se encuentra toda la lógica de negocios del juego. Las vistas, en nuestro caso las clases que implementan View, sirven para interactuar con el usuario. El controlador (GameController.java) es un intermediario entre las vistas y el modelo. Este sigue siendo de frontend y se encarga de llamar al modelo que está en el backend.

1.1. Game

2. Problemas

2.1. Organización

Decidimos utilizar github (git) como repositorio para nuestro proyecto y las decisiones tomadas por chat en gmail para tenerlas documentadas.

2.2. Reset

Al principio, diseñamos un opción de restart. El problema fue que esto nos llevo a armar una estructura que almacenaba como se guardaban las tiles inicialmente, para poder resetear el juego desde un juego cargado sin el archivo board original. Con esta estructura armada nos empezó a demorar y ensuciar el código de cargar un juego previamente guardado. Solución: cut down features...

2.3. Forma imperativa del parser

Para poder consumir el archivo board tuvimos que armar toda una secuencia lógica bastante imperativa..

3. Decisiones de desarrollo

3.1. Game vs. Board

Una de las decisiones que tomamos fue hacer un tablero genérico (con celdas), independiente del juego para el que se use. Es decir, las constantes de tamaño máximo y el método para validarlo se encuentran en la clase Game y no en la clase Board.

3.2. Método de guardar y cargar juego

Inicialmente, la clase Game contaba con métodos para cargar y guardar una partida. Decidimos armar un paquete IOGame para organizar todas las

opciones de guardado y cargado. Aunque en nuestro caso solo es posible mediante serialización, se podría extender a otras formas.[1]. De la misma forma se llegó a la conclusión de separar el parser de Game.

[1]Habría que armar un controlador con los métodos save y load, pero al ser static es imposible en esta versión de java