

README

Marco Antonio García Arce
Juan Carlos Zenteno Pompa

Enero 2022

1 Problemática

Cheems's Prison Simulator. La compañía Cheems Games nos ha encargado un simulador básico de una prisión del cual usaran como base para crear juegos en el futuro.

- El simulador debe de dar la opción de agregar presos y guardias. Cada preso y guardia debe de tener un nombre. Igualmente debe de permitir eliminar al preso o al guardia al gusto.
- Los presos deben de cumplir con las acciones que le corresponden y alguna que difieren de los guardias. Los guardias deben de responder y actuar si se encuentra un preso actuando de manera agresiva.
- El usuario puede ver la simulación con objetos representativos. Esto para observar como se comporta cada entidad. El simulador debe de poder mostrar la información basica del personaje al que se desee revisar.

2 Patrones

Tenemos los siguientes patrones:

- MVC Este esta presenta para que podamos tener claramente una division para modificar ya sea las funciones o la presentación del programa y tener una mejor organización. MVC involucra las clases Controlador, Prison, UIPanel y SimulationPanel.
- State Usamos el patron de state para nuestras entidades. Tenemos dos tipos de entidades en nuestra simulación que son los guardias y los presos. Cada uno tiene comportamientos distintos y en cada estado realizara dichas acciones. State involucra las clases de Preso y de Guardia, al igual de los comportamientos de Agresivo, Durmiendo, y Paseando para el preso, mientras que para el guardia los estados patrullando y persiguiendo.

- Observer Usamos observer para modificar el estado de los personajes dependiendo de ciertas acciones. Los guardias deben de actuar cuando un preso se vuelve agresivo y los presos deben de ser eliminados una vez neutralizados. Observer involucra las interfaces AgresionObserver, AgresionSubject, NeutralizationObserver, NeutralizationSubject, los estados de guardia Patrullando y Persiguiendo, ademas del estado de preso Agresivo
- Iterator Este patrón es utilizado para las estructuras que se encuentran en ciertas clases para acceder a los objetos de ellos y no a los objetos de los objetos. Iterator involucra las clases Prisión, Controlador, GuardControl y IDList.
- Strategy Strategy es implementado para procesar la acción que se realiza cuando el usuario interactua con el panel de la simulación. Strategy involucra la interfaz de ClickStrategy, ademas de las clases de ClickModes, Enrage, Euthanasia, Select, SpawnGuardia, SpawnPreso y UVoid.

3 Ejecución

Primero se tiene que estar en la carpeta de src. Para compilar es el siguiente:

```
$ javac $(find proyecto/* | grep .java)
```

Para ejecutar el programa se coloca lo siguiente al estar en la carpeta src:

```
$ java proyecto/Main
```