Monopoly

Álvaro Nieves Carretero

[alvaronieca@gmail.com](mailto:alvaronieca@gmail.com)

1. Descripción del problema

El cliente ha solicitado que se desarrolle el juego del Monopoly basado en servicios. Como metodología de gestión del proyecto vamos a utilizar Scrum. El lenguaje con el que desarrollaremos el juego será JavaScript. Las herramientas que utilizaremos para gestionar el proyecto, controlar las versiones y desarrollar el software serán Scrum, Git y SublimeText respectivamente.

1. Sprint 1
   1. Reunión de preparación del sprint

### Reunión con el Product Owner

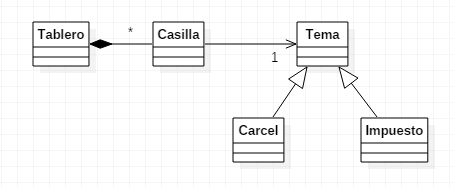
El objetivo principal de este Sprint es crear el Tablero, para ello crearemos los objetos Tablero, Casilla y un “tema” por cada tipo de casilla (Cárcel, Impuesto, Calle…).

Se creara una historia llamada “tablero con pruebas”.

### Reunión del equipo

El equipo de desarrollo se reúne para dividir la historia en tareas. De la historia “tablero con pruebas” se crean las siguientes pruebas con una estimación en horas:

* Aprender Jasmin 🡪 1 hora
* Picar código del tablero 🡪 3 horas
* Codificar las pruebas 🡪 2 horas
* Pasar las pruebas 🡪 1 hora
  1. Desarrollo del Sprint



En el diseño del modelo podemos ver que Tablero tiene muchas Casillas, que cada Casilla tiene un Tema, y que Tema puede ser una Cárcel, un Impuesto, una Calle, una Estación, una CajaComunidad, una Sorpresa, un ImpuestoLujo, un Parcking o VolverCarcel.

Para iniciar el juego primero tenemos que crear un tablero con 40 casillas y en cada casilla crear un tema, para ello voy a mostrar una prueba de cada tipo de tema donde se verán los atributos de cada tema.

Con esta prueba comprobamos que se ha creado el tablero y que cuenta con 40 casillas.

describe("Crear el juego",function(){

it("debería existir tablero",function(){

expect(tablero).toBeDefined();

});

it("debería existir 40 casillas",function(){

expect(tablero.numeroCasillas).toEqual(40);

});

});

Aquí se prueba que la casilla 1 se corresponde con la casilla 1 del tablero, una calle. Se comprueba el nombre de la calle, el color y el precio.

it("en la casilla 1 debería estar la calle RONDA DE VALENCIA",function(){

expect(tablero.casillas[1].tema.nombre).toEqual('RONDA DE VALENCIA');

expect(tablero.casillas[1].tema.color).toEqual('MARRON');

expect(tablero.casillas[1].tema.precio).toEqual(60);

});

Aquí se prueba que la casilla 0 se corresponde con la casilla 0 del tablero, la salida.

it("en la casilla 0 debería estar la casilla de Salida ",function(){

expect(tablero.casillas[0].tema.nombre).toEqual('Salida');

});

Aquí se prueba que la casilla 5 se corresponde con la casilla 5 del tablero, una estación.

it("en la casilla 5 debería estar la estacion de goya ",function(){

expect(tablero.casillas[5].tema.nombre).toEqual('ESTACION DE GOYA');

});

Aquí se prueba que la casilla 2 se corresponde con la casilla 2 del tablero, una caja de comunidad.

it("en la casilla 2 debería estar la caja de comunidad ",function(){

expect(tablero.casillas[2].tema.nombre).toEqual('Caja de comunidad');

});

Aquí se prueba que la casilla 10 se corresponde con la casilla 10 del tablero, la carcel.

it("en la casilla 10 debería estar la carcel",function(){

expect(tablero.casillas[10].tema.nombre).toEqual('Carcel');

});

Aquí se prueba que la casilla 20 se corresponde con la casilla 20 del tablero, el parcking.

it("en la casilla 20 debería estar el parcking",function(){

expect(tablero.casillas[20].tema.nombre).toEqual('Parcking');

});.

Aquí se prueba que la casilla 4 se corresponde con la casilla 4 del tablero, un impuesto.

it("en la casilla 4 debería estar el impuesto sobre capital ",function(){

expect(tablero.casillas[4].tema.nombre).toEqual('IMPUESTO SOBRE CAPITAL');

});

* 1. Reunión al finalizar el Sprint

### Reunión de revisión

El equipo de desarrollo muestra al Product Owner el desarrollo durante el Sprint. Se muestra que las pruebas se ejecutan correctamente y se puede probar la creación correcta del tablero.

### Reunión retrospectiva

Nos reunimos con el Scrum Master y comentamos todos los Sprint Burndown para ver el tiempo que hemos empleado en desarrollar el primer Sprint y así aprender a estimar mejor para futuros sprints.

Anexo A – PDF del primer Sprint creado por kunagi

