Impostor

Procesos de Ingeniería del Software

Juan Manuel Verdejo Utrilla

Tabla de contenido

Capítulo 4. Desarrollo del proyecto

En este capítulo se va a contar mediante los diferentes Sprints como ha sido llevado el proceso de desarrollo.

En cada uno de estos Sprints se describirá la reunión de preparación, el desarrollo del Sprint, y la revisión y retrospectiva al finalizar el Sprint.

Sprint 1. Sprint inicial: lógica básica del juego.

Reunión de preparación del Sprint

a) Con el Product Owner:

El objetivo del Sprint inicial será el de definir la lógica básica del juego, de forma que tratamos de imitar el funcionamiento del recientemente popular juego, “Among Us”.

El Product Owner (alumnos) debido a su experiencia previa con el juego ha ayudado a describir así los requerimientos necesarios para lograr este funcionamiento.

Se han definido las siguientes historias de usuario para este Sprint:

- Crear partida
- Unir a partida
- Abandonar partida
- Asignar impostor
- Asignar tareas
- Atacar
- Votaciones

b) Del equipo:

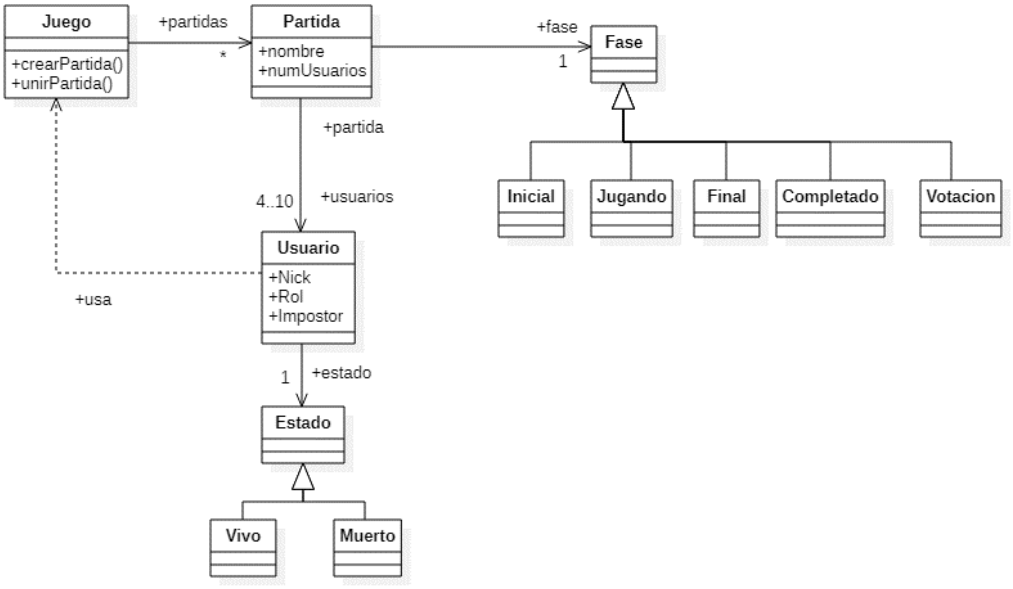
El Sprint ha tenido una duración de 15 días, desde el 7 de octubre al 22 de octubre, en los cuales se han estimado 22 horas de trabajo.

Para cada historia de usuario se han establecido 3 tareas: diseño, implementación y pruebas.

Además de esas tareas genéricas de cada historia de usuario también se han obtenido conocimientos sobre lenguajes de programación: una tarea para HTML y JavaScript; y otra para jQuery.

Desarrollo del Sprint

En las tareas del diseño se ha obtenido el siguiente diagrama de clases para la definición del modelo lógico del juego:



Problema: Gestión de código.

Solución: Se está subiendo el código fuente a un repositorio público en GitHub.

Problemas: Partidas.

Solución: Para almacenar las partidas se ha optado por implementarlas mediante una lista asociativa que se accede por el código.

Problema: Requisitos para unirse a partida o iniciar la partida.

Solución: Para permitir o no la unión de jugadores se ha implementado el patrón de diseño State de forma que cuando la partida ya está completa o ya se esté jugando no puedan unirse más jugadores.

Problema: Implementación ataques.

Solución: Se ha optado por implementar un patrón State para el usuario para controlar cuando un usuario impostor puede atacar o ser atacado, según este muerto o vivo. Se comprueba mediante la propiedad “impostor” del usuario si se puede atacar.

Problema: Votaciones.

Solución: Se ha incluido una nueva fase “Votación“ para la realización de la votaciones, permitiendo a los usuarios votar solo cuando la partida se encuentre en esa fase. Para el cuento de los votos se ha incluido un atributo numérico “votos” y un booleano “skip” a cada usuario.

Problema: Asignación de tareas.

Solución: Las tareas están definidas en la lista tareas (cada tarea una cadena de caracteres) de la partida y a cada usuario se le asigna una tarea de forma secuencial una tarea colocando la cadena en el atributo “tarea” del usuario.

Problema: Asignación de impostor.

Solución: Se pone a verdadero el atributo impostor de un usuario aleatorio al iniciar la partida.

Problema: Pruebas.

Solución: Para las pruebas se ha empleado el framework Jasmine. La forma de trabajar con este framework es definiendo bloques “describe” en los que se define unas condiciones de inicio en las clausula “beforeEach” y se establecen los resultados esperados del comportamiento mediante la sentencia “expect”.

```
describe("El juego del impostor", function () {
    var juego;
    var usuario;

    beforeEach(function () {
        juego = new Juego();
        usuario = new Usuario("Pepe", juego);
    });

    it("inicialmente.....", function () {
        expect(Object.keys(juego.partidas).length).toEqual(0);
        expect(usuario.nick).toEqual("Pepe");
    });

    it("comprobar valores de la partida", function () {
        var codigo = juego.crearPartida(3, usuario);
        expect(codigo).toEqual("fallo");
        codigo = juego.crearPartida(11, usuario);
        expect(codigo).toEqual("fallo");
    });
```

Reunión al finalizar el Sprint

a) Revisión

Como incremento hemos obtenido el modelo lógico del juego y las pruebas de este.

b) Retrospectiva

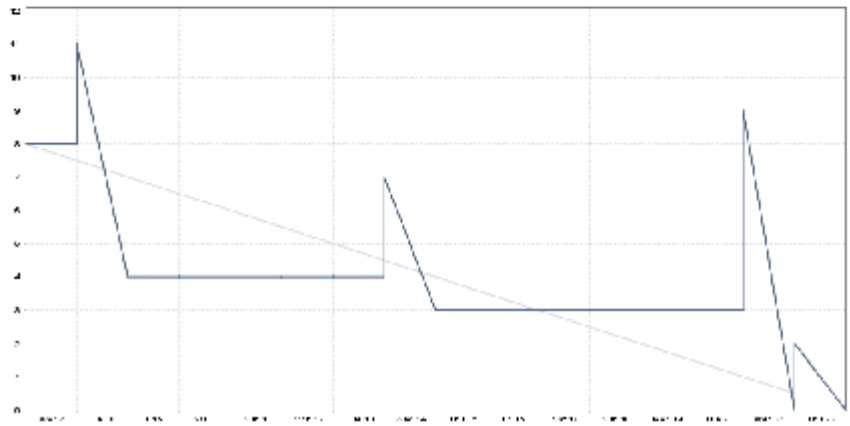
Se ha logrado en este Sprint realizar todas las tareas de las historias de usuario, consiguiendo así bajar completamente el Sprint Burndown el último día. Podemos ver como tanto el incremento de trabajo a realizar como el trabajo realizado se produce principalmente los días de clase.

Anexos:

Impostor-Juanma

Sprint Report, noviembre 3, 2020

Sprint	*spr2* Sprint Inicial
Period	mié, oct 07 - jue, oct 22 (15 days)
Velocity	26.5 StoryPoints
Burned work	22 hours
Product Owner	Juanma
Scrum Master	Juanma
Team	Juanma



Goal

Definir la logica inicial basica del juego.

Completed stories

sto1	Crear partida	2SP
Story description	Como usuario quiero crear partidas de modo que el sistema le asigna un código.	
Acceptance tests		
Closed tasks		
tsk1	Diseño inicial de la solucion	
tsk8	Pruebas	
tsk3	Tutoriales de HTML y JavaScript	
tsk4	Tutorial de jQuery	
tsk5	Implementar crear partida	

sto2	Unir a partida	1SP