

ATARION

Proyecto de fin de módulo. IES
Las Galletas, 2017

Índice de contenido

Cumplimiento de desarrollo.....3

Mecánica del juego.....3

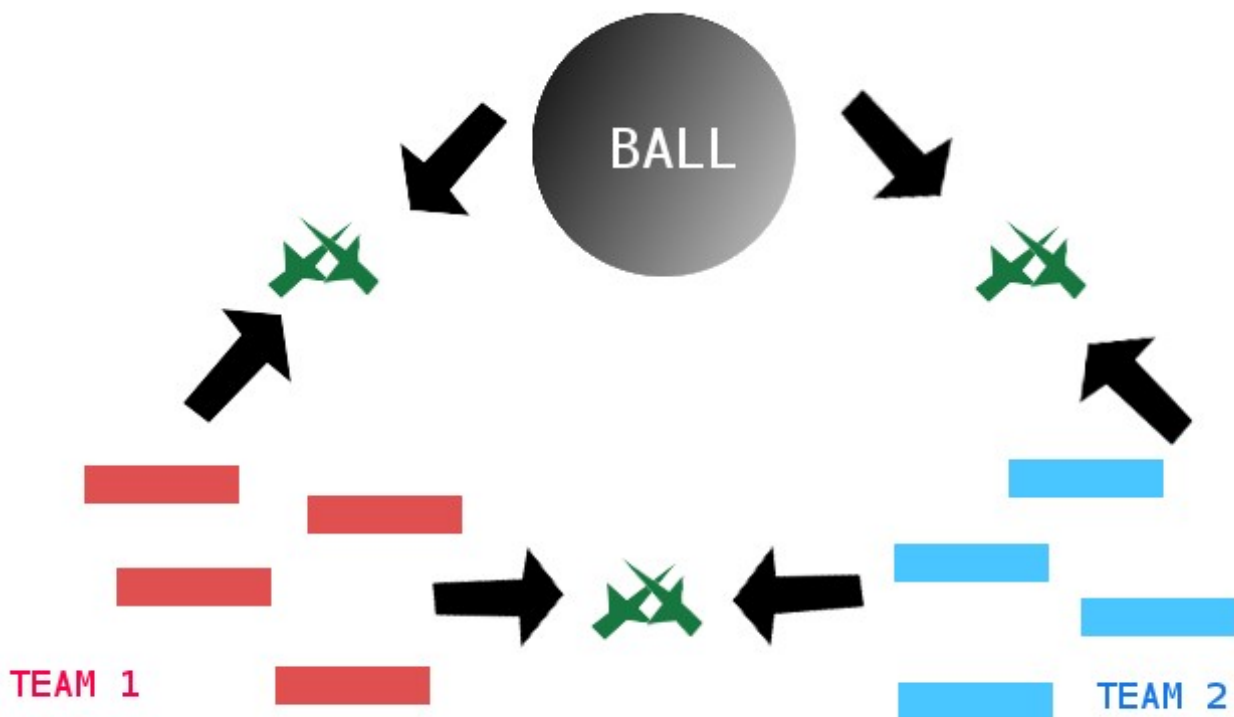
Estructura del proyecto.....5

Escenarios.....7

Cumplimiento de desarrollo

Desarrollo	Javier Godás Viéitez
Lenguaje	<ul style="list-style-type: none">• Java/ Testeado en Java 8.• LibGDX/ Desarrollado en libGDX.
Plataformas	<ul style="list-style-type: none">• Android/ Se le quitó prioridad y se hizo más hincapié en la mecánica.• PC/ Testeado en Ubuntu 14.04.
Compromisos mínimos de desarrollo	<ul style="list-style-type: none">• Dos tipos de bola/ Desarrollados tres tipos de bola y sus respectivos escenarios.• Dos tipos de barra/ Desarrollados tres tipos de barra y sus habilidades especiales.

Mecánica del juego.



Se trata de una mecánica basada en el arquetipo de “Todos contra el monstruo”, pero incluyendo conceptos del Atari Breakout e incluyendo mecánica multijugador, concretamente dos equipos compitiendo entre sí por ser los primeros en acabar con el “monstruo”.

Las reglas generales son:

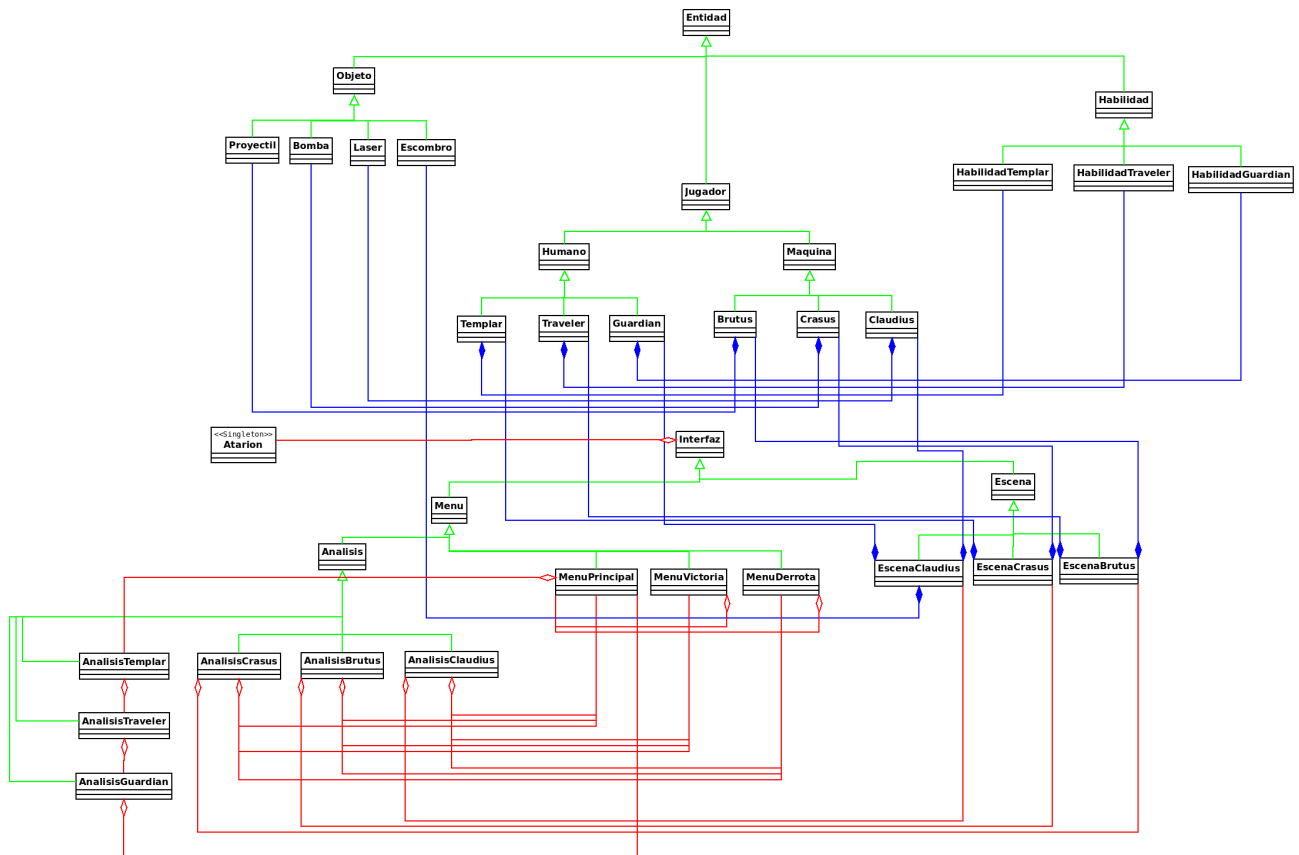
- El primer equipo que destruya la bola gana la partida y gana 2 puntos.
- Si un equipo destruye al otro, el equipo exterminador gana 1 punto y el equipo exterminado pierde 1 punto.
- Si ambos equipos mueren a manos de la bola, se considera empate y cada equipo pierde 2 puntos.

La idea es saber balancear la codicia con la prudencia, ya que hay que destruir la bola para ganar mientras se mantiene al otro equipo a raya. Un desequilibrio en éste balance genera uno de éstos dos escenarios:

1. El equipo se concentra en la bola, arriesgándose a que el equipo contrario espere a que se debiliten y exterminarlos, o a que al asestar el último golpe a la bola el equipo contrario se adelante/les frene y asesten ellos el golpe final, ganando.
2. El equipo se concentra en el otro equipo, arriesgándose a que la bola esté esperando a que los equipos se debiliten entre sí para exterminarlos, o esperar a que un equipo desaparezca para lidiar mejor con otro cuyos integrantes tengan habilidades que funcionen peor contra él.

Por razones de tiempo y recursos, se ha prescindido del modo multijugador para probar elementos básicos de la mecánica.

Estructura del proyecto



Se ha usado la herencia como principal herramienta para estructurar la aplicación, haciendo uso de relaciones de composición para desglosar clases grandes en clases más pequeñas y mantenibles, y de relaciones de agregación para navegar entre los elementos de la interfaz mediante la creación de instancias.

Atarion es la clase principal que gestiona todas las pantallas de la aplicación, ya que hereda de la clase principal Game, encargada de navegar entre distintos tipos de objetos Screen, clase de la cual hereda Interfaz.

Se ha escogido el patrón Singleton para permitir a instancias de las clases hijas de Interfaz obtener la instancia de Atarion y cambiar la pantalla a un nuevo objeto Interfaz sin necesidad de clases externas.

La clase Menú recoge los eventos de teclado, y navega entre los menús mediante la instancia de la clase Atarion.

La clase Escena es la interfaz que gestiona el nivel, determina las condiciones de victoria y derrota, crea las instancias de los jugadores y crea los objetos característicos del nivel.

La clase Análisis es meramente una introducción al nivel o una entrada del tutorial. Se tiene que decidir si integrar en la clase Menu o la clase Escena.

Todos los objetos de la clase Entidad heredan de la clase Rectangle de LibGDX, de manera que, obteniendo por el constructor la instancia SpriteBatch (clase encargada de dibujar objetos como Rectangle y actualizar sus estados) de la Escena correspondiente, pueda gestionar internamente su posición, sus colisiones y tamaño sin necesidad de codificarlo todo dentro de la Escena correspondiente.

La clase Jugador dispone de todos los atributos de los personajes (salud,fuerza,velocidad), y de una instancia del enemigo contra el que está luchando. Ésto no sólo es útil para que la AI pueda realizar decisiones en función de la posición y salud de los contrincantes, sino que también es útil para desarrollar habilidades especiales de jugadores humanos que interactúen con la AI, como por ejemplo la habilidad de la clase Guardian que desactiva la habilidad especial del jugador al que haga objetivo.

En el futuro, dado la verdadera naturaleza multijugador del juego, ésta instancia podría ser reemplazada por un array de instancias (ya que habría múltiples enemigos), además de otro array de jugadores aliados para soportar habilidades de bonus, como una habilidad de curación.

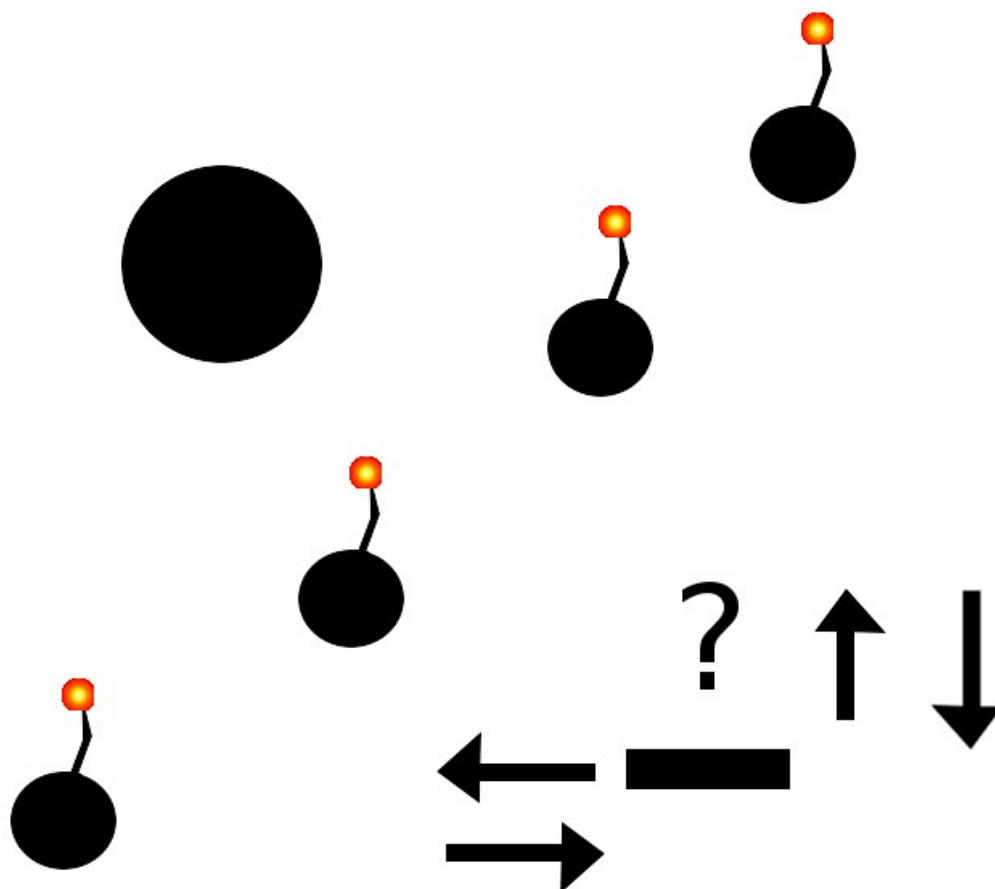
La clase Objeto, como su nombre indica, son componentes del entorno de cada Escena. Pueden ser de dos tipos; pertenecientes a un escenario, como la subclase Escombros pertenece a EscenaClaudius, simulando una estructura derrumbándose, y pertenecientes al jugador, como Brutus lanza instancias de la clase Proyectoil para tratar de alcanzar al jugador humano.

En el futuro pueden desmembrarse en dos ramas; Arma y Entorno, pero de momento por simplificar el esquema se ha hecho una relación de herencia más simple.

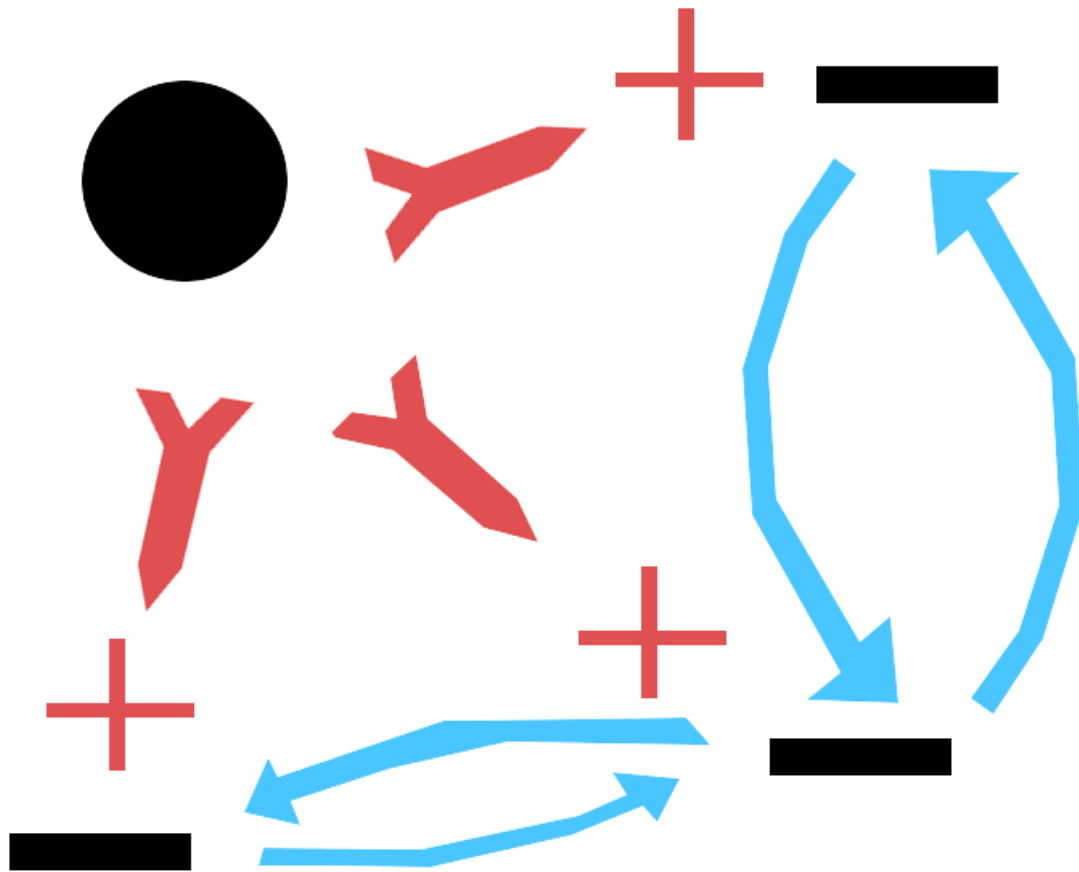
La clase Habilidad de momento únicamente contiene las animaciones de las habilidades especiales de cada barra, pero en el futuro se supone que va a gestionar toda lo relacionado con la habilidad de dicha barra.

Dada la simplicidad del código de LibGDX, no ha hecho falta la creación de utilidades.

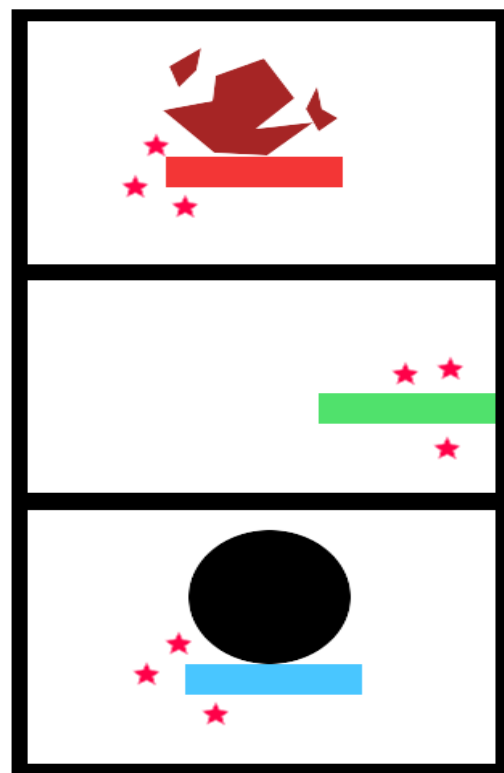
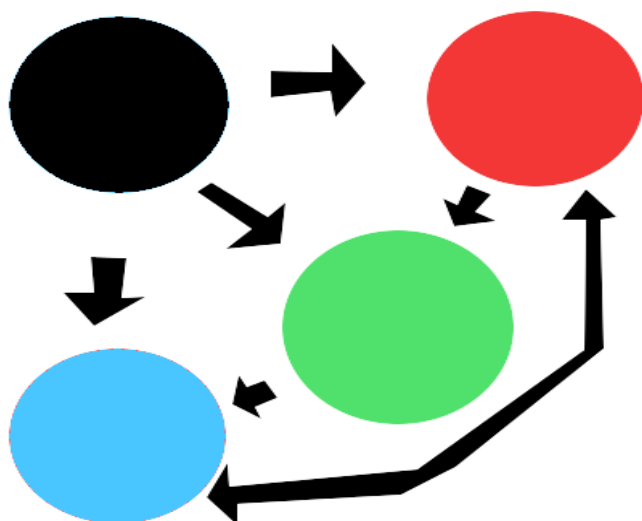
Escenarios



Nombre del escenario	Batalla de fuerza
Condición de derrota	<ul style="list-style-type: none">• El jugador humano es alcanzado por la explosión de las bombas una sola vez.
Condición de victoria	<ul style="list-style-type: none">• La bola es empujada contra los bordes y sus vidas se reducen a 0.
Descripción	Se trata de un escenario en el que una barra clase Templar se enfrenta a una bola clase Crasus. Crasus tratará de escapar del jugador humano mientras carga su habilidad especial, con la que pondrá bombas que explotarán en tres segundos mientras los controles de la barra se invierten, tratando de que el jugador de un mal paso y la explosión le atrape. El jugador tendrá que empujar la bola a los bordes para dañarla y usar su habilidad especial para escapar de las bombas.



Nombre del escenario	Batalla de maña
Condición de derrota	<ul style="list-style-type: none"> • El jugador humano recibe daño de la bola y sus vidas quedan reducidas a 0. • El jugador recibe daño del proyectil y sus vidas quedan reducidas a 0.
Condición de victoria	<ul style="list-style-type: none"> • La bola recibe daño de su propio proyectil y sus vidas se reducen a 0.
Descripción	<p>Éste escenario enfrenta a una barra clase Traveler contra una bola clase Brutus. La bola clase Brutus es muy agresiva y perseguirá al jugador en todo momento, rebotando contra los bordes de la escena para tratar de confundir al jugador sobre la trayectoria que tomará. Mientras que la bola es muy fuerte, el Traveler es débil. Cada cierto tiempo, Brutus usará su habilidad especial, lanzando un proyectil mientras la velocidad del jugador se ve reducida. Explota la superior inteligencia del Traveler para que una bola virtualmente invencible se venza a sí misma.</p>



Nombre del escenario	Batalla de resistencia
Condición de derrota	<ul style="list-style-type: none"> • El jugador humano hace contacto con la bola cuando está azul. • El jugador humano hace contacto con los bordes de la escena cuando la bola está verde. • El jugador hace contacto con los escombros cuando la bola esté roja. • El jugador hace contacto con la bola. • El jugador recibe daño del láser.
Condición de victoria	<ul style="list-style-type: none"> • La bola pierde su inmunidad a los escombros y es golpeada por ellos hasta que sus vidas lleguen a 0.
Descripción	<p>Aquí se ven las caras los tipos de jugador más resistentes; la Guardian, con doble salud y capaz de atravesar objetos, y Crasus, inmune a daños y que, a diferencia de los demás tipos, tratará de estar lo más quieta y firme posible, sólo se moverá para esquivar escombros y cargar con el láser. Durante la carga, la bola pierde su resistencia a los escombros que caen, así que la Guardian deberá estar esperando con su módulo a que un escombros esté en la trayectoria de su carga para apagar su inmunidad y hacerle morder el polvo.</p>