

Resumen - Tarea 3

Profesora: Nancy Hitchfeld K.

Alumno: Nicolás García Ríos

Rut Alumno: 20.394.672-4

5 de enero de 2021

Introducción

El diseño implementado para el proyecto fué modificado tal como se mostrará a continuación. Cada paquete vendrá con una breve descripción y su diagrama correspondiente, mientras que los diagramas generales y más detallados (que además incluyen los constructores) estarán en el repositorio.

El Link del repositorio es el siguiente:

<https://github.com/CC3002-Metodologias/final-reality-Nicolas-Francisco/tree/master>

Desiciones de Diseño

El proyecto completo se encuentra en el paquete FINAL REALITY, dividido entre los paquetes MODEL y GUI (que solo contiene la gui del juego). Dentro del paquete MODEL se encuentran todas las clases trabajadas de esta tarea, donde se incluye el paquete CHARACTER, el paquete CONTROLLER y el paquete WEAPON que se detallarán más adelante.

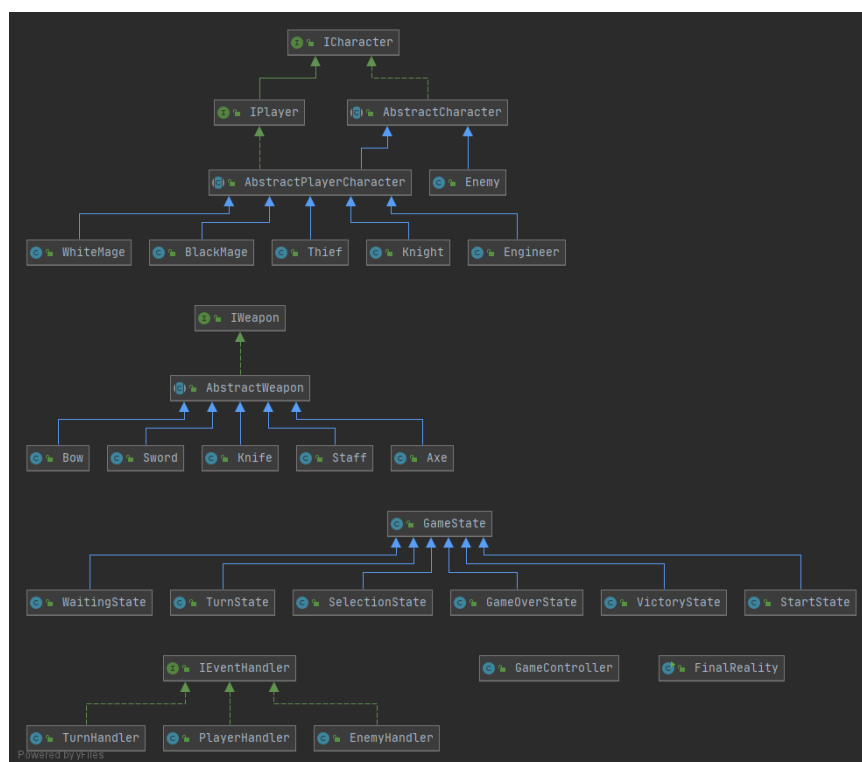


Figura 1: Diagrama del paquete FINAL REALITY

Dentro del paquete CHARACTER se encuentran todos los personajes manejados por la máquina y el jugador. La clase abstracta AbstractCharacter implementa la Interface ICharacter, mientras que el Enemy y el paquete PLAYER heredan de esta clase abstracta.

AbstractCharacter contiene los atributos de nombre, hp (puntos de vida) y defensa, además del BlockingQueue que representa la cola para la batalla entre los personajes.

Enemy hereda directamente de la clase abstracta mencionada ya que no tiene más métodos en común con los jugadores. Dentro de esta clase se agregan los atributos de Peso y Defensa del enemigo, y además los métodos waitTurn() (del enemigo) y de hashCode/Equals que irán siempre juntos y no pueden ser heredables.

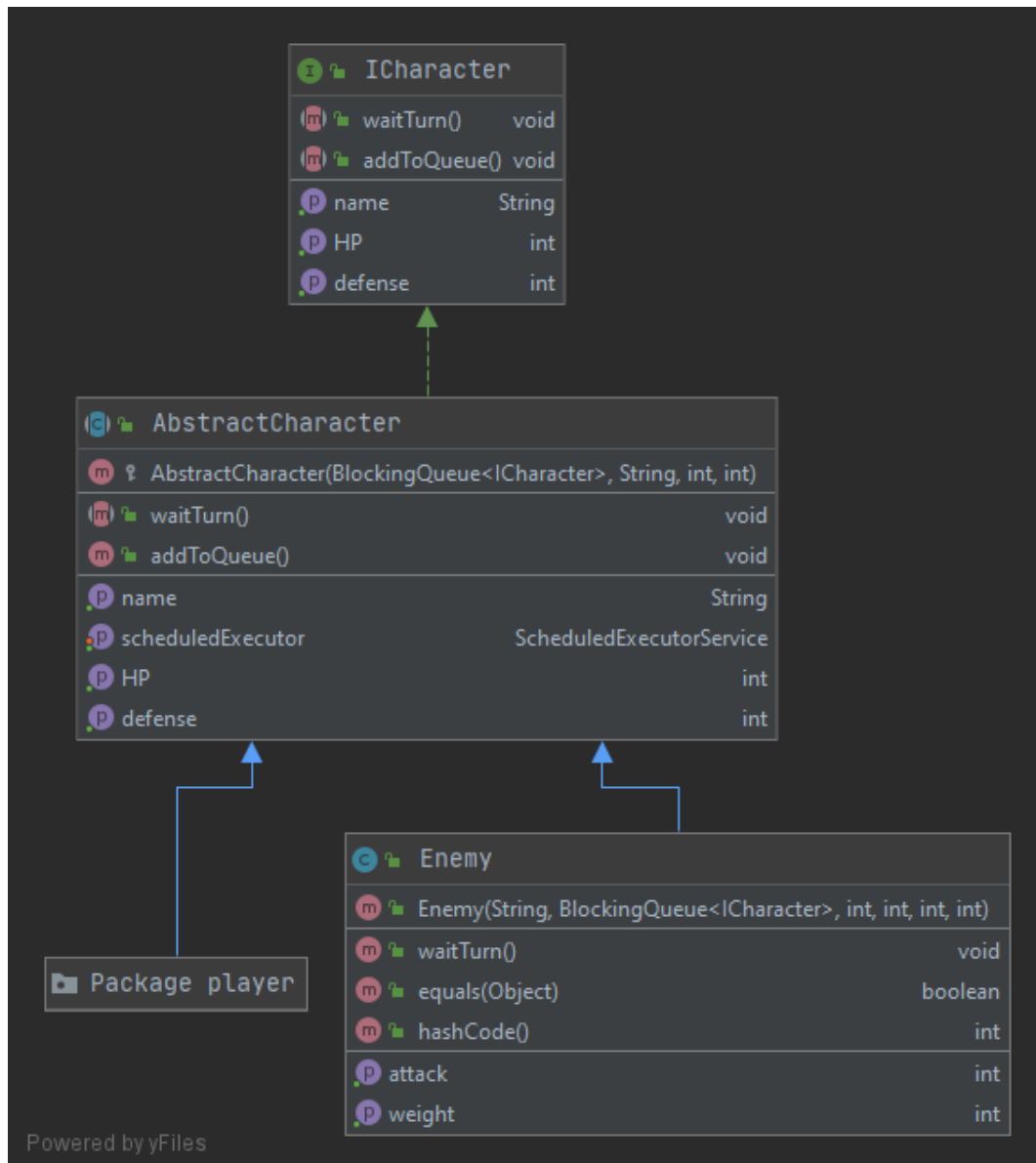


Figura 2: Diagrama del paquete CHARACTER

Se modifica la anterior clase PlayerCharacter (la cuál uno de sus atributos era la clase de los personajes) para crear la Clase Abstracta AbstractPlayerCharacter, de la cuál heredan todas las clases de personajes. Esta clase abstracta representa cualquier personaje controlado por el jugador, e incluye los métodos equip (que solo pueden usar los jugadores) y el waitTurn (de los jugadores).

A partir de esta clase Abstracta hereda cada clase o tipo de jugador, donde cada uno contiene su propio Hashcode/Equals y en el caso de las clases mágicas el atributo MP (maná) y el método getMP. Tener las clases separadas de esta forma ayudará para el Double Dispatch que se hará en el futuro con las armas.

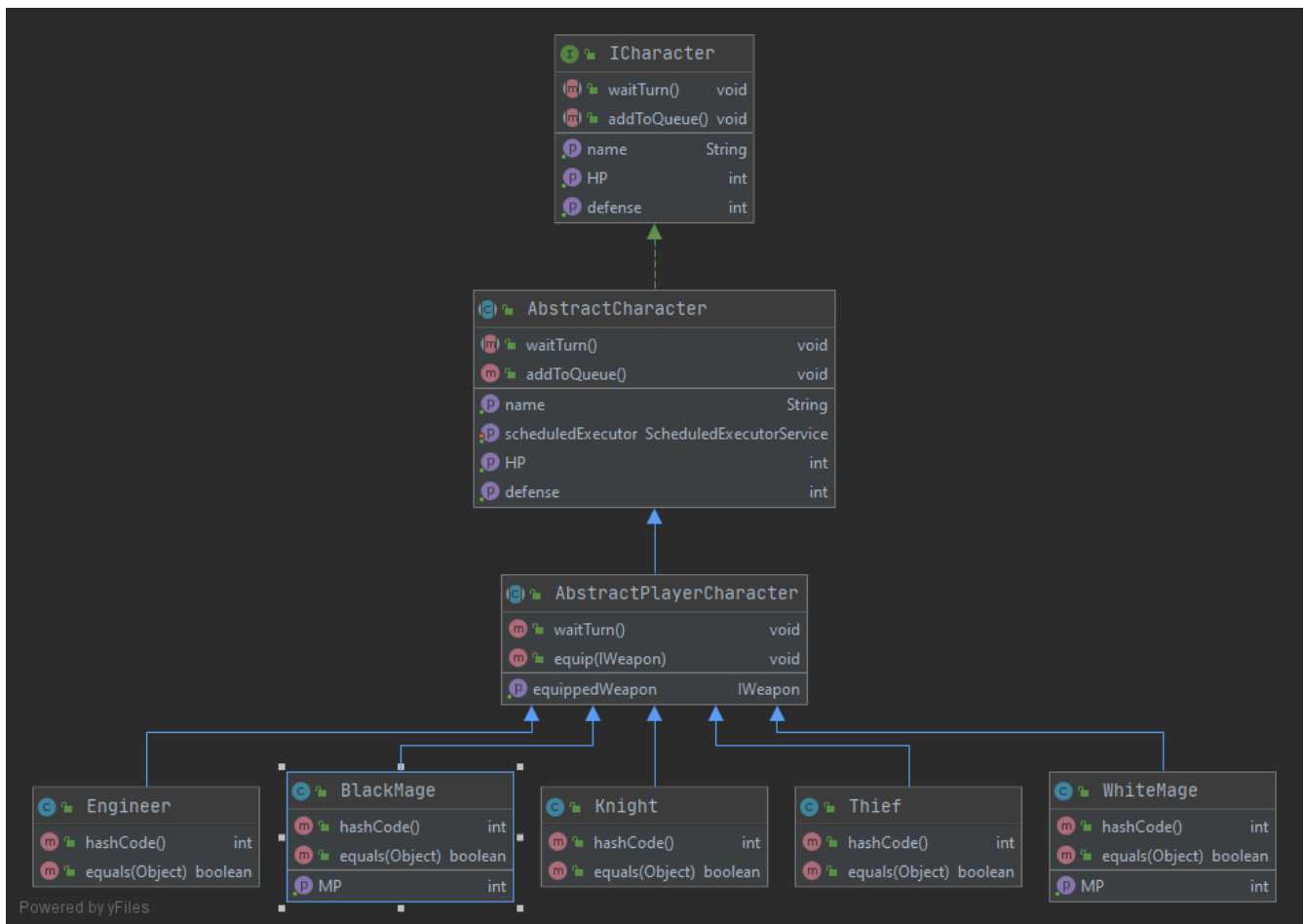


Figura 3: Diagrama del paquete PLAYER

En el paquete WEAPON tenemos las armas, las cuales para ser todas unificadas se crea la interface IWeapon y la clase abstracta AbstractWeapon. Cada tipo de arma hereda de esta clase abstracta y al igual que con las clases de jugadores, separamos las armas para facilitar en un futuro el Double Dispatch.

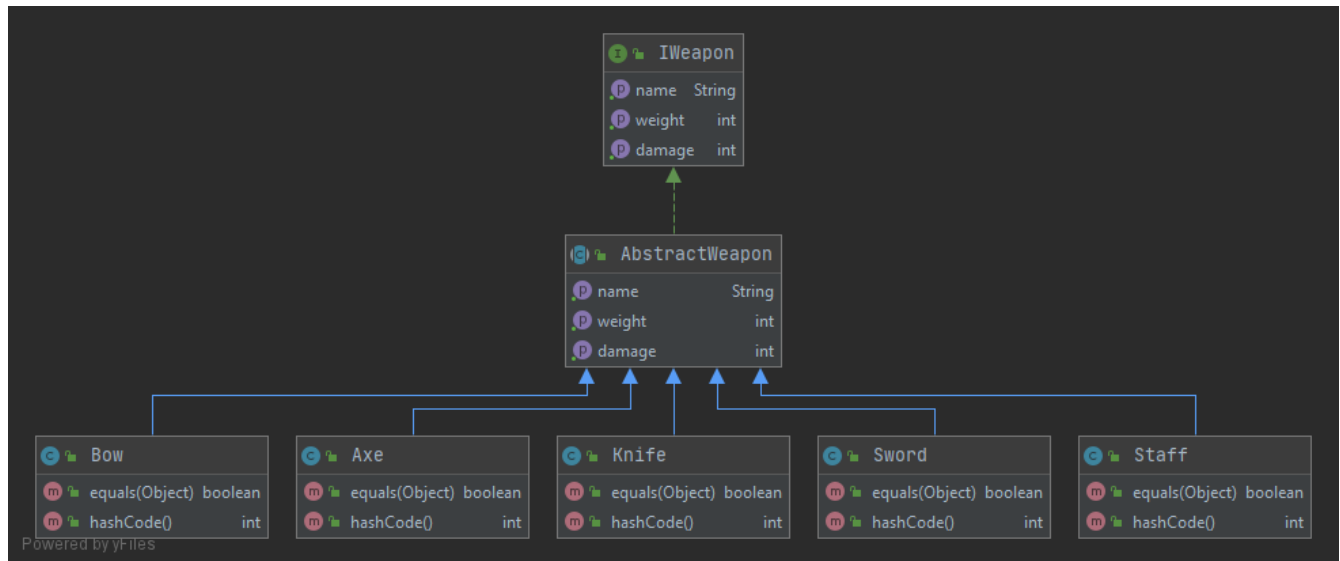


Figura 4: Diagrama del paquete WEAPON

En el paquete controller tenemos el controller del juego, los handlers para el observer de ambos tipos de personajes (IPlayer y Enemy), la interface que engloba a estos dos, y los estados del juego. Dentro del controller se encuentra gran parte del código del juego, ya que utiliza todos los métodos de los paquetes mostrados anteriormente además de iniciadores y finalizadores del juego. Basta agregar que el controller también tiene el inventario del jugador.

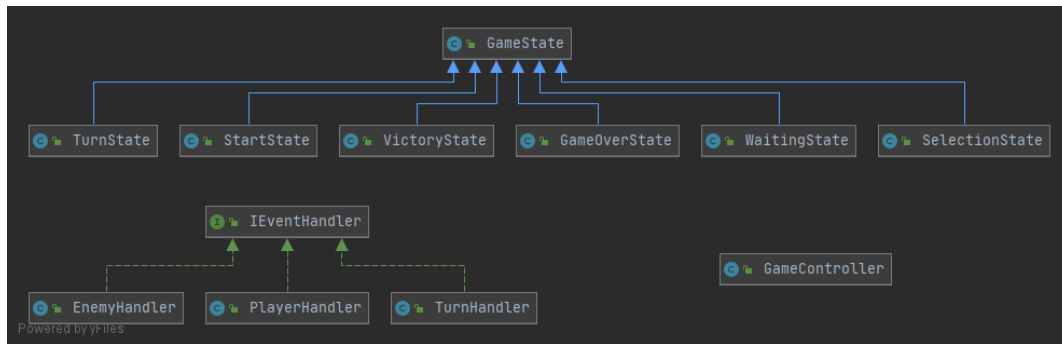


Figura 5: Diagrama del paquete CONTROLLER

Los diagramas completos se encuentran en el repositorio github dejado al principio del documento, donde se detallan tanto los métodos como los constructores.