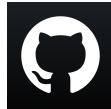


Grupo: David Marin Yepes, Rey Valentin Arias & Ana María Díaz.

1. GITHUB

- Diagrama de clases
- Implementación inicial



2. Descripción de clases:

- Descripción de las clases, interfaces, etc, que participarán de su solución.
- En dicha descripción estén definidos los atributos y métodos de cada clase, las firmas de interfaces y demás elementos necesarios

Texto, jugabilidad:

El héroe debe salvar a la doncella del último piso de la torre, para esto debe vencer cada obstáculo por piso, esto es, otros personajes con una puntuación en específico. Para derrotarlos el héroe debe ir a su piso y garantizar que tiene un puntaje mayor al de dicho personaje (Si va a su piso con un menor puntaje, el héroe pierde). Sobre el puntaje: los obstáculos lo tienen por preestablecido mientras que el héroe lo adquiere por cada nueva victoria (Y es acumulativo). Más aún, por cada movimiento del héroe algo más sucede simultáneamente en su entorno (Ejemplo, otra torre emerge o aparecen más collectibles). El propósito de este videojuego de plataforma es que el jugador debe ser atento con los puntajes y estratégico en sus movimientos, y por nivel deberá salvar múltiples doncellas en distintas torres. // *Palabra clave:* Merge Puzzle. // *Key visual:* [click](#).

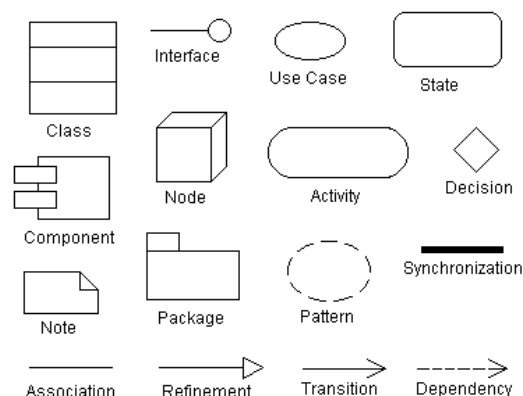


Clases: *Atributos: (Valores que almacenan) & Métodos: (Operaciones posibles)*

- **Personaje** (héroe): Sus atributos serán su actual posición y su número de puntos, según esto se determinará (método) si sigue en la partida o si muere (derrotado por algún obstáculo).
- **Jugador** (usuario): Este tiene dos posibles operaciones, la primera es moverse (Cambiando así la posición actual del héroe) y la segunda es atacar. Para atacar el jugador debe desplazar al héroe hasta algún piso donde esté un obstáculo y aquí el juego observará si vence (Tiene mayor puntaje que la del obstáculo) o si pierde (El obstáculo tiene más puntos que el héroe)
- **Obstáculos:**
 - **Enemigos** (personajes por piso): El enemigo responde al ataque del héroe y proporcionado por el jugador. Si el obstáculo gana en puntaje, el héroe muere; pero si el pierde en puntaje, el obstáculo y su piso desaparece (Haciendo que el héroe esté un paso más cerca de salvar a la doncella y con esto, de ganar todo el juego)
 - **Aliados** (coleccionables): Conforme avanza el nivel, en el juego podrán aparecer power ups u otros peligros para ayudar o impedir, respectivamente, que el héroe logre la gran hazaña.
- **Piso:** Su atributo responde a si tiene un personaje en aquel piso o no, y su método responde a si sigue en la torre (Y desaparece al héroe) o si desaparece el piso y el obstáculo (Pero no el héroe, el asciende)
- **Torre:** El número de pisos cambiará conforme el jugador juegue (Valga la redundancia). Por ende, su atributo será cuántos pisos tiene la torre en un principio, y su operación será removerlos secuencialmente.

Interfaces: IFighter

Relaciones: Guia:



- Dependencias: Un objeto usa a otro (En una sola dirección)
 - El enemigo y el jugador dependen de la interface
 - Asociaciones: Objetos de una clase están conectados a objetos de otra (Bidireccional)
 - (Tipo: agregación) El personaje y el piso
 - (Tipo: composición) El personaje y la torre
 - Organizaciones: Objetos diferentes tienen responsabilidades o propiedades en común
 - Generalización: Relación entre una clase más general y una más específica o especializada
 - El enemigo y el aliado a obstáculos
 - Obstáculos a personaje
 - Jugador a personaje
-