

Atributos

Space Invaders

Realizado por:

Nacho Díez García

1ºDAW

Todos los atributos de nuestro juego:

Desde un primer instante hay que añadir que el main tiene todos los objetos del juego, después nos encontramos con el modelo que yo debo de explicar de esto los atributos existentes de este.

Atributos, según las clases:

Todos los atributos son **private**, por si cabía alguna duda, empecemos a dictar los atributos de cada clase.

1. Clase Game

Los atributos serán:

nombre: String

“El videojuego por lo normal se llamará Space Invaders”

puntaje: int

“Naves enemigas detonadas”

estado: String

“Ejemplo: "En curso", "Finalizado"”

2. Clase Tablero

Los atributos serán:

ancho: int

“Para el ancho del nuestro juego, limite de ancho”

alto: int

“Para el alto que acota también las dimensiones de nuestro juego”

fondoInteractivo:Texture

“Es para tener un fondo que se mueva de derecha a izquierda”

3. Clase (Herencia) Nave

Los atributos serán:

ancho: int

“Ancho de las naves”

alto: int

“Alto de las naves”



velX: int

“Velocidad en la posición X de las naves que van a ser las mismas en todas”

velY: int

“Velocidad en la posición Y de las naves que van a ser las mismas en todas”

enum: Direccion (ARRIBA, ABAJO, IZQ, DER)

“Necesario para mover nuestra nave y para donde van a ir las enemigas”

4. Clase Batallon

Los atributos serán:

idBatallon: int

“Identificador de batallón”

escuadrones: List<Escuadron>

“Lista de escuadrones que hay en un batallón”

EscuadronesMax: int

“Número máximo de escuadrones”

5. Clase Escuadron

Los atributos serán:

idEscuadron: int

“Identificador de escuadrón”

navesEnemigas: List<NaveEnemiga>

“Todas las naves enemigas que hay dentro de un escuadrón”

navesEnemigasmax: int

“Maximo de naves enemigas en un escuadrón”

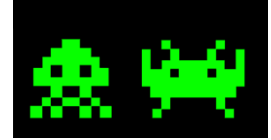
6. Clase NaveEnemiga (Hereda de Nave)

Los atributos serán:

Texture[]: navesEnemigas

“Dado que puede ser la nave enemiga con texturas diferentes en nuestro caso el pulpo y el otro que no se que forma tiene a ciencia cierta, tal vez como un conejo”

“Y dado que para las animaciones de dichas imágenes debemos utilizar dicho array, aunque podríamos hacer uso de dos distintas, vamos a utilizar únicamente una”



y: int

“Posición y”

x:int

“Posición x”

estaVivo: boolean

“Lo que dice literalmente si están o no están las naves si están vivas o muertas, para después poder hacer una serie de acciones según esto”

7. Clase NaveAmiga (Hereda de Nave)

Los atributos serán:

Texture: naveAmiga

“Dado que solo hay una nave amiga”

y: int

“Posición y”

x:int

“Posición x”

estaVivo: boolean

“Lo que dice literalmente si están o no están las naves si están vivas o muertas, para después poder hacer una serie de acciones según esto”

vidas: int

“Las vidas de las que dispone una nave amiga que se restarán dependiendo de estarViva”

8. Clase Disparos (Hereda de Entidad)

Los atributos serán:

Texture[]: disparo

“Con dos posiciones una para el amigo y otro para el enemigo”

ancho: int

“Ancho de los disparos”

alto: int

“Alto de los disparos”

velX: int

“Velocidad en la posición X de todos los disparos”

velY: int

“Velocidad en la posición Y de todos los disparos”

9. Clase DA (Disparos Amigos)

Los atributos serán:

y: int

“Posición y”

x:int

“Posición x”

disparoAmigo: Texture;

“Foto de la nave amiga, si no gusta la implementación realizada con anterioridad”

10. Clase DE (Disparo Enemigo)

Los atributos serán:

y: int

“Posición y”

x:int

“Posición x”

disparoEnemigo: Texture;

“Foto de la nave amiga, si no gusta la implementación realizada con anterioridad”