

Code Legends

Christian Villanueva Paez (Mici7120)

Johan Stevens Charria (Natchmahr)

Jeferson Danilo Arevalo (winandtu)

Versión 1.0

Santiago de Cali
25 de Mayo de 2020

Escuela de Ingeniería de Sistemas y Computación
Ingeniería de Sistemas
Universidad del Valle

Tabla de contenidos

Table of contents

- README
- Índice de clases
 - Lista de clases
- Índice de archivos
 - Lista de Archivos
- Documentacion de las clases
- Documentacion de los Archivos

README

#Code Legends

Code Legends es un juego de estrategia tipo competitivo, basado en la arquitectura monolítica P2P (Peer to Peer) en el que se fusiona la velocidad y la estrategia, con elementos de juegos de rol. En el juego aparece un ejército de batalla conformado por una cantidad de unidades que se especifica al comienzo de la partida entre 3 tipos de avatar, cada uno con un diseño y estilo único, que deberán luchar contra otro ejército. La dinámica del juego consiste en que un ejército trata de destruir la torreta y el otro ejército trata de protegerla. ¡Por defender la Torreta, en batalla se perderán muchas vidas!

#Contenido

La configuración usada para compilar el código fue la siguiente:

```
g++ -o main main.cpp avatar.cpp box.cpp ejercito.cpp tablero.cpp menu.cpp
```

#Ejecutar

También se puede ejecutar a través de

#Vista Previa

Menu del juego:

Configuración de ejército:

Mapa de batalla:

#Notas

Equipo:

Christian Villanueva Paez (Mici7120)

Johan Stevens Charria (Natchmahr)

Jeferson Danilo Arevalo (winandtu)

Índice de clases

Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

Avatar	5
Box	7
Ejercito	10
JuegoBase	12
Menu	14
Tablero	15

Indice de archivos

Lista de archivos

Lista de todos los archivos con descripciones breves:

avatar.cpp	17
avatar.h	18
box.cpp	19
box.h	20
ejercito.cpp	21
ejercito.h	22
juegoBase.cpp	24
juegoBase.h	25
main.cpp	26
menu.cpp	27
menu.h	28
tablero.cpp	32
tablero.h	33

Documentación de las clases

Referencia de la Clase Avatar

```
#include <avatar.h>
```

Métodos públicos

- **Avatar** (string _tipoAvatar)
- void **setVida** (float _Vida)
- void **restarVida** (float _DaNo)
- string **getTipo** ()
- float **getPoder** ()
- float **getVida** ()
- float **getMana** ()

Atributos protegidos

- string **tipoAvatar**
- float **Poder**
- float **Mana**
- float **Vida**

Documentación del constructor y destructor

Avatar::Avatar (string _*tipoAvatar*)

Dara las características iniciales dependiendo al tipo de avatar

Documentación de las funciones miembro

float Avatar::getMana ()

Para obtener el mana del avatar

float Avatar::getPoder ()

Para obtener el poder del avatar

string Avatar::getTipo ()

Para obtener el tipo de avatar

float Avatar::getVida ()

Para obtener la vida actual del avatar

void Avatar::restarVida (float _*DaNo*)

Resta el daño recibido a la vida del avatar

void Avatar::setVida (float _*Vida*)

Modifca la vida del avatar

Documentación de los datos miembro

float Avatar::Mana[protected]

Mana del **Avatar** (Defensa)

float Avatar::Poder[protected]

Poder del **Avatar** (Ataque)

string Avatar::tipoAvatar[protected]

Tipo de **Avatar** (luchador, tirador o mago)

float Avatar::Vida[protected]

Vida

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- avatar.h
- avatar.cpp

Referencia de la Clase Box

```
#include <box.h>
```

Métodos públicos

- **Box ()**
- void **setID** (int id)
- int **getID** ()
- int **getVidaTorreta** ()
- void **setVidaTorreta** (int _vidaTorreta)
- void **setTorreta** (int _valoresTorreta[5])
- void **setTorretaInicial** ()
- void **daNoTorreta** ()
- void **setEjercitoInicial** (int numeroLuchadores, int numeroTiradores, int numeroMago)
- void **setEjercito** (int numeroLuchadores, int numeroTiradores, int numeroMago, vector< float > vidas)
- void **movimientoEjercito** (Box boxMovido)
- bool **Derrotado** ()
- void **informacionEjercito** ()

Atributos públicos

- int **ID**
- bool **Torreta**
- bool **Norte**
- bool **Sur**
- bool **Este**
- bool **Oeste**
- **Ejercito** ejercito

Atributos protegidos

- int **vidaTorreta**

Documentación del constructor y destructor

Box::Box ()

Documentación de las funciones miembro

void Box::daNoTorreta ()

Daño a la torreta

bool Box::Derrotado ()

Especifica si el ejercito ha sido derrotado

int Box::getID ()

int Box::getVidaTorreta ()

Retorna la vida actual de la torreta

void Box::informacionEjercito ()

Imprime las estadísticas del ejército

void Box::movimientoEjercito (Box *boxMovido*)

Mueve el ejército a otro box

void Box::setEjercito (int *numeroLuchadores*, int *numeroTiradores*, int *numeroMago*, vector< float > *vidas*)

Crea el ejército con los respectivos valores de vida de cada avatar

void Box::setEjercitoInicial (int *numeroLuchadores*, int *numeroTiradores*, int *numeroMago*)

Introduce el número de unidades de cada tipo de ejército

void Box::setID (int *id*)

void Box::setTorreta (int *_valoresTorreta*[5])

Configura el box como una torreta, con los valores cargados

void Box::setTorretaInicial ()

Inicializa los valores predeterminados a la torreta

void Box::setVidaTorreta (int *_vidaTorreta*)

Modifica la vida de la torreta

Documentación de los datos miembro

Ejercito Box::ejercito

bool Box::Este

int Box::ID

Identificación de la casilla que indica si es ejército (1 o 2) o torreta (9)

bool Box::Norte

bool Box::Oeste

Posición de los escudos de la torreta

bool Box::Sur

bool Box::Torreta

Posición de la torreta

int Box::vidaTorreta[protected]

Puntos de golpe de la torreta (vida)

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- `box.h`
- `box.cpp`

Referencia de la Clase Ejercito

```
#include <ejercito.h>
```

Métodos públicos

- **Ejercito ()**
- **int getCantidadLuchadores ()**
- **int getCantidadTiradores ()**
- **int getCantidadMagos ()**
- **int soldadosVivos ()**
- **bool derrotado ()**
- **void operator/ (Ejercito atacado)**
- **void restarVida (float vidaPerdida)**
- **float sumaPoder ()**
- **float sumaMana ()**
- **void setEjercitoInicial (int luchadores, int tiradores, int magos)**
- **void setEjercito (int luchadores, int tiradores, int magos, vector< float > vidas)**
- **void movimientoEjercito (Ejercito ejercitoMovido)**
- **float obtenerPeorVida (vector< float > vidas)**
- **void restaurarVida (float peorVida)**

Atributos públicos

- **vector< Avatar * > ejercitoAvatar**
- **int Luchadores**
- **int Tiradores**
- **int Magos**
- **int cantidadEjercito**

Documentación del constructor y destructor

Ejercito::Ejercito ()

Documentación de las funciones miembro

bool Ejercito::derrotado ()

Indica si el ejercito ha sido destruido

int Ejercito::getCantidadLuchadores ()

Esta implementacion nos dara la cantidad total de luchadores vivos

int Ejercito::getCantidadMagos ()

Esta implementacion nos dara la cantidad total de magos vivos

int Ejercito::getCantidadTiradores ()

Esta implementacion nos dara la cantidad total de tiradores vivos

void Ejercito::movimientoEjercito (Ejercito *ejercitoMovido*)

Mueve los avatares del vector de un ejercito a otro

float Ejercito::obtenerPeorVida (vector< float > *vidas*)

Ordenara todas las vidas de menor a mayor y obtendra la peor vida

void Ejercito::operator/ (Ejercito *ejercitoAtacado*)

Recibe como parametro al otro ejercito y hace el combate

void Ejercito::restarVida (float *vidaPerdida*)

Resta vida al avatar de menor poder

void Ejercito::restaurarVida (float *peorVida*)

Restaura al avatar con la vida más baja

void Ejercito::setEjercito (int *luchadores*, int *tiradores*, int *magos*, vector< float > *vidas*)

Inicializa el ejercito con la cantidad correspondiente de avatares y su respectiva vida

void Ejercito::setEjercitoInicial (int *luchadores*, int *tiradores*, int *magos*)

Inicializa el ejercito con la cantidad correspondiente de avatares

int Ejercito::soldadosVivos ()

Indica el numero de soldados vivos

float Ejercito::sumaMana ()

Suma los puntos de mana del ejercito

float Ejercito::sumaPoder ()

Suma los puntos de poder del ejercito

Documentación de los datos miembro

int Ejercito::cantidadEjercito

Numero de unidades dentro del ejercito

vector<Avatar*> Ejercito::ejercitoAvatar

Es un vector de punteros de clase **Avatar**

int Ejercito::Luchadores

int Ejercito::Magos

int Ejercito::Tiradores

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- `ejercito.h`
- `ejercito.cpp`

Referencia de la Clase JuegoBase

```
#include <juegoBase.h>
```

Métodos públicos

- **JuegoBase** ()
- void **Main** (string configuracionPartida)
- void **configurarNuevaPartida** ()
- void **configurarCargarPartida** ()
- void **sorteoTurno** ()
- void **Movimiento** (int _Ejercito, int &coordeX, int &coordeY, int &opcion)

Atributos públicos

- int **Turno**
- Tablero **tableroDeJuego**

Documentación del constructor y destructor

JuegoBase::JuegoBase ()

Documentación de las funciones miembro

void JuegoBase::configurarCargarPartida ()

Carga una partida guardada

void JuegoBase::configurarNuevaPartida ()

Inicia una nueva partida

void JuegoBase::Main (string *configuracionPartida*)

Administra los metodos de los movimientos de los ejercitos y determina cuando se acaba la partida, o por lo contrario se guarda la partida

void JuegoBase::Movimiento (int *_Ejercito*, int & *coordeX*, int & *coordeY*, int & *opcion*)

Compara la dirección hacia donde se va a mover cada ejercito y ejecuta su acción correspondiente de acuerdo a lo que encuentre, ya sea ataque movimiento normal o bloqueo

void JuegoBase::sorteoTurno ()

Elige al azar el jugador que hará la primera jugada en la partida

Documentación de los datos miembro

Tablero JuegoBase::tableroDeJuego

int JuegoBase::Turno

Turno actual de la partida

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- juegoBase.h
- juegoBase.cpp

Referencia de la Clase Menu

```
#include <menu.h>
```

Métodos públicos

- `Menu ()`
- `void MenuPrincipal ()`
- `void Instrucciones ()`

Atributos públicos

- `JuegoBase Partida`
-

Documentación del constructor y destructor

`Menu::Menu ()`

Documentación de las funciones miembro

`void Menu::Instrucciones ()`

Carga las instrucciones del juego del archivo "Instrucciones.txt"

`void Menu::MenuPrincipal ()`

Ejecuta una serie de metodos dependiendo de la opción que se elija

Documentación de los datos miembro

`JuegoBase Menu::Partida`

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- `menu.h`
- `menu.cpp`

Referencia de la Clase Tablero

```
#include <tablero.h>
```

Métodos públicos

- **Tablero ()**
- **~Tablero ()**
- **void setMatrizTablero ()**
- **void nuevaPartida ()**
- **void cargarPartida ()**
- **void guardarPartida (int _Turno)**
- **void imprimirTablero ()**
- **void setCoordenadasEjercito (int _Ejercito, int X, int Y)**

Atributos públicos

- **Box ** matrizTablero**
- **int turnoInicial**
- **int Ejercito1X**
- **int Ejercito1Y**
- **int Ejercito2X**
- **int Ejercito2Y**
- **int TorretaX**
- **int TorretaY**

Documentación del constructor y destructor

Tablero::Tablero ()

Tablero::~~Tablero ()

Documentación de las funciones miembro

void Tablero::cargarPartida ()

Lee el archivo y guarda su determinada información en cada uno de los boxes

void Tablero::guardarPartida (int _Turno)

Guarda el estado de la partida actual en el archivo "partidaGuardada"

void Tablero::imprimirTablero ()

Recorre la matriz del tablero e imprime sus valores, "1" para ejercito 1, "2" para ejercito 2 y "9" para la torreta

void Tablero::nuevaPartida ()

Carga los valores iniciales para una nueva partida

void Tablero::setCoordenadasEjercito (int _Ejercito, int coordenadasX, int coordenadasY)

Guarda las coordenadas del ejercito

void Tablero::setMatrizTablero ()

Crea los box del tablero

Documentación de los datos miembro

int Tablero::Ejercito1X

int Tablero::Ejercito1Y

int Tablero::Ejercito2X

int Tablero::Ejercito2Y

Box Tablero::matrizTablero**

int Tablero::TorretaX

int Tablero::TorretaY

Posición de la torreta en la matriz

int Tablero::turnoInicial

Turno del Jugador al iniciar la partida

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- **tablero.h**
- **tablero.cpp**

Documentación de archivos

Referencia del Archivo avatar.cpp

```
#include "avatar.h"
```

Referencia del Archivo avatar.h

```
#include <string>
#include <iostream>
#include <fstream>
#include <sstream>
#include <stdlib.h>
#include <ctime>
```

Clases

- class **Avatar**

Referencia del Archivo box.cpp

```
#include "box.h"
```

Referencia del Archivo box.h

```
#include "ejercito.h"
```

Clases

- class **Box**

Referencia del Archivo ejercito.cpp

```
#include "ejercito.h"
```


Referencia del Archivo ejercito.h

```
#include "avatar.h"  
#include <vector>
```

Clases

- class **Ejercito**

defines

- #define **EJERCITO_H**
-

Documentación de los 'defines'

```
#define EJERCITO_H
```

Referencia del Archivo Instrucciones.txt

Variables

- Instrucciones __pad0__
- Instrucciones con elementos de juegos de rol En el juego aparece un ejército de batalla cada uno con un diseño y estilo **único**
- Instrucciones con elementos de juegos de rol En el juego aparece un ejército de batalla cada uno con un diseño y estilo que deberán luchar contra otro ejército La dinámica del juego consiste en que un ejército trata de destruir la Torre y el otro ejército trata de protegerla ¡Por defender la Torre

Documentación de las variables

Instrucciones __pad0__

Instrucciones con elementos de juegos de rol En el juego aparece un ejército de batalla cada uno con un diseño y estilo que deberán luchar contra otro ejército La dinámica del juego consiste en que un ejército trata de destruir la Torre y el otro ejército trata de protegerla ¡Por defender la Torre

Instrucciones con elementos de juegos de rol En el juego aparece un ejército de batalla cada uno con un diseño y estilo único

Referencia del Archivo juegoBase.cpp

```
#include "juegoBase.h"
```

Referencia del Archivo juegoBase.h

```
#include "avatar.h"  
#include "ejercito.h"  
#include "box.h"  
#include "tablero.h"
```

Clases

- class **JuegoBase**

Referencia del Archivo main.cpp

```
#include "avatar.h"
#include "ejercito.h"
#include "box.h"
#include "tablero.h"
#include "juegoBase.h"
#include "menu.h"
```

Funciones

- `int main ()`

Documentación de las funciones

`int main ()`

Referencia del Archivo menu.cpp

```
#include "menu.h"
```

Referencia del Archivo menu.h

```
#include "avatar.h"  
#include "ejercito.h"  
#include "box.h"  
#include "tablero.h"  
#include "juegoBase.h"
```

Clases

- class **Menu**

Referencia del Archivo nuevaPartida.txt

Referencia del Archivo partidaGuardada.txt

Referencia del Archivo README.md

Referencia del Archivo tablero.cpp

```
#include "tablero.h"
```

Referencia del Archivo tablero.h

```
#include "box.h"
```

Clases

- class **Tablero**

Referencia del Archivo Titulo.tx

