

**Escuela Profesional Ciencia de la Computación  
Universidad Nacional de San Agustín – Arequipa**

# **1702118**

## **Ciencias de la Computación II**

Agosto 2022

Docente: Alex Josué Flórez Farfán

### **Revisión del Proyecto**

**Grupo que realizó la revisión: Grupo 6**

- Quicaño Miranda, Victor Alejandro
- Ramirez Zarate, Patrick Rene
- Rodriguez Cutimbo, Gabriel Fernando

## Tabla de Contenido

1.	Introducción	3
2.	Instalación del software	3
3.	Estructura del software	4
4.	Revisión del código	4
5.	Ejecución del software	7
6.	Conclusiones	7

# Documento para revisión del software

Noten que ya no decimos *proyecto*, porque en este punto ya dejó de ser un proyecto, y ahora ya es un *trabajo final* que están presentando, por lo tanto ya es una versión de software para uso.

## 1. Introducción

Grupo revisado: *Grupo 5*

Link Github: <https://github.com/DiegoSaldanaM/EPCC-CCII-PROYECTO-FINAL.git>

Nombre del software: *Juego Bomberman*

Integrantes:

- *Ccolque Quispe, Anthony Criz*
- *Huamani Vilca, Juan Diego*
- *Lazo Paxi, Natalie Marleny*
- *Rodriguez Pumacayo, Ricardo Alexander*
- *Saldaña Moscoso, Diego Fernando*

## 2. Instalación del software

Sigan las instrucciones de instalación encontradas en el repositorio del software y respondan las siguientes preguntas.

Pregunta	Respuesta	Comentario
¿Cuáles son los requisitos para instalar el software?	Tener Visual Studio 2022.	
¿En qué plataforma se puede instalar el software?	Windows 10.	
¿Cuáles son las dependencias de software que indican las instrucciones?	Haber instalado WinForms en Visual Studio.	
¿Al seguir las instrucciones de instalación, consiguieron instalar y generar un código ejecutable?	No hay instrucciones de instalación.	Pero, al haber trabajado mi equipo también con Winforms, ya teníamos todo configurado para correr el código.
¿Las instrucciones de instalación son claras?	No hay instrucciones de instalación.	

### 3. Estructura del software

Pregunta	Respuesta	Comentario
¿En el repositorio se encuentra detallada la estructura del software?	No.	
¿La estructura del software está completa?	No.	
¿Tienen sugerencias para mejorar la estructura del software?	En el proyecto le falta que le agreguen comentarios a su código, para entenderlo mejor.	
¿El repositorio tiene algún diagrama que les ayude a comprender cómo está diseñado el software?	No.	
¿El repositorio contiene carpetas o archivos que no son útiles para generar el software?	Si.	Esto es debido a que tienen diferentes versiones del proyecto en un mismo repositorio. Para esta revisión, el proyecto que se está utilizando está dentro de la carpeta "ProyectoFinal_Bomberman_definitivo" que es que tiene los commit más recientes.
Enumere las carpetas en el código que tienen nombres adecuados que ayudan a identificar su propósito	imagenes y música.	Estas dos carpetas se ven en el proyecto de Visual Studio.
Enumere los archivos en el código que tienen nombres adecuados que ayudan a identificar su propósito	ArrBombas.h, ArrEnemigos.h, ArrMejoras.h, Bomba.h, Controladora.h, Enemigo.h, Escenario.h, Juego.h, Juego.cpp, Jugador.h, Mejora.h, MenuPrincipal.h, MenuPrincipal.cpp.	Todos los archivos del proyecto tienen un nombre que sí es adecuado para describir su función.
¿Cuál es el tipo de licencia usado en el software?	BSD-3.	

## 4. Revisión del código

### a. Preguntas acerca del estándar C++

Pregunta	Respuesta	Comentario
¿Cuál es el estándar de C++ utilizado que indica el software?	C++14	En realidad el lenguaje utilizado es C++/CLI, pero este, al ser un derivado de C++ depende del estándar de C++ que se esté utilizando.
¿Según Github cuáles son los lenguajes utilizados en este software?	Solo C++	Github lo detecta como si fuera C++, aunque, parte del código en realidad es C++/CLI.

### b. Para las siguientes preguntas indique en la respuesta el nombre del archivo, la clase y el número de línea de código.

Pregunta	Respuesta: nombre del archivo, la clase y el número de línea de código	Comentarios
Enumere las clases en el código que tienen nombres adecuados que ayudan a identificar su propósito	(ArrBombas.h, CArrBombas, 6), (ArrEnemigos.h, CArrEnemigos, 7), (ArrMejoras.h, CArrMejoras, 6), (Bomba.h, CBomba, 10), (Controladora.h, CControladora, 9), (Enemigo.h, CEnemigo, 12), (Escenario.h, CEscenario, 9), (Juego.h, Juego, 18), (Jugador.h, CJugador, 6), (Mejora.h, CMejora, 7), (MenuPrincipal.h, MenuPrincipal, 17).	
¿Hay clases cuyos nombres no ayudan a identificar su propósito? Indique cuales	No hay ninguna clase con un nombre que no ayude a identificar su propósito.	
Enumere los métodos de las clases en el código que tienen nombres adecuados que	(ArrBombas.h, CArrBombas, crear_Bomba, 13),	

ayudan a identificar su propósito	(ArrBombas.h, CArrBombas, dibujar_una_bombas, 19), (ArrBombas.h, CArrBombas, getarregloBombas, 39), (ArrEnemigos.h, CArrEnemigos, crearEnemigos, 14), (ArrEnemigos.h, CArrEnemigos, dibujarEnemigo, 15), (ArrEnemigos.h, CArrEnemigos, getarregloEnemigos, 17), (ArrMejoras.h, CArrMejoras, crearMejoras, 11), (ArrMejoras.h, CArrMejoras, dibujarMejoras, 12), (ArrMejoras.h, CArrMejoras, getVector_Mejoras, 14), (Bomba.h, CBomba, validarLugar, 35), (Bomba.h, CBomba, DibujarBomba, 42), (Bomba.h, CBomba, animar, 54), (Bomba.h, CBomba, DibujarExplosion, 65), (Bomba.h, CBomba, animarExplosion, 139), (Bomba.h, CBomba, animarExplosion, getEstado, 148), (Controladora.h, CControladora, cambiarNivel, 22), (Controladora.h, CControladora, AgregarBomba, 27), (Controladora.h, CControladora, disminuir_Vidas_por_Enemigo, 35), (Controladora.h, CControladora, disminuir_Vidas_por_Bomba, 44), (Controladora.h, CControladora, dibujar, 63), (Controladora.h, CControladora, crearMejoras, 76), (Controladora.h,	
-----------------------------------	--	--

	CControladora, crearEnemigos, 79), (Controladora.h, CControladora, getoJugador, 83), (Controladora.h, CControladora, getoArrMejoras, 87), (Controladora.h, CControladora, getoArrEnemigos, 91), (Controladora.h, CControladora, getNivel, 95), (Enemigo.h, CEnemigo, animarEnemigo, 31), (Enemigo.h, CEnemigo, dibujarEnemigo, 32), (Enemigo.h, CEnemigo, borrarEnemigo, 33), (Enemigo.h, CEnemigo, retornarRectangulo, 34), (Escenario.h, CEscenario, generarMatriz, 17), (Escenario.h, CEscenario, pintarBase, 46), (Escenario.h, CEscenario, pintarMatriz, 59), (Escenario.h, CEscenario, getMatriz, 76), (Juego.h, Juego, MusicaNivel, 138), (Juego.h, Juego, timer1_Tick, 143), (Juego.h, Juego, Juego_Load, 152), (Juego.h, Juego, mantenerTecla, 157), (Juego.h, Juego, ultimaTecla, 177), (Juego.h, Juego, trCarga_Tick, 188), (Jugador.h, CJugador, getVidas, 15), (Jugador.h, CJugador, getX, 19), (Jugador.h, CJugador, getY, 23), (Jugador.h, CJugador, ValidarMovimiento, 27), (Jugador.h, CJugador, disminuirvidas, 45), (Jugador.h, CJugador, disminuir_Vidas, 51), (Jugador.h, CJugador, dibujarJugador, 64),	
--	--	--

	(Jugador.h, CJugador, moverJugador, 80), (Jugador.h, CJugador, setDireccion, 149), (Mejora.h, CMejora, dibujar, 12), (Mejora.h, CMejora, animar, 13), (MenuPrincipal.h, MenuPrincipal, Musica, 273), (MenuPrincipal.h, MenuPrincipal, btnInstruccionDesaparecer_Click, 278), (MenuPrincipal.h, MenuPrincipal, btnInstruccionAparecer_Click, 283), (MenuPrincipal.h, MenuPrincipal, btnJugar_Click, 289), (MenuPrincipal.h, MenuPrincipal, MenuPrincipal_Load, 296).	
¿Hay métodos cuyos nombres no ayudan a identificar su propósito? Indique cuales	(Controladora.h, CControladora, eliminar_Enemigo, 31), (Jugador.h, CJugador, retornarRectangulo, 12),	Aquí van a estar los métodos que o bien su nombre no identifica su propósito, o estén vacíos.
Enumere las funciones libres en el código que tienen nombres adecuados que ayudan a identificar su propósito	(MenuPrincipal.cpp, main, 5).	
¿Hay funciones libres cuyos nombres no ayudan a identificar su propósito? Indique cuales	No hay ninguna función libre con un nombre que no identifique su propósito.	
¿El código contiene números "mágicos"? Es decir constantes que no se sabe cuál es su propósito.	(Controladora.h, CControladora, int nivel, 105)	
¿Los archivos tienen documentación? Enumérelas	No tienen documentación.	
¿Las clases tienen documentación? Enumérelas	No tienen documentación.	
¿Los métodos de las clases tienen documentación? Enumérelas	No tienen documentación.	



¿Las funciones libres tienen documentación? Enumérelas	No tienen documentación.	
¿Encontraron algún código que no sea C++ moderno? Enumérelas	En los archivos (Juego.h, MenuPrincipal.h y MenuPrincipal.cpp) se usa código en lenguaje C++/CLI, ya que, estos archivos fueron creados usando WinForms.	
¿Encontraron características que no corresponden al estándar C++?	Si, esto debido a que algunos archivos utilizan lenguaje C++/CLI en vez de solo C++.	

- c. Las siguientes preguntas están orientadas a evaluar el uso de las características y funcionalidades de C++. *Si encuentran estas características en el código enumérelas.*

Pregunta: Indique donde usa la característica siguiente	Respuesta: nombre del archivo, la clase y el número de línea de código	Comentarios
auto	-----	
new y delete	<p>New (ArrBombas.h, CArrBomas, 15), (ArrEnemigos.h, CArrEnemigos, 26), (ArrMejoras.h, CArrMejoras, 33), (Controladora.h, CControladora, 13, 14, 15, 16, 17), (Escenario.h, CEscenario, 13, 20), (Juego.h, Juego, 42).</p> <p>Delete (Juego.h, Juego, 56, 150), (MenuPrincipal.h, MenuPrincipal, 38).</p>	
Sobrecarga de funciones	-----	
Constructor		
Constructor explicit	-----	
Constructor copia	-----	
Operador de asignación	-----	
Destructor		
Idiom RAI	-----	
Herencia simple	-----	
Herencia múltiple	-----	

Métodos virtual	-----	
Métodos override	-----	
Métodos virtuales puros	-----	
Polimorfismo en tiempo de ejecución	-----	
Sobrecarga de operadores	-----	
Funciones template	-----	
Clases template	-----	
Asociación estructurada	-----	
Uso de namespaces	(ArrEnemigos.h, CArrEnemigos, 5), (Bomba.h, CBomba, 4, 5), (Enemigo.h, CEnemigo, 9), (Escenario.h, CEscenario, 7), (Juego.h, Juego, 7, 8, 9, 10, 11, 12, 13), (Jugador.h, CJugador, 3), (Mejora.h, CMejora, 5), (MenuPrincipal.h, MenuPrincipal, 6, 7, 8, 9, 10, 11, 12),.	
Uso de pair	-----	
Uso de tuple	-----	
Contenedor array	(Escenario.h, CEscenario, 13, 20).	
Contenedor vector	(ArrBombas.h, CArrBombas, 39, 44), (ArrEnemigos.h, CArrEnemigos, 11, 17, 22), (ArrMejoras.h, CArrMejoras, 14, 19).	
Contenedor list	-----	
Contenedor deque	-----	
Contenedor stack		
Contenedor queue, priority_queue	-----	
Contenedor map, unordered_map	-----	
Contenedor set, unordered_set	-----	
Manejo de excepciones	-----	
Código que use características de <random>	-----	
Entrada de archivos		
Salida de archivos	-----	
Algoritmos de la STL: sort, for_each, iota, etc.	-----	

Funciones o métodos recursivos	-----	
Operador de llamada a función	-----	
Especialización parcial de templates	-----	
Especialización completa de templates	-----	
Funciones lambda	-----	
Punteros inteligentes	-----	
Patrones de diseño	-----	
Programación concurrente	-----	

## 5. Ejecución del software

Pregunta	Respuesta	Comentario
¿Puede ejecutar el software desde un único archivo ejecutable?	No.	
¿La interfaz del software es fácil de usar?	Si.	
¿Al interactuar con el software puede identificar los objetivos que indican en el readme del repositorio?	Si.	
¿El software proporciona alguna ayuda en tiempo de ejecución para poder utilizarlo?	Si.	Únicamente las instrucciones para poder manejar el personaje.

## 6. Conclusiones

- Las clases que usaban new para generar valores en los punteros no tenían destructores y por ende no usaban delete para liberar la memoria correctamente.
- Por otro lado, hay una función en el programa que está vacía o sea no tienen código, está función está dentro del archivo "Controladora.h" y se llama "eliminar\_Enemigo".
- El juego consta de un solo nivel, por ende, no se puede pasar más allá del primer nivel.
- Los controles a la hora de jugar pueden ser algo toscos, pero no afectan demasiado a la experiencia.
- La única sugerencia para este proyecto sería que comentaran mucho más el código.