

Proyecto Final  
Informe

Informática II

Juan Diego Carrera Quintero - 1.004.249.012

Universidad de Antioquia  
Facultad de Ingeniería  
Departamento de Ingeniería Electrónica y Telecomunicaciones  
Semestre 2023-2

## ***Índice***

Resumen.....	- 1 -
Introducción .....	- 1 -
Marco teórico .....	- 2 -
Programación en C++ .....	- 2 -
Programación Orientada a Objetos (POO) .....	- 2 -
Procedimiento experimental y resultados .....	- 2 -
Resumen del capítulo .....	- 2 -
Ideas sobre el juego .....	- 2 -
Problemas en el desarrollo.....	- 3 -
Discusión de resultados.....	- 4 -
Conclusiones .....	- 5 -

## ***Resumen***

En este proyecto, se aplicaron los conocimientos adquiridos durante el curso de Informática II a lo largo del semestre para desarrollar un juego inspirado en un episodio de la serie Rick & Morty. Se implementaron habilidades fundamentales de programación en C++, abordando la semántica y sintaxis, así como el manejo avanzado de conceptos como punteros, memoria dinámica, contenedores y manipulación de archivos. Además, se integraron principios de programación orientada a objetos para estructurar de manera eficiente el código y mejorar la modularidad del proyecto. El resultado es un juego sencillo que destaca por la aplicación cohesiva y efectiva de los elementos mencionados, demostrando la habilidad para combinar y aplicar los fundamentos de la programación, así como los principios de la programación orientada a objetos, en la creación de un producto funcional y entretenido.

Palabras clave: C++, Contenedores, Memoria dinámica, POO, Punteros

## ***Introducción***

La presente investigación se adentra en el análisis y desarrollo de un proyecto que fusiona los fundamentos esenciales de la programación en C++ con los principios de la programación orientada a objetos (POO), haciendo hincapié en el manejo de contenedores, memoria dinámica y punteros. Inspirado en un episodio de la serie Rick & Morty, este proyecto se erige como un escenario idóneo para poner a prueba los conocimientos adquiridos durante el curso de Informática II a lo largo del semestre.

El objetivo principal de esta iniciativa es someter a examen la comprensión y aplicación práctica de los conceptos fundamentales de la programación en C++ y la POO. A través de la creación de un juego, se busca identificar posibles falencias en el entendimiento de la semántica, sintaxis y otros aspectos clave del lenguaje de programación C++. Además, se pretende evaluar la capacidad para estructurar y organizar el código mediante la aplicación efectiva de los principios de la POO, promoviendo la modularidad y la reutilización de código.

Este proyecto también se propone como un instrumento para fortalecer los conocimientos previos y reforzar aquellas áreas que puedan presentar debilidades. La manipulación de contenedores, el manejo de memoria dinámica y la comprensión de punteros son aspectos cruciales en la programación en C++, y este proyecto ofrece la oportunidad de profundizar en estos conceptos a través de una aplicación práctica.

Además de evaluar los conocimientos técnicos, se busca fomentar la capacidad para abordar y resolver problemas de manera creativa y eficiente. La resolución de desafíos específicos que surgirán durante el desarrollo del juego permitirá no solo demostrar las habilidades técnicas, sino también cultivar la destreza para enfrentar situaciones complejas de programación.

En resumen, este proyecto no solo aspira a la creación de un juego basado en Rick & Morty, sino a constituirse como una plataforma integral para evaluar, fortalecer y consolidar los conocimientos en programación en C++ y programación orientada a objetos, proporcionando una experiencia práctica en la que los estudiantes puedan poner a prueba sus habilidades y avanzar en su desarrollo como programadores competentes.

## ***Marco teórico***

### **Programación en C++**

La programación en C++ constituye el fundamento esencial de este proyecto. Se basa en la implementación de algoritmos y estructuras de datos, aprovechando la semántica y sintaxis de C++ para desarrollar software eficiente y funcional. Los conceptos clave incluyen la manipulación de punteros, el uso de memoria dinámica y la gestión eficaz de contenedores, elementos que desempeñarán un papel crucial en la creación del juego.

### **Programación Orientada a Objetos (POO)**

La POO proporciona un enfoque estructurado para el diseño de software, fomentando la modularidad y la reutilización del código. En este contexto, los conceptos de clases y objetos permitirán organizar la lógica del juego de manera eficiente. La herencia y el polimorfismo, pilares de la POO, ofrecen herramientas poderosas para crear una arquitectura flexible y fácil de mantener.

## ***Procedimiento experimental y resultados***

Inicialmente, optamos por seleccionar un capítulo de manera aleatoria, evaluando su atractivo y la calidad de su narrativa. No perseguíamos exclusivamente una trama épica o llena de acción; más bien, buscábamos una historia que resultara envolvente y entretenida. Tras una cuidadosa revisión, nos inclinamos por el capítulo 1 de la temporada 5, originalmente titulado "Mort Dinner Rick Andre".

### **Resumen del capítulo**

Tras escapar por poco con vida de su más reciente aventura con su abuelo, Morty se las arregla para aterrizar la nave en el océano habiendo convencido a su compañera de clases Jessica de tener una cita. La presencia de Rick en el océano pone en juego una tregua entre él y el Sr. Nimbus, el gobernador del mar en la tierra. Un frustrado Rick se ve forzado a preparar una cena para renegociar su tregua con Nimbus mientras envía a Summer a robar una concha en el reino subacuático. Morty por su parte quiere tener su cita con Jessica, pero se ve constantemente interrumpida por tener que ayudar a su abuelo con la cena además de que inicia una guerra contra los habitantes de una dimensión alterna en la que el tiempo transcurre miles de veces más rápido. Las cosas se complican cuando Jessica es secuestrada por los habitantes de la dimensión por lo que Morty obliga a su abuelo a que lo ayude a rescatar a la chica, logrando escapar gracias a que Nimbus interviene. Aunque Jessica regresa a salvo es completamente diferente por haber permanecido congelada por siglos en la otra dimensión por lo que decide quedar solo como una amiga con Morty mientras que Nimbus aprisiona a Rick en la cárcel al descubrir sus intenciones de hacerse con la concha de su reino.

### **Ideas sobre el juego**

El producto final difiere significativamente de nuestra concepción inicial, la cual lamentablemente no pudo concretarse debido a diversos factores que se detallarán más adelante. En un principio, contemplábamos la creación de un juego compuesto por 4 a 6 niveles. El primer nivel proponía la caída de Rick y Morty en su nave hacia la Tierra, donde debían esquivar obstáculos antes de aterrizar. Los niveles 2 y 3 consistían en una búsqueda de botellas de vino y la

eliminación de alienígenas para obtener experiencia y mejorar la vida y el daño del personaje principal. El cuarto nivel planteaba una confrontación contra un ejército de alienígenas y el rescate de otro personaje, mientras que el quinto nivel se centraba en una batalla épica contra un jefe final.

Sin embargo, por diversas complicaciones, tomamos la decisión de reducir el alcance del juego a solo 2 niveles. Estos consistían en la recolección de un número específico de botellas de vino y, finalmente, el enfrentamiento con el jefe final. A pesar de nuestros esfuerzos, nos encontramos con obstáculos que limitaron la realización del proyecto a la mitad de nuestras expectativas originales. Estos desafíos nos impulsaron a adaptarnos y concentrarnos en la ejecución exitosa de las partes seleccionadas del juego.

### **Problemas en el desarrollo**

A pesar de contar inicialmente con una visión clara para nuestro juego, nos vimos confrontados por un adversario implacable: el tiempo. Conscientes de esta limitación, optamos por realizar ajustes en nuestra idea original, simplificándola para adaptarnos a las restricciones temporales. A pesar de estos esfuerzos, el tiempo continuó siendo un desafío constante hasta el último minuto de entrega.

Sin embargo, el tiempo no fue el único desafío que enfrentamos. Inicialmente, éramos un equipo de dos personas dedicadas al proyecto, una dinámica que se mantuvo hasta los días finales previos a la entrega final. En ese momento crítico, uno de los integrantes decidió abandonar el proyecto. Esta decisión se materializó justo cuando su responsabilidad recaía en la creación del jefe final del juego, y lamentablemente, no había avanzado en esa parte. Como resultado, la tarea de terminar el juego quedó en mis manos con un tiempo limitado, lo que impidió completar la segunda parte del proyecto dentro del plazo establecido. Este imprevisto añadió una capa adicional de desafío a la gestión del tiempo ya complicada.

### ***Desarrollo del Proyecto***

A pesar de los desafíos iniciales, durante la ejecución del proyecto, decidí implementar una estructura de código (Ilustración 1) que contribuyera a la claridad y eficiencia del desarrollo. Esta estructura se compone de dos clases fundamentales, además de la clase "mainwindow" destinada a la interfaz gráfica.

La primera clase, denominada "Game", asume la responsabilidad crucial del funcionamiento del juego. Aunque no aborda aspectos específicos como el movimiento de objetos, se encarga de orquestar y coordinar dichos movimientos. La segunda clase, denominada "GameObject", cumple su función según su nombre, proporcionando información general acerca de los objetos utilizados en el juego.

Finalmente, se introducen dos clases adicionales, "Morty" y "Enemy", que heredan de la clase "GameObject". Estas clases especializadas representan respectivamente al personaje principal y a los enemigos del juego, estableciendo una jerarquía que facilita la gestión y manipulación de los elementos dentro del código.

Este enfoque estructurado tiene como objetivo optimizar la mantenibilidad y comprensión del código, brindando una organización lógica que facilita la implementación, modificación y expansión del proyecto en el futuro.

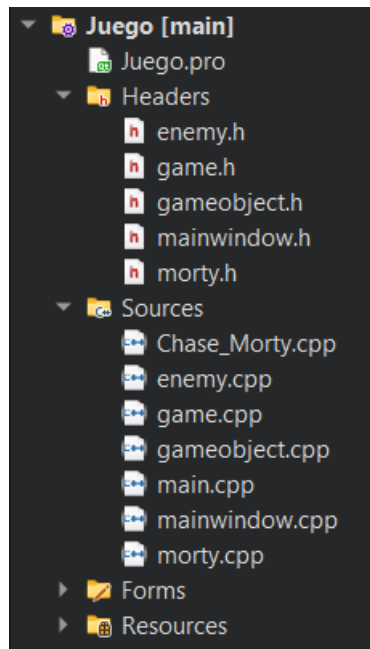


Ilustración 1 - Estructura de código

Con la estructura de código ya conceptualizada y parte de la implementación en curso, la siguiente fase se centró en la incorporación de recursos visuales (Ilustración 2). El paso subsecuente implicó la creación de los sprites que representarían los movimientos tanto del personaje principal como de los enemigos. Además, se abordó la generación de los objetos dentro del entorno del juego, considerando que el mapa sería construido a partir de un archivo de texto.

La creación del mapa, al basarse en un archivo de texto, requirió iteraciones en el diseño para alcanzar la versión que mejor se ajustara a los objetivos del proyecto. Este proceso involucró la elaboración de múltiples modelos de mapa, evaluando y refinando cada iteración hasta llegar a la versión que mejor satisfacía mis criterios y requisitos.

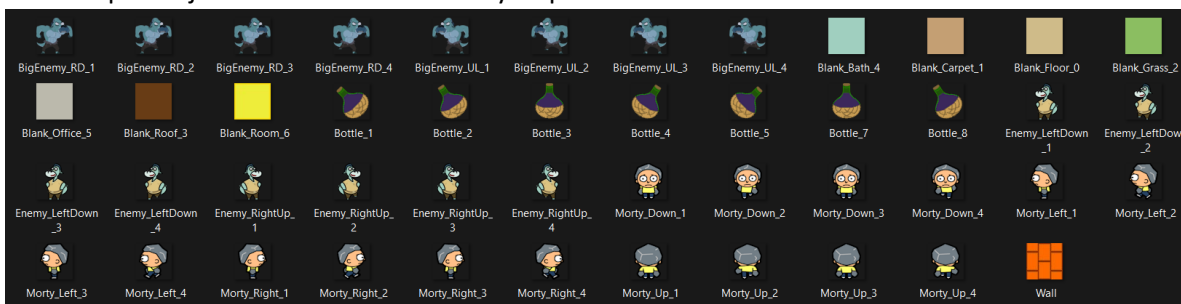
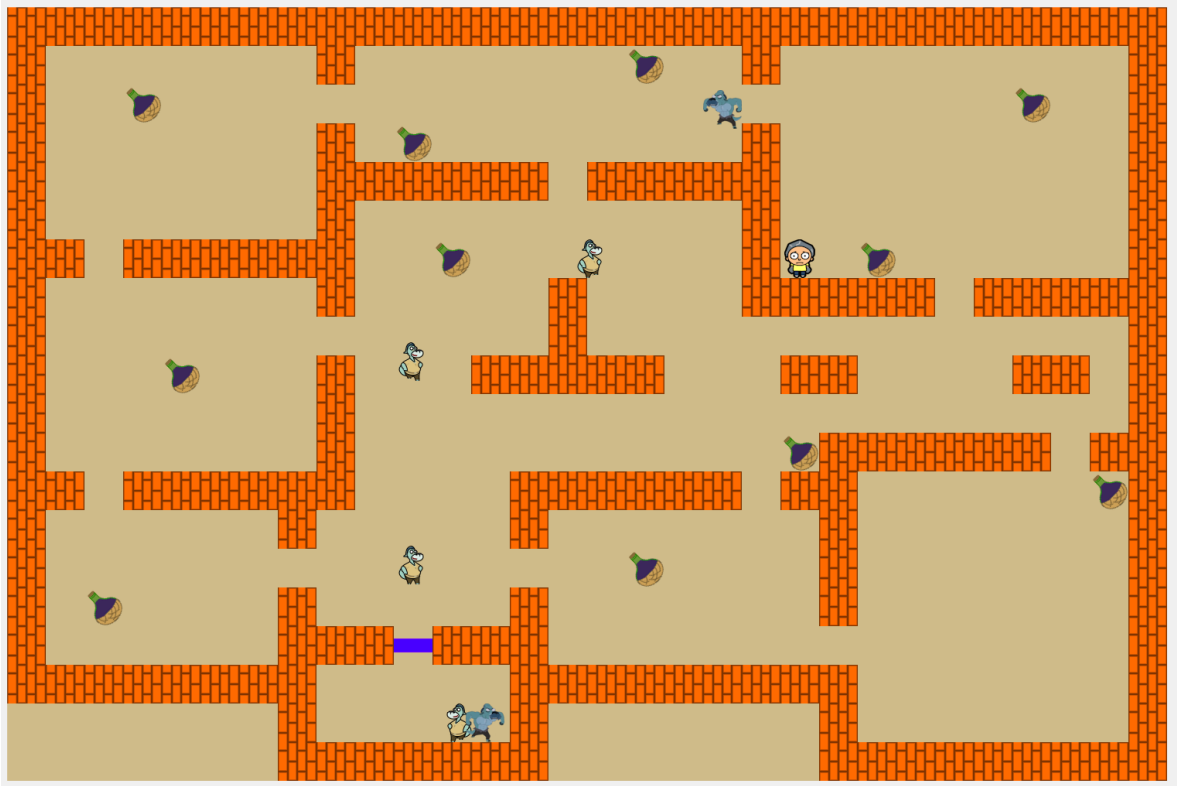


Ilustración 2 - Recursos visuales

## Discusión de resultados

El producto final es un juego simple cuyo propósito es recolectar todas las botellas, aunque la ambición inicial de seguir con la idea original se vio limitada por circunstancias inevitables. A pesar

de las restricciones, el resultado final es motivo de satisfacción. Aunque no logramos concretar la visión completa, el juego destaca por su simplicidad y el desafío de reunir todos los elementos. La experiencia refleja la adaptabilidad ante obstáculos y la capacidad para obtener un producto funcional y entretenido dentro de las restricciones existentes. Este proyecto subraya la importancia de la flexibilidad y la satisfacción derivada de superar desafíos en el proceso creativo.



## ***Conclusiones***

A través de este proyecto, pude apreciar la notable facilitación que la programación orientada a objetos brinda al desarrollo. La versatilidad y organización se manifestaron claramente, permitiendo una estructuración eficiente del código.

La capacidad de representar conceptos como clases y objetos no solo simplificó la implementación, sino que también proporcionó un marco ordenado y escalable para abordar los diversos aspectos del juego.

Además, me brindó la oportunidad de someterme a pruebas, trabajar de manera efectiva bajo presión y demostrar mi capacidad para adaptarme a las adversidades. Este proyecto no solo fue un ejercicio técnico, sino también una experiencia en la gestión del tiempo y la resiliencia frente a desafíos imprevistos. La necesidad de ajustar la visión original y tomar decisiones rápidas en situaciones apremiantes fortaleció mi habilidad para enfrentar obstáculos y encontrar soluciones prácticas en un entorno dinámico de desarrollo de software.

Aunque reconozco que hay áreas de mejora, estoy comprometido a perfeccionar mis habilidades con el tiempo. Esta conciencia de mis puntos de desarrollo me impulsa a buscar constantemente oportunidades de aprendizaje y práctica. Estoy seguro de que, a medida que

continúo dedicándome al crecimiento profesional, podré abordar de manera efectiva las áreas identificadas para mejorar. El proceso de perfeccionamiento es continuo, y estoy enfocado en aprovechar cada experiencia para avanzar y alcanzar mis metas.