Desarrollo de Pong

Arnold Merchan, Santiago Guarin Alfaro

*Universidad Nacional de Colombia*

*Bogotá, Colombia*

ajmerchanr@unal.edu.co

sguarina@unal.edu.co

***Abstract***— **En este documento se aborda lo correspondiente a aspectos importantes para el desarrollo del juego pong así como lo son los diferentes requerimientos que deberá cumplir el proyecto, la historia de usuario y las librerías que se usarán para la programación del juego.**

I. INTRODUCCIÓN

El presente documento tiene como objetivo detallar los aspectos esenciales para el desarrollo del juego *Pong*, abordando requerimientos técnicosue fundamentan el proyecto. Este trabajo incluye una descripción de las funcionalidades clave que el juego deberá cumplir, una narrativa breve basada en la historia de usuario para contextualizar su diseño, y una explicación de las herramientas y librerías seleccionadas para su implementación.

A través de este documento, se busca proporcionar una guía clara y estructurada para llevar a cabo el desarrollo del proyecto, asegurando que todas las etapas, desde la planificación hasta la ejecución, se realicen de manera eficiente y acorde con las expectativas de los usuarios finales. Con ello, se espera garantizar la creación de un producto funcional

II. REQUERIMIENTOS

Para el desarrollo del proyecto, es fundamental establecer los requerimientos que deberá cumplir el producto final. Esto permitirá garantizar su correcta funcionalidad y ejecución.

A continuación, se presenta una tabla que detalla los principales requerimientos necesarios para el desarrollo del juego *Pong*. Estos requerimientos se han definido para asegurar que el proyecto cumpla con los objetivos funcionales y de experiencia del usuario establecidos.

TABLA I

REQUERIMIENTOS NECESARIOS PARA EL DESARROLLO DEL JUEGO PONG

| **Requerimientos** | |
| --- | --- |
| **Código** | **Descripción** |
| RQMTO001 | Se requiere de una interfaz gráfica intuitiva y funcional que permita al usuario interactuar con un  menú de ingreso al sistema |
| RQMTO002 | Se requiere de una interfaz gráfica interactiva que permita a los usuarios ingresar al sistema y  seleccionar entre tres modos de juego disponibles. |
| RQMTO003 | Se requiere implementar un primer modo de juego en el que dos usuarios puedan participar  simultáneamente en el mismo dispositivo. |
| RQMTO004 | Se requiere implementar un segundo modo de juego en el que un solo usuario pueda practicar  libremente contra un bot |
| RQMTO005 | Se requiere implementar un marcador visible, el marcador se encargará de indicar el puntaje de  cada jugador en la partida y finalizará el juego cuando el marcador de algún jugador llegue a 5. |
| RQMTO006 | Se necesita implementar un tercer modo de juego que mantenga la dinámica clásica de Pong  pero modifique el objetivo principal. En este modo, el jugador deberá controlar una paleta para  golpear una pelota que, en lugar de enfrentarse a otro oponente, debe destruir una serie de  bloques ubicados en la parte superior de la pantalla. |
| RQMTO007 | Se requiere programar una paleta móvil controlada por el usuario, capaz de desplazarse  horizontalmente mediante las teclas de dirección, sin salir de los límites de la pantalla. |
| RQMTO008 | Se requiere diseñar una pelota que se mueve en diagonal con una velocidad constante, capaz  de rebotar en los bordes de la pantalla, la paleta y los ladrillos. |
| RQMTO009 | Se requiere implementar un sistema que reinicie la posición de la pelota al centro de la pantalla  en caso de que esta caiga fuera de los límites inferiores. |
| RQMTO010 | Se deben colocar ladrillos en filas y columnas, diseñados para eliminarse al ser golpeados por  la pelota y que incrementen la puntuación del jugador. |
| RQMTO011 | Se debe implementar un sistema de puntuación que muestre en tiempo real la cantidad de  puntos acumulados |
| RQMTO012 | Es necesario añadir un sistema de vidas que reduzca el número de vidas restantes al perder  la pelota y finalice el juego cuando las vidas lleguen a cero. |
| RQMTO013 | Se de una interfaz gráfica que actualice en tiempo real la posición de los elementos del juego,  así como la puntuación y las vidas restantes. |
| RQMTO014 | Se Requiere implementar un sistema de dificultad que aumente progresivamente la velocidad  de la pelota a medida que avanza el juego. |
| RQMTO015 | Se requiere implementar un botón de pausa accesible en cada uno de los tres modos de juego.  Este botón permitirá al usuario detener temporalmente la partida, mostrando un menú con  opciones adicionales y permitiendo reanudar el juego o realizar otras acciones. |
| RQMTO016 | Se requiere hacer uso de una paleta de colores óptima que garantice la visibilidad clara de  todos los objetos presentes en el juego |
| RQMTO017 | Se requiere configurar los controles del juego para que los jugadores puedan manejar sus  acciones utilizando tanto las teclas de flecha como las letras A y D. |
| RQMTO018 | Se requiere utilizar librerías óptimas y adecuadas para la creación de las interfaces de  usuario y la implementación de las funcionalidades principales del juego |
| RQMTO019 | Se requiere que cada modo de juego disponga de un botón específico que permita iniciar la  partida correspondiente |

III. HISTORIA DE USUARIO

Definir una historia de usuario resulta esencial para contextualizar el diseño del proyecto y delimitar sus alcances. Este enfoque no solo ayuda a visualizar las necesidades y expectativas del jugador, sino que también orienta el proceso de programación y construcción del juego.

Para guiar el diseño y desarrollo del juego, se ha definido una historia de usuario que describe las necesidades y expectativas del usuario final. Esta narrativa proporciona un contexto claro sobre el uso del producto y permite enfocar los esfuerzos en la creación de una experiencia que sea atractiva, funcional y alineada con las expectativas del jugador. A continuación, se presenta una tabla que detalla la historia de usuario y sus principales características.

TABLA II

HISTORIA DE USUARIO PARA EL JUEGO PONG

| Historia de Usuario 1 |
| --- |
| Precondición: El jugador está en el menú principal y las opciones de los modos de  juego son visibles. |
| El jugador visualiza el título del juego y las opciones disponibles en el menú. |
| El jugador utiliza las teclas de flecha para desplazarse entre las opciones del menú |
| El jugador presiona Enter para seleccionar una opción |
| El juego carga el modo seleccionado y muestra la interfaz correspondiente. |
| El sistema muestra una pantalla específica del modo seleccionado, incluyendo un  botón "Iniciar". |
| El jugador presiona el botón "Iniciar" para comenzar la partida |
| Historia de Usuario 2 |
| Precondición: El jugador ha seleccionado el modo multijugador desde el menú  principal. |
| Los jugadores visualizan el botón "Iniciar" en la pantalla. |
| Uno de los jugadores presiona el botón "Iniciar" para empezar la partida |
| El juego comienza con las paletas y la pelota en posición inicial. |
| Inicialmente la pelota va hacia el jugador 1 |
| El jugador 1 usando teclas de flecha ubica la paleta de tal manera que la pelota  rebote en la paleta |
| El jugador 2 usando las teclas A y D ubica la paleta de tal manera que la pelota  rebote en la paleta |
| Los dos jugadores continúan de la misma manera |
| A medida que va avanzando el tiempo la velocidad de la pelota va aumentando  progresivamente |
| Historia de Usuario 2.1 |
| Precondición: Alguno de los dos usuarios deja pasar la pelota |
| El otro jugador suma un punto |
| Se reinicia la posición inicial de las pelotas y las paletas |
| Alguno de los dos jugadores completan 5 puntos |
| Los jugadores visualizan el ganador de la partida y dos botones, el botón  "Reiniciar" y el botón "volver al Inicio" |
| Historia de usuario 3 |
| Precondición: El jugador ha seleccionado el modo contra bot desde el menú  principal. |
| El jugador visualiza el botón "Iniciar" en la pantalla. |
| El jugador presiona el botón "Iniciar" para empezar la partida |
| El juego comienza con las paletas y la pelota en posición inicial. |
| El jugador controla su paleta utilizando las teclas asignadas (flechas ). |
| El bot responde dinámicamente a los movimientos de la pelota para simular una  competencia. |
| Historia de Usuario 3.1 |
| Precondición: El usuario o el bot deja pasar la pelota |
| Pierde una de las vidas |
| Se reinicia la posición inicial de las pelotas y las paletas |
| Alguno de los dos jugadores pierde todas las vidas |
| Los jugador visualiza el ganador de la partida y dos botones, el botón  "Reiniciar" y el botón "volver al Inicio" |
| Historia de Usuario 4 |
| Precondición: El jugador ha seleccionado el modo destruye los bloques |
| El jugador visualiza el botón "Iniciar" en la pantalla. |
| El jugador presiona el botón "Iniciar" para empezar la partida |
| Se encuentra la pelota y la paleta en su posición inicial |
| El jugador visualiza una serie de bloques organizados en filas y columnas |
| La pelota rebota en los bloques |
| El jugador ubica la paleta de tal forma que la pelota rebote en la paleta |
| El jugador a medida que va destruyendo bloques va aumentando su puntaje |
| Al jugador le pasa la pelota por la parte inferior de la pantalla |
| El jugador visualiza su puntaje a sido(xxx)y dos botones, el botón  "Reiniciar" y el botón "volver al Inicio" |

IV. LIBRERÍAS

A continuación, se describen las posibles librerías que se emplearán en la programación del juego, junto con una explicación de su función y relevancia dentro del desarrollo.

*pygame: librería especializada principalmente en la creación de videojuegos en 2D usando python dando más acceso a manipulación de ficheros audios y periféricos*

*random: librería para generar valores aleatorios o cambiar el orden de algunos elementos de forma aleatoria*

*turtle: librería principalmente para apartados gráficos y movimiento con coordenadas, se usa principalmente para dibujos y personalización de los mismos*

*math: librería especializada en operaciones y signos matemáticos (se pueden usar configuraciones de ángulos para las físicas en el pong)*

REFERENCIAS

| [1] | <https://docs.python.org/3/library/math.html> math — Mathematical functions. |
| --- | --- |
| [2] | <https://docs.python.org/3/library/turtle.html> turtle — Turtle graphics. |
| [3] | https://www.pygame.org/wiki/about |
| [4] | <https://docs.python.org/3/library/random.html> random — Generate pseudo-random numbers |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |