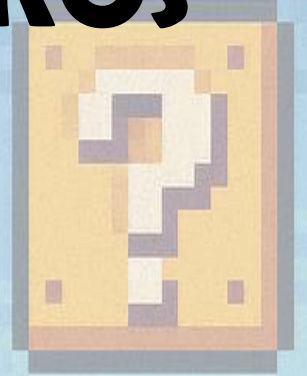


SUPER WALLE BROs

Wall-E en el **Reino Champiñón**



RAÚL ROBERTO ESPINOSA POMA

2do Proyecto de Programación

C121 MATCOM 2024-2025

Contexto del proyecto:

El cosmos, tan vasto y misterioso, a veces tiene un sentido del humor peculiar. Wall-E flotaba a la deriva, sumido en sus pensamientos ecológicos y soñando con su querida EVA. De repente, un destello verde esmeralda, una estela de polvo de estrellas y un inexplicablemente familiar sonido de "¡Pling!" lo envolvieron. Cuando el brillo se desvaneció, Wall-E no estaba en el frío vacío del espacio, sino sobre un vibrante bloque de interrogación flotante, con nubes pixeladas y un castillo reconocible a lo lejos.

Había aterrizado, sin querer, en el Reino Champiñón. Su inocente recolección de "basura" (Goombas y Koopa Troopas rotos) no tardó en llamar la atención de un furioso Mario. "¡Mamma mia! ¡Estás pisoteando mis niveles y usando mis cosas sin licencia!", exclamó Mario. Wall-E, sin entender, solo podía decir "¡Wall-E!". Sin saberlo, había activado la alarma de Nintendo por infracción de derechos de autor. Ahora, este pacífico robot se encontraba en medio de una demanda interdimensional. Su única esperanza de salir de este lío y volver a casa era encontrar lo máspreciado que traía consigo: su pequeña bota con la última planta viva.

Con abogados de setas afilando sus plumas, Wall-E debe navegar por los peligrosos niveles de Mario, evitando no solo a los enemigos sino también la factura de la demanda. Su misión: recuperar su bota antes de convertirse en una exhibición legal.

¿Podrá Wall-E recuperar su bota y escapar de las garras legales del Reino Champiñón? ¿O terminará como una exhibición en el Museo de la Historia de Nintendo por "uso no autorizado de activos"?

El proyecto "**SUPER WALLE BROS**" se divide en dos módulos principales:

- Editor de Pixel Art: Permite a los usuarios instrucciones simples que le permiten "pintar" en un lienzo.
- Generador de Niveles de Plataformas: Una característica extendida que transforma imágenes de textura 2D en niveles jugables de plataformas, donde Wall-E puede moverse y alcanzar objetivos.

Tecnologías utilizadas:

Para el desarrollo de este juego se emplearon diversas tecnologías para garantizar una experiencia fluida y agradable:

- Unity: Motor principal utilizado para el desarrollo del juego.
- C#: Lenguaje de programación empleado para la lógica del juego y sus mecánicas.
- Git: Herramienta de control de versiones para gestionar el progreso del proyecto.

Desarrollo y Plataforma

El intérprete de comandos y la lógica de dibujo sobre el canvas están implementados para procesar una serie de comandos específicos y cumplir ciertas características de un lenguaje de programación, definido en el documento PDF Pixel-WallE. Este lenguaje permite realizar acciones como:

- Spawn (int x, int y): Inicializa la posición de Wall-E en el canvas.
- Color (string color): Cambia el color del pincel.
- Size (int k): Modifica el tamaño del pincel.
- DrawLine (int dirX, int dirY, int distance): Dibuja una línea en una dirección y distancia específicas.
- DrawCircle (int dirX, int dirY, int radius): Dibuja un círculo.
- DrawRectangle (int dirX, int dirY, int distance, int width, int height): Dibuja un rectángulo.
- Fill (): Rellena un área con el color actual del pincel.

- Asignación de variables: Permite la declaración y uso de variables numéricas y booleanas.
- Expresiones aritméticas y booleanas: Soporta operaciones básicas y comparaciones.
- Funciones: Incluye funciones para obtener la posición de Wall-E, el tamaño del canvas, contar píxeles de un color, y verificar el color y tamaño del pincel.
- Saltos Condicionales (GoTo [label] (condition)): Permite el control de flujo del programa.

Mecánicas de Juego (Niveles de Plataformas)

- Objetivo Principal: Navegar a través de niveles derrotando enemigos para encontrar la "bota con la planta de Wall-E", que representa el punto de victoria del nivel.
- Movimiento: Wall-E se controla con comandos básicos de plataforma.
- Generación de Niveles: Los niveles se construyen a partir de texturas 2D predefinidas por el usuario. Los colores en la textura se mapean a diferentes tipos de tiles o prefabs dentro del juego, creando el entorno del nivel.

Controles:

- SPACE: Saltar
- DOWN ARROW: Agacharse
- LEFT ARROW: Mover hacia la izquierda
- RIGHT ARROW: Mover hacia la derecha

Requisitos para la apertura y edición

En caso de no desear abrir o descargar el ejecutable, para abrir y ejecutar el proyecto "Pixel Wall-E", se requiere el siguiente entorno de desarrollo y software:

- Unity Hub: Es la herramienta de gestión para proyectos de Unity y versiones del editor. Se recomienda instalar la versión más reciente o una compatible con la versión de Unity utilizada en el desarrollo del proyecto.

- Unity Editor: Una versión compatible del editor de Unity (se recomienda Unity 2022.3.x LTS o superior para asegurar compatibilidad).
- Visual Studio (con la carga de trabajo "Desarrollo de juegos con Unity") o Visual Studio Code (con las extensiones de C# y Unity) para editar los scripts.
- Sistema Operativo: Compatible con Unity (Windows, macOS, Linux).

Pasos para abrir el proyecto:

1. Clonar o Descargar el Repositorio: Obtener el código fuente del proyecto desde el repositorio de GitHub provisto.
2. Abrir con Unity Hub: Iniciar Unity Hub, hacer clic en "Add" (Agregar) y seleccionar la carpeta raíz del proyecto descargado.
3. Abrir en el Editor de Unity: Seleccionar el proyecto en Unity Hub y abrirlo con la versión de Unity Editor adecuada. Unity puede tardar un tiempo en importar todos los activos y scripts la primera vez.

Una vez abierto, el proyecto esta listo para ser ejecutado directamente desde el editor de Unity, permitiendo probar tanto el intérprete de comandos como los niveles generados a partir de texturas generadas por documentos con la extensión .pw que deberán poseer el código a procesar.

Extras

En caso de querer saber parte de la estructura de los scripts del proyecto y el intérprete, puedes consultar el documento EstructuraDelProyecto.pdf.

