

BALKANOID

RAZONAMIENTO AUTOMÁTICO

Integrantes del grupo:

- Eduard Andrei Duta.
- Francisco Wendeburg.

Introducción

Presentación del juego

Balkanoid es un videojuego que se sitúa en el género de arcade, siendo una reimaginación del clásico juego Arkanoid. Desarrollado para la plataforma Amstrad CPC 464, Balkanoid se distingue por su fusión única de elementos tradicionales y culturales balcánicos con mecánicas de juego ágil y desafiante.

La realización de Balkanoid se llevó a cabo utilizando el lenguaje de programación ensamblador Z80, aprovechando la robustez y las capacidades del framework CPCtelera. Este marco de desarrollo proporciona un conjunto de herramientas que facilitaron la implementación de las características esenciales del juego, así como la incorporación de elementos gráficos y musicales auténticos.

La música balcánica, conocida por sus ritmos vivaces y su carácter emotivo, se integra en el corazón de la experiencia de juego, creando una atmósfera envolvente que captura la esencia de la región balcánica. Además, los sprites y los elementos de diseño visual están inspirados en motivos tradicionales balcánicos, lo que enriquece la estética del juego y proporciona una conexión cultural profunda.

La motivación detrás de Balkanoid va más allá de la simple recreación de un juego clásico; busca ofrecer una experiencia que resuene tanto a nivel estético como mecánico, proporcionando un punto de encuentro entre lo tradicional y lo contemporáneo en el vasto dominio de los videojuegos.

Inspiración

La chispa inicial para la creación de Balkanoid se encendió a partir del clásico juego de arcade, Arkanoid. Este título, conocido por su jugabilidad adictiva y sus mecánicas simples, pero desafiantes, sentó un sólido precedente en el género arcade. Sin embargo, nuestro objetivo era llevar esta experiencia clásica a un nuevo nivel, enriqueciendo la estética y la atmósfera del juego con elementos culturales balcánicos.

La cultura balcánica, con su rica tradición musical y su distintivo estilo artístico, ofreció un vasto terreno de inspiración. La música balcánica, en particular, con sus ritmos vibrantes y su capacidad para evocar emociones profundas, se identificó como un elemento clave que podría mejorar la inmersión y la conexión emocional de los jugadores con el juego. Así, se emprendió una cuidadosa selección y adaptación de piezas musicales balcánicas que se integran armoniosamente en el diseño sonoro de Balkanoid.

En cuanto al diseño visual, nos propusimos explorar y reinterpretar motivos tradicionales balcánicos en una forma que respetase la esencia de esta rica tradición, al tiempo que aporta una estética nueva al juego. Los sprites y los elementos de diseño visual se concibieron con atención al detalle, logrando una fusión visual que homenajea tanto a la herencia balcánica como al estilo retro de los videojuegos de arcade.

Además, el deseo de explorar las posibilidades de desarrollo en una plataforma clásica como Amstrad CPC, junto con el uso del framework CPCTelera, también fue una fuente de inspiración técnica. Esta elección nos permitió enfrentar los desafíos únicos que supone el desarrollo en una plataforma con recursos limitados, lo que a su vez impulsó nuestra creatividad y habilidades técnicas.

En suma, Balkanoid se forjó en la intersección de la admiración por un clásico de los videojuegos, la apreciación de la cultura balcánica y el deseo de explorar y expandir nuestras capacidades técnicas en un marco de desarrollo desafiante.

Objetivos del proyecto

El objetivo de este proyecto fue el desarrollo de un videojuego junto con sus sprites y la música para el Amstrad CPC 464 lo que implica fuertes restricciones hardware que obligan a la optimización del uso de memoria y también de los algoritmos utilizados.

Proceso de desarrollo

Restricciones y Herramientas

El lenguaje utilizado es ensamblador para el procesador Zilog Z80. Algunas de las características de este procesador son:

- El tamaño de la palabra es de 8 bits .
- Los datos se guardan en memoria en formato little-endian lo que significa que los valores menos significativos se guardan en la dirección más alta.
- Los registros son de 8 bits, pero algunas instrucciones permiten usar dos registros como uno solo.
- La velocidad del reloj del procesador es de 4Mhz.
- Las instrucciones de este procesador son muy básicas. Siendo las más utilizadas: sub, add, ld, call, ret, jr, jp.

En cuanto a la memoria RAM disponible esta es de 64Kb.

Para el desarrollo se utilizó el editor de texto Visual Studio Code.

Arquitectura del Juego

El game engine utilizado es CPCtelera, lo que facilitó mucho la conversión de sprites y sonidos a bytes para poder ser usados dentro del juego. También fue muy útil debido a sus funciones para dibujar sprites en pantalla, reproducir música, etc.

El código se estructuró en diferentes partes:

- Archivo main.s: Este archivo se encarga de configurar el modo de la pantalla, la paleta de colores y de llamar a las funciones para mostrar las pantallas de inicio y de fin. También se encarga de llamar al game manager para inicializar el juego comenzar el game loop.
- Manager: Dentro de esta carpeta se encuentran los archivos para la gestión del juego. Por ejemplo el game manager, para el control de flujo del juego, el scoreboard manager quien se encarga de mostrar y actualizar las vidas, el score, el high score, etc.
- System: Dentro de esta carpeta se encuentran los sistemas de físicas, renderizado y colisiones.
- Archivo utils.s: Contiene funciones de operaciones comunes como la multiplicación, división y módulo.
- Archivo cpct_functions: Archivo donde se declaran todas las funciones de cpctlera utilizadas en el juego. El resto de archivos solo hace include de este envés de cada función de cpctlera que necesite.

Physics system

Este sistema se encarga del movimiento de la pelota y también de obtener el input del teclado para el movimiento de la barra. Cabe destacar que se implementó aritmética en coma fija para hacer que la pelota tuviera ángulos de rebote más variados. Tanto la posición de la pelota como su velocidad en cada momento se guardan en forma de 2 bytes con uno representando la parte entera y otro la parte decimal. Además el sistema de físicas trabaja en conjunto con el sistema de colisiones para implementar el rebote de la pelota al detectar una colisión.

Collision system

En cuanto a las colisiones, solamente hay un elemento que va a colisionar contra otros: la pelota. Este sistema es quien detecta colisiones entre la pelota y los diferentes objetos como los bordes, bloques y la barra.

Render system

Este sistema utiliza funciones de CPCTelera para dibujar y borrar sprites de la pantalla. Implementa movimiento pixel a pixel para la pelota.

Game manager

Aquí se controla el flujo del juego. Hace las llamadas a los managers y systems correspondientes en cada ocasión. Por ejemplo para actualizar las posiciones de los objetos, dibujar en pantalla los sprites en las nuevas posiciones, etc. También contiene funciones que se utilizan en otros lugares como la función "man_game_block_collision" en donde se definen las acciones a realizar cuando la pelota colisiona con un bloque.

Entity manager

En este manager es donde se almacenan y gestionan las entidades de cada nivel, que son los bloques.

Aquí se definen también macros para la creación de los bloques de los 3 tipos.

Todos los bloques tienen las siguientes propiedades:

- tipo
- coordenada x
- coordenada y
- durabilidad
- color

Los únicos bloques que utilizan la durabilidad son los de plata pues pueden ser configurados para que se rompan con distinto número de golpes. Los bloques

comunes pueden ser de varios colores por lo que se utiliza la propiedad color para indicar cual sprite hay que dibujar.

Game over screen manager y start screen manager

En estos managers es donde se muestra la pantalla de fin o inicio del juego respectivamente y donde se espera el input del usuario para continuar. En la pantalla de inicio se muestra el título del juego y los controles. En la pantalla de fin se muestra un texto mostrando si el jugador ha ganado o si se ha terminado el juego por quedarse sin vidas. También se muestra el score y el high score.

Scoreboard manager

Aquí se gestiona la actualización del score, high score, el nivel actual y las vidas restantes en el scoreboard, que se encuentra en la parte derecha de la pantalla.

Stage manager

En este manager se declaran los diferentes niveles del juego. El array "man_stage_stages_array" guarda punteros a los stages. Dentro de los stages se guarda el número a mostrar en el scoreboard en la parte de "Level". Se guarda también el número de bloques destruibles en el nivel, el array con los bloques del nivel, el tamaño del array y un array con las probabilidades que tiene el jugador de obtener cada power up cuando se rompe un bloque. El tener un array para las probabilidades para cada nivel permite configurar que powerups se pueden obtener y con qué frecuencia en cada nivel. Debido a la falta de tiempo los powerups no se implementaron.

Problemas encontrados y soluciones

Uno de los desafíos técnicos más notables durante el desarrollo de Balkanoid fue la implementación del movimiento de la pelota. Inicialmente, se utilizaban valores enteros para representar la posición y la velocidad de la pelota, lo que resultaba en un movimiento restringido a diagonales perfectas. Esta limitación afectaba la jugabilidad y reducía la diversidad de interacciones dentro del juego.

Para abordar este problema, se optó por implementar aritmética en coma fija, utilizando dos bytes para representar la posición y la velocidad de la pelota; uno para la parte entera y otro para la parte decimal. Este formato permitió una representación más precisa del movimiento, multiplicando el valor por 256 para realizar las operaciones necesarias.

Sin embargo, al introducir ángulos variados con la aritmética de coma fija, se observó que el movimiento de la pelota se veía escalonado. La raíz de este issue residía en cómo el eje X representaba el movimiento cada dos píxeles, debido a la

forma en que se programa en Z80 y con el framework CPCTelera. Esta representación resultaba en una transición brusca de la pelota a través de los píxeles, lo que interfería con la fluidez del juego.

Para rectificar este aspecto y lograr un movimiento más fluido, se implementó un sistema de movimiento pixel a pixel. Esta solución implicó la utilización de dos sprites para la pelota, con uno de ellos desplazado un píxel a la derecha. Al alternar entre estos dos sprites, se logró una transición suave y continua del movimiento de la pelota en la pantalla, mejorando significativamente la jugabilidad y la estética del juego.

Imágenes y Música

Las imágenes se encuentran en assets/sprites y las música en assets/music. La conversión es llevada a cabo por cpctelera. Los sprites se realizaron usando la herramienta online "Piskel", mientras que la para la música se usó "Arkos Tracker 1".

Diseño de Sprites

El diseño de sprites en Balkanoid ha sido una labor meticulosa y creativa, centrada en la integración armoniosa de elementos culturales balcánicos dentro de la estética del juego. Todos los sprites fueron diseñados desde cero, lo que nos permitió tener un control total sobre la coherencia visual y temática del juego. A continuación, se detallan los sprites más significativos y sus respectivas inspiraciones y realizaciones:

- **Bordes de las Pantallas:**
 - Los bordes que enmarcan las distintas pantallas del juego representan un motivo tradicional balcánico. La elección de este diseño específico se hizo con la intención de sumergir al jugador en la temática desde el primer momento, creando un marco visual que refleja la riqueza estética de la región.
- **Título del Juego:**
 - En la pantalla de inicio, el título de Balkanoid se diseñó con colores distintivos de la cultura balcánica.
- **Pantalla de Game Over:**
 - La pantalla de Game Over introduce una figura mítica de la región balcánica, un Strigoi. Con el texto "THE STRIGOI HAVE EATEN YOU", buscamos añadir una capa de narrativa regional que enriquece la experiencia de juego con un toque de folklore local. La representación visual del Strigoi fue diseñada para introducir un elemento de sorpresa y desafío.
- **Pantalla de Victoria:**
 - Al ganar el juego, los jugadores son recibidos con una pantalla de victoria que celebra su logro con el texto "YOU WIN!", enmarcado con decoraciones y motivos balcánicos. Este diseño no solo sirve como una recompensa visual, sino que también refleja la amalgama de la tradición balcánica y la mecánica de juego que Balkanoid busca representar.

Cada sprite fue creado con una atención meticulosa al detalle, asegurando que la estética balcánica estuviera bien representada y se integrara de manera fluida con las mecánicas y la narrativa del juego. Esta fusión de diseño visual y cultural es fundamental para lograr la experiencia única que Balkanoid ofrece.

Música

La música juega un papel crucial en Balkanoid, sumergiendo a los jugadores en una atmósfera balcánica evocativa desde el primer momento. La selección musical se hizo con una profunda consideración de la autenticidad cultural y la cohesión temática.

- **Pantalla Principal:**

- La canción que adorna la pantalla principal es una adaptación de una melodía popular de la región de Dobrogea, Rumanía. Esta elección no fue casual; buscábamos un tema musical que resonara con la riqueza cultural de los Balcanes y que, al mismo tiempo, estableciera el tono para la aventura que aguarda al jugador. La adaptación se realizó con respeto hacia la composición original, manteniendo su esencia mientras se ajustaba para encajar en el contexto del juego.

- **Pantalla de Victoria:**

- Al alcanzar la victoria o derrota, los jugadores son recibidos con los acordes de "Kaval Sviri", una popular melodía búlgara. Esta pieza fue elegida por su ritmo siniestro que se convierte paulatinamente en alegre, con la intención de inspirar en el jugador las ganas de comenzar una nueva partida.

- **Sonidos FX:**

- Se incorporaron una serie de efectos especiales en forma de sonidos para dotar al rebote de la pelota de un feedback agradable al jugador. Hay 3 sonidos distintivos que suenan cuando:
 - La pelota rebota en la pala del jugador
 - La pelota rebota en los bordes de la pantalla
 - La pelota rebota en un bloque

La incorporación de estas piezas musicales y sonidos especiales no solo enriquece la experiencia de juego, sino que también celebra y da vida a la diversidad cultural de la región balcánica. El diseño sonoro de Balkanoid es un homenaje a la música balcánica, sirviendo como un puente entre la tradición musical de la región y la experiencia de juego. A través de la música, Balkanoid invita a los jugadores a una travesía que es tan cultural como lúdica, proporcionando una capa adicional de inmersión y conexión emocional.

El proceso de integración musical implicó una consideración de cómo cada pieza complementa la estética visual y la narrativa del juego, así como la técnica de adaptación a través de "Arkos Tracker 1" para asegurar una reproducción fiel y de alta calidad en la plataforma Amstrad CPC.

Conclusiones y Futuras Mejoras

Reflexiones

El desarrollo de Balkanoid ha sido una travesía enriquecedora y desafiante que nos ha permitido profundizar en nuestra comprensión del desarrollo de videojuegos en plataformas con recursos limitados. La programación en ensamblador Z80 para la máquina Amstrad CPC, con sus instrucciones restringidas y escasa memoria disponible, ha sido una experiencia educativa invaluable. Nos enfrentamos a retos únicos que exigieron un nivel de optimización y eficiencia que no solo mejoraron nuestra destreza técnica, sino que también nos proporcionaron una apreciación profunda de las capacidades y limitaciones inherentes a la programación a nivel de ensamblador.

A lo largo del proyecto, hemos adquirido una comprensión más profunda de cómo las restricciones de hardware pueden moldear y, en última instancia, enriquecer la creatividad y la innovación en el diseño de videojuegos. Las soluciones implementadas para superar los desafíos técnicos, especialmente en lo que respecta al movimiento de la pelota y la implementación de la música y el diseño visual, son testimonio de la capacidad de innovar dentro de las limitaciones.

Mejoras Futuras

En cuanto a mejoras futuras, vemos un horizonte prometedor para Balkanoid. Algunas áreas de enfoque incluyen:

- **Expansión del Contenido:**
 - Ampliación de los niveles del juego, incorporando más variedad en los desafíos y en las interacciones con los objetos dentro del juego. Un ejemplo puede ser la incorporación de power-ups o incluso de enemigos.
- **Optimización del Rendimiento:**
 - Continuar con la optimización del código para mejorar el rendimiento y la fluidez del juego, explorando técnicas avanzadas de programación en ensamblador.
- **Integración de Sonido Mejorada:**
 - Explorar la posibilidad de integrar más elementos sonoros que enriquezcan la atmósfera del juego, manteniendo la temática balcánica. Por ejemplo se puede hacer que cada color de bloque produzca un sonido distinto al ser golpeado. Además se puede introducir nueva música para reproducirla durante o entre los niveles.

- **Interfaz de Usuario Mejorada:**

- Desarrollar una interfaz de usuario más completa y estéticamente agradable, que permita una mejor navegación y experiencia de usuario. Por ejemplo se puede introducir la opción de pausar el juego y un menú para seleccionar el nivel.

Balkanoid sirve como un testimonio de cómo la pasión por la cultura y la programación pueden converger para crear una experiencia de juego única y memorable.

Anexos

Código Fuente

El código fuente se puede encontrar en la página del juego:

<https://cassettepixelcrafters.itch.io/balkanoid>

Recursos Gráficos y de Audio

Junto con el código fuente se encuentran los sprites y la música.