

CI-0119 Proyecto Integrador de Arquitectura y Ensamblador Alienígenas Alineados

Integrantes:

- Nathalia Maria Chaves Oviedo C22195
- Lisa Chen Tang C02253
- Emanuel García Rojas C23166
- Kevin Ramirez Barboza C26240

Introducción

Este proyecto consiste en la programación y elaboración de un juego al estilo "Match-3". Este tipo de juego se puede definir como un rompecabezas o "puzzle" que tiene el objetivo de formar combinaciones de tres o más elementos de una misma característica, usualmente esta característica es el color. Algunos ejemplos de estos juegos son: el conocido Candy Crush, Bejeweled, Shariki, entre otros.



Los juegos mencionados tienen la siguiente jugabilidad que consiste de un tablero matricial donde en cada casilla se encuentra un elemento, los elementos pueden tener distintos colores y formas. Cuando tres o más elementos adyacentes de una misma característica, en estos casos y para el proyecto será el color, se alineen de forma que creen una combinación (explicadas más adelante), son removidos del tablero. Posteriormente, su espacio es ocupado por otros elementos que caen por gravedad.

En cada turno el jugador intercambia dos elementos vecinos, con el fin de crear una combinación de elementos adyacentes del mismo color. Su objetivo es conseguir eliminar tantos elementos del tablero como le sea posible.

Descripción del problema

Elementos gráficos del problema:

Como fue mencionado anteriormente, el juego a programar consiste de diferentes elementos visuales que van a ser necesarios para su elaboración. Para la elaboración de toda la parte gráfica se utilizará el modelo, vista y controlador con C++ y la librería SFML.

El **menú** del juego contiene:

- El título del juego
- Un texto que indique que al presionar 'X' en el teclado inicia el juego, lleva al jugador a la ventana de juego
- Texto que indica que al presionar la flecha derecha lleva al jugador a la pantalla de créditos donde aparecen los nombres de los integrantes del grupo
- Y por último un texto que indique que al presionar la flecha izquierda se mostrará un breve resumen de las instrucciones del juego

La idea es que al presionar la flecha izquierda o derecha se haga una animación de slide a la pantalla correspondiente. Agregando una animación suave de transición para que la pantalla de inicio se deslice hacia fuera o se desvanezca mientras la pantalla de instrucciones/créditos se deslice o aparezca en su lugar. Esto proporciona una sensación más agradable para el usuario.

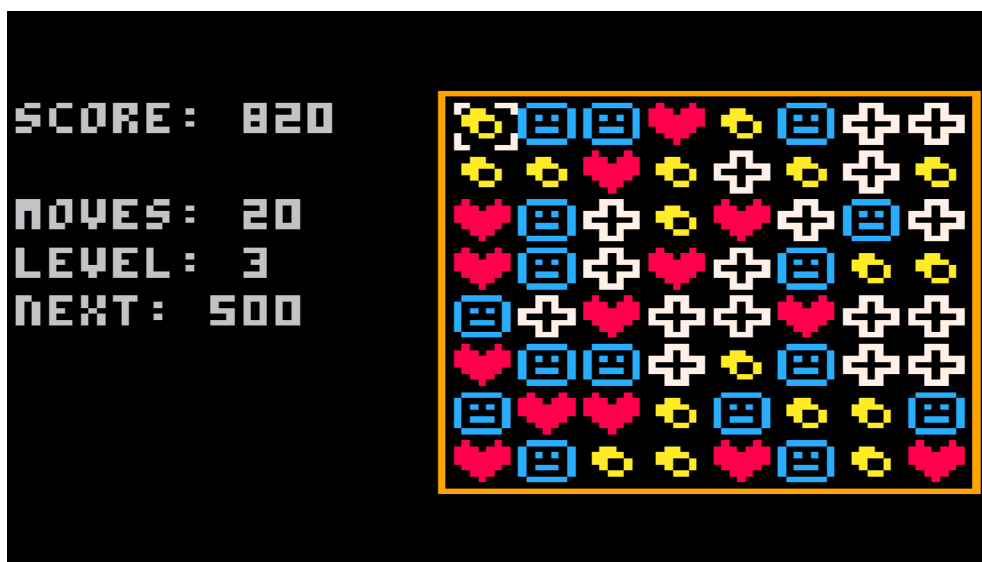
Ventana de juego:

El principal elemento es la ventana emergente con la información que necesitará el usuario para poder realizar su juego. Esta tiene que tener un tamaño específico para que pueda ser visible en cualquier computadora, es decir que no sea dependiente de un solo monitor, por lo que su tamaño se definirá posteriormente.

Además debe tener cierto tipo de elementos como:

- El **tablero** donde se encontrarán los iconos, será una matriz de 9x9 elementos
- El texto con la información de la ronda, se mostrará la puntuación total del jugador, el nivel actual, los movimientos restantes y la cantidad de minutos para alcanzar el siguiente nivel.

A continuación un ejemplo de cómo se piensa implementar:





Iconos:

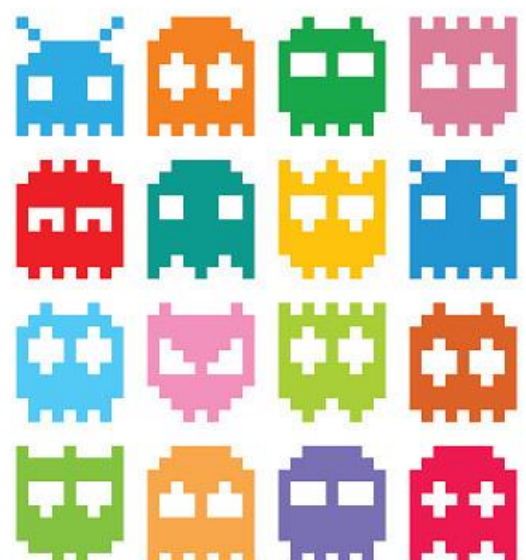
Por su parte los iconos que se piensan incluir tendrán una elaboración al estilo pixelArt, las cuales encajan con la estética del juego. A su vez, los elementos o iconos podrán tener alguno de los siguientes 6 colores diferentes:

- Rojo
- Naranja
- Verde
- Azul
- Morado
- Amarillo

Los iconos además de tener colores diferentes, van a tener formas/diseños distintos, esto para que el juego sea más accesible en caso de que algún usuario no logre distinguir bien entre los colores; para que aún pueda lograr combinaciones basándose en las formas.

Cabe destacar que a una forma se le va a asignar un color, por lo que las combinaciones pueden ser basadas en color o forma, al final va a ser lo mismo, solo son 2 maneras diferentes de distinguir entre elementos.

A continuación un ejemplo del diseño para los iconos:

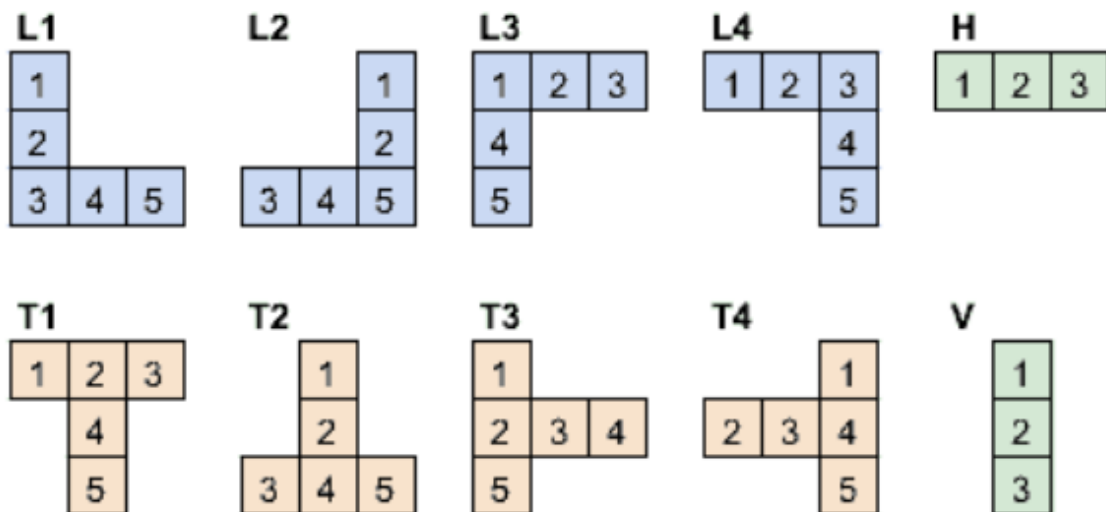


Aparte de estos elementos gráficos se piensan incluir **elementos sonoros** al igual que **animaciones** para que el jugador tenga una experiencia placentera al realizar una combinación exitosa.

Mecánicas del juego

Combinaciones Posibles:

En un juego tradicional de Match-3 se tiene una pequeña cantidad de combinaciones, éstas siendo las tradicionales tres en línea ya sea horizontal o vertical. En el juego que se tiene pensado realizar se tendrán las siguientes combinaciones, además de la opción que se tengan 3 o más de forma horizontal o vertical:



Al lograr una combinación válida los elementos involucrados en la combinación desaparecen y se le suman puntos al jugador, al desaparecer otros caen y toman su lugar.

Las combinaciones tienen orden de prioridad, si al mover un elemento se forman 2 combinaciones se aplica el orden de prioridad para escoger cuál combinación eliminar.

1. Vertical de 5 o más
2. Horizontal de 5 o más
3. Forma de L
4. Forma de T
5. Vertical de 4
6. Horizontal de 4
7. Vertical de 3
8. Horizontal de 3

Controles:

Para el juego se quiere que el usuario pueda navegar todo el menú y jugar con uso de únicamente el teclado.

- Tecla 'X': para cambiar 2 elementos de posición.
 - Si no hay elementos seleccionados y se presiona la tecla 'X' se selecciona tal elemento

- Si ya hay un elemento seleccionado y se presiona 'X' en un elemento adyacente a él y que al cambiarlos hace una combinación válida, ambos elementos cambian de lugar.
- Flecha arriba, abajo, izquierda, derecha: son utilizadas para moverse de casilla en casilla a la dirección correspondiente.

Game Over:

El juego empieza en el nivel 1, con cierta cantidad de movimientos y una puntuación a la que tiene que llegar para seguir a la próxima ronda/nivel. Al subir de nivel se le agregan más movimientos al jugador. Se pierde el juego si el jugador se queda sin movimientos y no logra pasar al siguiente nivel. El objetivo del juego es llegar a la puntuación más alta.

Alcances y limitaciones

Alcances:

La optativa de una programación híbrida que involucra la combinación de C++ y ensamblador nos brinda una serie de ventajas significativas en el desarrollo de nuestro proyecto. Por un lado, el uso de ensamblador para la lógica del juego permite un control preciso sobre las operaciones de bajo nivel. Esto es esencial para mantener un rendimiento óptimo del juego, ya que las operaciones pueden ser implementadas directamente en ensamblador para maximizar la velocidad de ejecución. Por otro lado, la elección de C++ para el desarrollo de la parte gráfica del juego facilita la creación de interfaces de usuario atractivas y funcionales. SFML y C++ proporcionan bibliotecas que simplifican la implementación de elementos visuales, animaciones y efectos. Esto nos permite centrarnos en la creatividad visual, sin tener que preocuparnos por los detalles de bajo nivel.

Limitaciones

Elementos especiales, los cuales se generan al hacer combinaciones de 4 o más elementos del mismo color. Estos, al ser eliminados crean reacciones especiales.

Decidimos mover los elementos especiales a la parte de limitaciones, para enfocarnos en una versión más simple del juego y centrarnos en aspectos de diseño y satisfacción del usuario. Para responder a problemas de accesibilidad, diseño, sonido y animación.

Como un extra se estaría implementando que los elementos tendrán la posibilidad de tener características que causen un efecto en el tablero al ser eliminados. Estas características son las siguientes:

- Elemento regular: no causa cambios en el tablero.
- Elemento vertical: elimina todos los elementos en esa vertical.
- Elemento horizontal: elimina todos los elementos en esa horizontal.
- Elemento cajón: elimina 8 adyacentes.
- Bomba de color: elimina todos los elementos del tablero de ese color.

Requisitos Funcionales

1. Lógica del Juego en **Ensamblador**: La lógica principal del juego será implementada en ensamblador, lo que permitirá un control preciso sobre las operaciones de bajo nivel y garantizará un rendimiento óptimo. Los siguientes componentes se desarrollarán en ensamblador:
 - Funcionamiento base de la matriz.
 - Generación y disposición inicial de elementos en la cuadrícula.
 - Lógica de todas las combinaciones entre los elementos.
 - Los contadores de movimientos que se han hecho.
 - Validación de movimientos legales.
 - Detección y eliminación de combinaciones de elementos.
 - Actualización de la cuadrícula después de cada movimiento.
 - Gestión de puntos y niveles: El nivel actual en el que se encuentra el usuario y la puntuación.

2. Parte Gráfica en Lenguaje de **Alto Nivel**: La parte gráfica del juego se implementará en un lenguaje de alto nivel para facilitar el diseño de interfaces de usuario atractivas y funcionales. Nosotros para este proyecto, pensamos utilizar el lenguaje de programación C++, y entre las posibles partes que pensamos que se pueden implementar en este lenguaje de alto nivel, están principalmente el apartado visual y gráfico del juego. Los siguientes aspectos serán manejados en este lenguaje:
 - Creación de la interfaz de usuario, incluyendo el tablero de juego y los elementos visuales.
 - Animaciones de movimientos de elementos y efectos visuales.
 - Menú, fin de juego, puntuaciones y créditos.
 - Interacción del Usuario: Los jugadores podrán interactuar con el juego de las siguientes maneras:
 - Seleccionar un elemento e intercambiarlo con uno adyacente.
 - Realizar combinaciones de tres o más elementos del mismo tipo.
 - Visualizar su puntuación y nivel actual.
 - Iniciar un nuevo juego.
 - Mostrar todos los objetos del Match-3.
 - Menú principal, el cual contará con diversas opciones en las que el usuario podrá elegir lo que desee hacer (salir, ver récords, etc...).
 - Efectos de sonidos.

Referencias

Proyecto Match-3: <https://jeisson.ecci.ucr.ac.cr/progra1/2022b/proyectos/>

Match-3

SFML:

https://www.youtube.com/watch?v=YNXrFOynalE&list=PLB_ibvUSN7mzUffhiay5g5GUHyJRQ4DYr&index=10

SFML: <https://www.geeksforgeeks.org/sfml-graphics-library-quick-tutorial/>

Transición entre pantallas :
<https://gamedev.stackexchange.com/questions/13454/how-to-implement-transition-between-screens>