Universidad La Salle Noroeste

Proyecto Final
Lucky Monkey

Materia: **Programación Orientada a Objetos**
Maestro: **Jesús Francisco Caro Cota**

Integrantes del equipo:

209264 | Cuevas Arce Alejo de Jesus
209268 | Felix Talamante Brunne
209405 | Valenzuela Díaz Agustín Eduardo
209089 | Villa Miranda Daffne Carolina

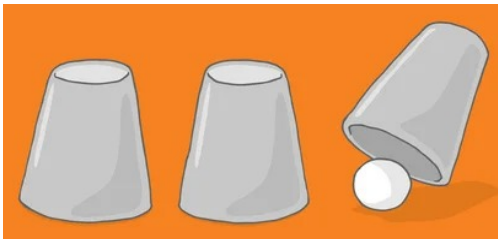
07 de diciembre del 2021

Introducción

El videojuego que nosotros realizamos está totalmente basado en C++. A parte de lo aprendido en clase, hicimos nuestra propia investigación para poder obtener un mejor resultado a la hora de tener nuestro videojuego terminado.

Nuestro propósito con este videojuego es aplicar lo aprendido en clase, para así tener un mejor conocimiento y que en un futuro podamos seguir aplicándolo.

Buscamos hacer un concepto algo tradicional para honrar a los videojuegos clásicos como ridge racer pero en concepto de Pixel Art como la imagen que estamos presentando.



Los juegos de suerte es algo que le gustan a la mayoría de las personas, al igual que el estilo de juego de azar que logrará dar una sensación de nostalgia recordando juegos que han sido parte de infancia de muchas personas.

Utilizamos C++ porque aparte de que es el lenguaje de programación que se utilizó durante el semestre, es uno de los lenguajes de programación más flexibles al momento de crear videojuegos de estilo de azar.

Con C++ se nos permite a los usuarios tener un mayor control sobre el hardware, la gestión de la memoria y los gráficos, lo que lo hace perfecto para la creación de videojuegos. C++ siempre se incluye en las mejores listas de programación para el desarrollo de videojuegos. Cuando se usa de manera eficiente, el lenguaje tiene un tiempo de ejecución súper rápido, lo cual es crítico cuando se trata de videojuegos.

Antecedentes

Todo tiene un comienzo, y por extraño que parezca, esos juegos de millones de polígonos y texturas increíbles a los que estás acostumbrado a jugar en flamantes consolas de nueva generación también lo tuvieron. Desde el primer videojuego (Pong) el mundo ha cambiado mucho. Tanto los gráficos como el sonido, así como el argumento de los videojuegos actuales han evolucionado de forma sorprendentemente rápida y de una manera sensacional, y aunque se puede decir que el primer videojuego apareció hace unos 50 años, los orígenes de este mundo fantástico se remontan mucho más atrás, al igual que las compañías que lo fundaron.

La historia de los videojuegos tiene su origen en la década de 1950 cuando, tras el fin de la Segunda Guerra Mundial, las potencias vencedoras de la guerra, construyeron los primeros superordenadores programables.nota 1 Los primeros intentos por implementar programas de carácter lúdico (inicialmente programas de ajedrez) no tardaron en aparecer, y se fueron repitiendo durante las siguientes décadas. Los primeros videojuegos modernos aparecieron en la década de los 60, y desde entonces el mundo de los videojuegos no ha dejado de crecer y desarrollarse con el único límite que le ha impuesto la creatividad de los desarrolladores y la evolución tecnológica. En los últimos años, se asiste a una era de progreso tecnológico dominada por una industria que promueve un modelo de consumo rápido donde las nuevas superproducciones quedan obsoletas en pocos meses, pero donde a la vez un grupo de personas e instituciones —conscientes del papel que los programas pioneros, las compañías que definieron el mercado y los grandes visionarios tuvieron en el desarrollo de dicha industria— han iniciado el estudio formal de la historia de los videojuegos.

Donkey Kong, de Shigeru Miyamoto (1981), uno de los videojuegos más populares de todos los tiempos.
El más inmediato reflejo de la popularidad que ha alcanzado el mundo de los videojuegos en las sociedades contemporáneas lo constituye una industria que da empleo a 120,000 personas y que genera unos beneficios multimillonarios que se incrementan año tras año. El impacto que supuso la aparición del mundo de los videojuegos significó una revolución cuyas implicaciones sociales, psicológicas y culturales constituyen el objeto de estudio de toda una nueva generación de investigadores sociales que están abordando el nuevo fenómeno desde una perspectiva interdisciplinar, haciendo uso de metodologías de investigación tan diversas como las específicas de la antropología cultural, la inteligencia artificial, la teoría de la comunicación, la economía o la estética, entre otras. Al igual que ocurriera con el cine y la televisión, el videojuego ha logrado alcanzar en apenas medio siglo de historia el estatus de medio artístico, y semejante logro no ha tenido

lugar sin una transformación y evolución constante del concepto mismo de videojuego y de su aceptación. Nacido como un experimento en el ámbito académico, logró establecerse como un producto de consumo de masas en tan solo diez años, ejerciendo un formidable impacto en las nuevas generaciones que veían los videojuegos con un novedoso medio audiovisual que les permitiría protagonizar en adelante sus propias historias.

Nosotros nos basamos en distintos videojuegos de azar y probabilidad. Uno de ellos es el famoso juego llamado “juego de los trileros”, o más conocido como “¿dónde está la bolita?”.

El trile es un juego en el que una persona, conocida como 'trilero', utiliza tres pequeños cubiletes y una bolita. Los que apuestan deben adivinar dónde está la bolita en uno de los distintos vasos; todo dependerá de la suerte que se tenga al momento de escoger el vaso que se cree que es el ganador.

Con el auge del juego competitivo, la suerte en los videojuegos se ha puesto en el punto de mira de aquellos que no quieren poner su habilidad en manos de la fortuna.

Sin embargo, como en casi todo lo lúdico, la suerte es una parte fundamental de por qué nos gustan los videojuegos. Incluso para aquellos que siguen pensando que la habilidad marca cada uno de sus pasos en un mundo virtual, la suerte es lo que les separa de estar jugando una eterna y aburrida partida de ajedrez.

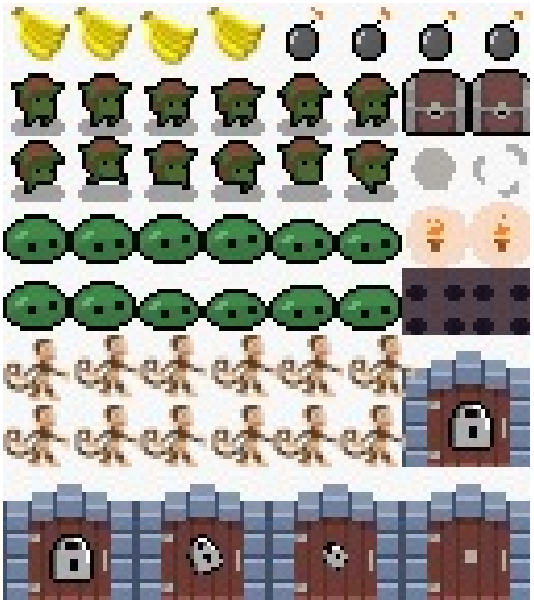
Las principales razones de por que a los seres humanos nos encantan los juegos de azar y suerte son las siguientes:

- Ego: “Si gano, gano por mi habilidad, y si pierdo, es que he tenido mala suerte”. La suerte puede darte o quitarte una victoria, pero también sirve de excusa para una mejora de ego capaz de evitar la frustración.
- Más jugadores: Si todos los juegos se basan únicamente en la habilidad sin tener en cuenta la suerte, el número de jugadores que podrían seguir sumándose al juego como principiantes cada vez sería menor y, a la larga, la posibilidad de encontrar a alguien con quien jugar acabaría desapareciendo.
- Motivación: “Esta vez es la buena” es el empujón emocional que nos invita a seguir jugando pese a que nuestra habilidad sea inferior a la requerida. Gozar de un golpe de suerte que acomode una victoria.
- Variedad: Está la posibilidad de vivir una situación distinta. Fortuito o casual, si todo sale tal y como estaba planeado el juego acaba siendo más un aburrido rompecabezas que una experiencia emocionante.

Desarrollo del proyecto

Nos pusimos a dialogar sobre de qué podríamos hacerlo y tomando en cuenta lo visto en clases y la manera en la que queríamos proyectar nuestra día decidimos realizar un concepto que ejecute los conceptos básicos de las físicas, como el movimiento, la suerte, etc.

Ya que supimos sobre el tema del videojuego que haríamos, empezamos a analizar lo que contendría. Primero, realizamos bocetos para elegir un estilo en específico, uno al que a todos nos convenciera, de ahí, ya que todos estuvimos de acuerdo, lo pasamos a illustrator, para así, obtener los sprites.




LUCKY MONKEY
"Luck is the tenacity of purpose"
START
TRY AGAIN

Funcionalidad de la CLI
Es un tipo de interfaz de usuario de computadora que permite a los usuarios dar instrucciones a algún programa informático o al sistema operativo por medio de una línea de texto simple. Debe notarse que los conceptos de CLI, shell y emulador de terminal no son lo mismo ya que CLI se refiere al paradigma, mientras que un shell o un emulador de terminal son programas informáticos específicos, que usualmente en conjunto implementan la CLI. Sin embargo, los tres suelen utilizarse como sinónimos.

En su forma más simple, una CLI consiste en un espacio donde se pueden escribir órdenes (por lo general, señalizado con un prompt). El usuario teclea una orden y la ejecuta al pasar a la línea siguiente, utilizando la tecla Entrar.

Al finalizar y enviar la orden con la tecla Entrar, un módulo interpretador de órdenes analiza la secuencia de caracteres recibida y, si la sintaxis de la orden es correcta, ejecuta la orden dentro del contexto del programa o del sistema operativo donde se encuentra. Esta forma de trabajo es secuencial, y equivale a un tipo de programación paso a paso.

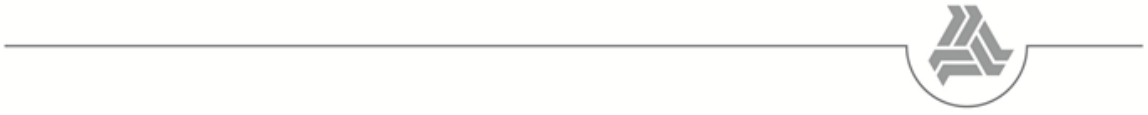
Funcionalidad del sistema de inputs
Input: Son las señales electrónicas que son enviadas hacia un destino de ingreso determinado, para su debido procesamiento. En primer lugar, el usuario utiliza un Periférico de Entrada (input)para poder, mediante la transmisión de impulsos eléctricos, ingresar un dato dentro del sistema.

Cuando el Periférico de Entrada (input) esté realizando esta acción, también tendrá una acción de Salida de Datos, es decir, comenzará a enviar impulsos eléctricos hacia un dispositivo específico dentro del sistema (en primer lugar, el Procesador).

Esta señal es Almacenada, Ordenada, Procesada y posteriormente es enviada hacia los Periféricos de Salida (output) mediante la misma vía de impulsos eléctricos, para que el usuario pueda percibir a través de sus sentidos la acción que está realizando, o bien qué es lo que se ha procesado.

Conceptos vistos en clase:

- Simple and Fast Multimedia Library (SFML) es una API portable, escrita en C++ pero también disponible en C, Python, Ruby, OCaml y D. Su propósito principal es ofrecer una biblioteca alternativa a la biblioteca SDL, usando un enfoque orientado a objetos.



Gracias a sus numerosos módulos, SFML puede ser usada como un sistema mínimo de ventanas para interactuar con OpenGL o como una biblioteca multimedia cuyas funcionalidades permiten al usuario crear videojuegos y programas interactivos.

- Un Rigidbody es el componente principal que permite el comportamiento físico para un objeto. Con un Rigidbody adjunto, el objeto inmediatamente responderá a la gravedad. Si uno o más componentes Collider son también agregados entonces el objeto será movido por colisiones entrantes.

  Debido a que un componente Rigidbody asume el movimiento del objeto el cual está adjunto, usted no debería intentar moverlo desde un script cambiando las propiedades del Transform como lo son la posición y rotación. En vez, usted debería aplicar forces para empujar el objeto y permitirle al motor de física calcular los resultados.

- Los GameObjects son objetos fundamentales que representan personajes, props, y el escenario. Estos no logran nada por sí mismos pero funcionan como contenedoras para Components, que implementan la verdadera funcionalidad.

- Box2D es una biblioteca libre que implementa un motor físico en dos dimensiones. Está programada en C++ por Erin Catto, y se distribuye bajo la licencia zlib.

- La Detección de Colisiones es el método utilizado por algunos videojuegos para detectar si dos objetos (Sprites) han colisionado. La detección puede ser por área o píxel a píxel:
  Por área: los objetos ocupan un área, rectangular o circular, cuando dos de estas áreas se superponen hay una colisión.

  Píxel a Píxel: los objetos ocupan un área rectangular, pero tienen una máscara que define qué píxeles son visibles. Primero se realiza una detección de colisión de área, luego, si hubo colisión, se realiza una detección píxel a píxel entre los píxeles superpuestos de ambos objetos. Si existen dos píxeles superpuestos, y ambos son visibles, entonces hay una colisión.

- La clase Drawable es una abstracción que representa “algo que puede ser dibujado”. Esta clase se extiende para definir gran variedad de objetos gráficos más específicos.

- Un asset es una representación de cualquier ítem que puede ser utilizado en su juego o proyecto. Un asset podría venir de un archivo creado afuera de cualquier programa, tal como un modelo 3D, un archivo de audio, una imagen, o cualquiera de los otros tipos de archivos que el programa soporte.

Conclusiones

Nuestra conclusión es que la creación de un videojuego es un proceso largo y complicado, se le tiene que dedicar mucho tiempo para obtener resultados satisfactorios y si se busca superar las expectativas de la audiencia a la que va dirigida.

Se habla en general (por todos los miembros del equipo) que las expectativas sobre la creación de nuestro videojuego no eran las más altas, ya que somos principiantes en el mundo de la programación, sin embargo nos ha dejado más que satisfechos, es algo que pudiéramos hacer en alguna parte de nuestra vida.

En la mayoría de los integrantes este proyecto ha despertado el gusto a la programación y que podamos seguir con expectativas de poder crear grandes cosas para nuestra carrera.

Desarrollando nuestro proyecto dinámico, hemos aprendido muchas cosas, un ejemplo sería las funciones, dependencias, librerías, etc., es algo complejo para un principiante. No importa el lenguaje que se utilice, si es Javascript, Python, Java o cualquier otro lenguaje para un videojuego, es algo demasiado complejo y que se necesita un tiempo de planificación y paciencia, ya que un solo error en una línea de código puede causar muchos errores en otras líneas de código.

El uso de videojuegos es algo importante en el desarrollo de las personas ya que mejora las capacidades cognitivas del cerebro. ejemplos de las capacidades que se pueden perfeccionar podrían ser:

- Atención
- Orientación
- Velocidad de procesamiento
- Praxias

Referencias

Ernesto Pereiras García. (2020). *Monografias.com: Historia de los videojuegos*. Recuperado de https://www.monografias.com/trabajos90/historia-videojuegos/historia-videojuegos.shtml

Wikipedia. (2021). *Trile*. Recuperado de https://es.wikipedia.org/wiki/Trile

Angela Judith Chacon. (2021). *¿Cómo es el juego de los trileros?* .Recuperado de https://aleph.org.mx/como-es-el-juego-de-los-trileros

Wikipedia. (2020). *Historia de los videojuegos*. Recuperado de https://es.wikipedia.org/wiki/Historia_de_los_videojuegos

Linda I. Olivares Flores. (2008). *UNAM: Manual de Programación en el Lenguaje de C++.* Recuperado de https://paginas.matem.unam.mx/pderbf/images/mprogintc++.pdf

Creative Commons Attribution. (2020). *Programming in C++.* Recuperado de https://www.whitman.edu/mathematics/c++_online/c++.pdf

CampusMVP. (2019). *Los mejores lenguajes de programación para el desarrollo de videojuegos.* Recuperado de https://www.campusmvp.es/recursos/post/los-mejores-lenguajes-de-programacion-para-el-desarrollo-de-videojuegos.aspx#:~:text=en%20el%20desarrollo.-,C%2B%2B,cuando%20se%20trata%20de%20videojuegos

Razón artificial. (2012). *Guía para aprender a programar videojuegos con C++*. Recuperado de http://razonartificial.com/2012/02/guia-aprender-programar-videojuegos-con-cpp/

Ricardo González. (2013). *No sólo ingeniería: Cómo crear un juego en C++ desde cero*. Recuperado de https://nosoloingenieria.com/juego_ordenador_c_basic/

Enrique Aguirre. (2014). *Desarrollo de videojuegos paso a paso con C++ y SFML*. Recuperado de https://intelligenciavirtual.blogspot.com/2014/03/programacion-de-juegos-con-codeblocks-y.html

Rubén Velasco. (2021). *Qué lenguajes de programación se usan para hacer videojuegos.* Recuperado de https://www.softzone.es/programas/lenguajes/programacion-videojuegos/

Curso de C++ Builder. (2020). *Programación Orientada a Objetos en C++*. Recuperado de https://elvex.ugr.es/decsai/builder/intro/5.html

Wikipedia. (2021). *Command-line interface*. Recuperado de https://es.wikipedia.org/wiki/Interfaz_de_l%C3%ADnea_de_comandos#Funcionamiento

Sistemas. (2021). *Input/Output.* Recuperado de https://sistemas.com/inputoutput.php

GameObject. (2020). *Manual*. Recuperado de https://docs.unity3d.com/es/2018.4/Manual/class-GameObject.html

Wikipedia. (2021). *SFML*. Recuperado de https://es.wikipedia.org/wiki/SFML

Tutorialesprogramación. (2018). E*structura y nombres de archivos y carpetas de un proyecto Angular - Carpeta 'assets'.* Recuperado de https://www.tutorialesprogramacionya.com/angularya/detalleconcepto.php?punto=60&codigo=60&inicio=40

Wikipedia. (2021). *Box2D*. Recuperado de https://es.wikipedia.org/wiki/Box2D

Documentation. (2016). *Carpeta audioclip*. Recuperado de https://docs.unity3d.com/es/530/Manual/class-AudioClip.html

Wikipedia. (2021). *Colisiones*. Recuperado de https://es.wikipedia.org/wiki/Detecci%C3%B3n_de_colisiones

UPV. (2021). *Drawable*. Recuperado de http://www.androidcurso.com/index.php/recursos/35-unidad-4-graficos-en-android/136-drawable

EcuRed. (2021). *C++*. Recuperado de https://www.ecured.cu/C%2B%2B

Con Clase. (2021). *C++ con clase*. Recuperado de http://conclase.net/c

HektorDocs. (2021). 12 frameworks y librerías para desarrollar juegos. Recuperado de https://docs.hektorprofe.net/escueladevideojuegos/articulos/frameworks-librerias-bibliotecas-recopilacion/

Ángel Robledano. (2019). *Qué es C++.* Recuperado de https://openwebinars.net/blog/que-es-cpp/

Lluis Gil Espert. (2018). *El C++ por la práctica: Introducción al lenguaje y su filosofía*. Recuperado de https://upcommons.upc.edu/bitstream/handle/2099.3/36408/9788483013380.pdf

Lenguajesdeprogramación. (2020). *C++*. Recuperado de https://lenguajesdeprogramacion.net/cpp/

Bitbrain. (2021). *Qué es la estimulación cognitiva y para qué sirve*. Recuperado de https://www.bitbrain.com/es/blog/que-es-estimulacion-cognitiva

Pablo Novara. (2021). *Fundamentos de programación*. Recuperado de http://zinjai.sourceforge.net/Anexo1.pdf

Lukas Dürrenberger. (2021). *2D Graphics with SFML*. Recuperado de https://meetingcpp.com/mcpp/slides/2018/2D%20Graphics%20with%20SFML.pdf

Microsoft. (2021). *Recursos para crear un juego de C++*. Recuperado de https://docs.microsoft.com/es-es/cpp/windows/resources-for-creating-a-game-using-directx?view=msvc-170

Lindsay Schardon. (2020). *Drawing Sprites with SFML & C++*. Recuperado de https://gamedevacademy.org/sfml-cpp-sprites-tutorial/

Lucasderego. (2015). *C++ and SFML game tutorial*. Recuperado de https://lucasderego.wordpress.com/category/c-and-sfml-game-tutorial/