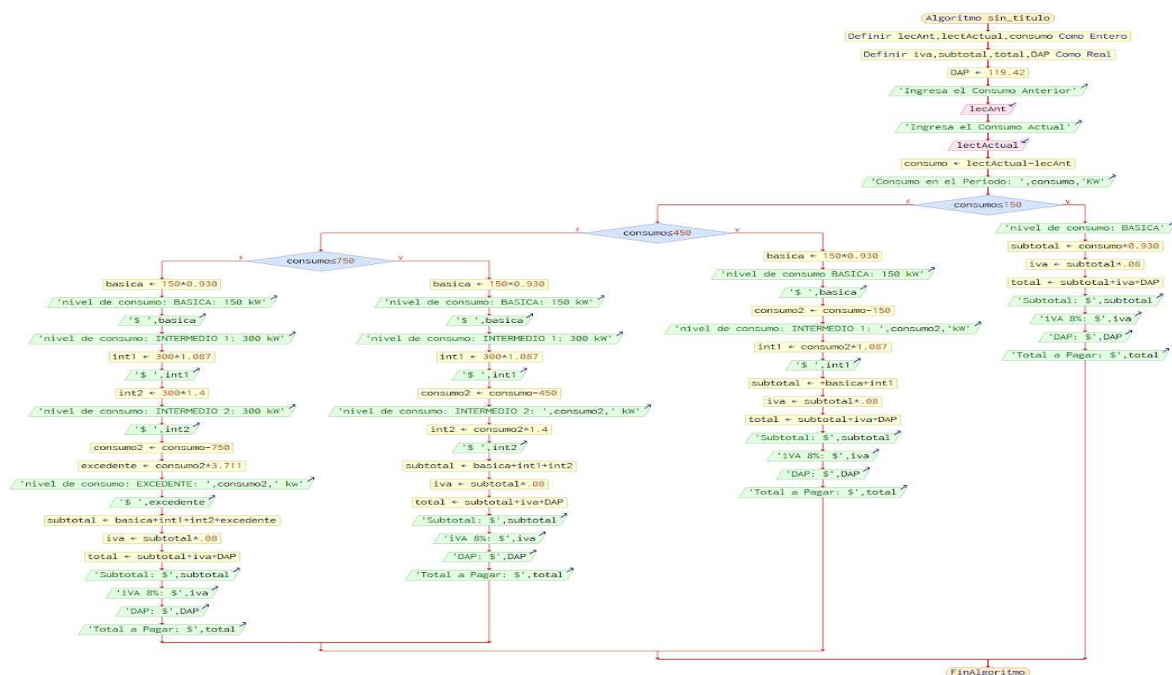
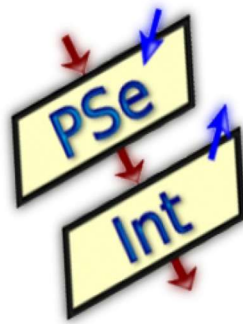




ModeX: Bitácora del Proyecto

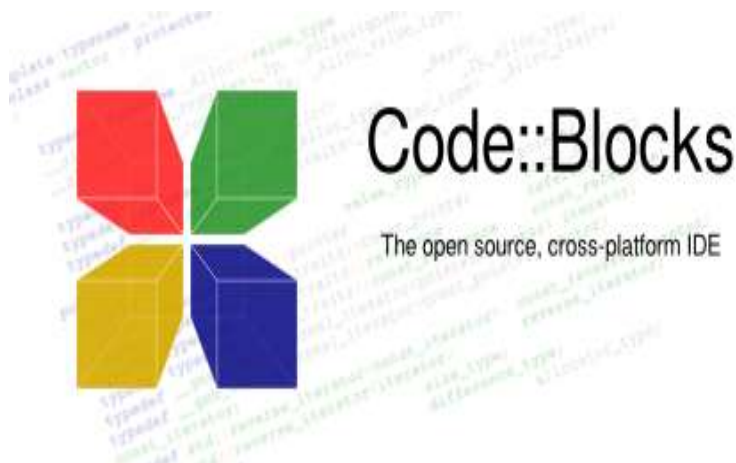
24 de febrero de 2025 - PseInt

Se inició con la introducción a PseInt, analizando sus bases, su forma de uso y su estructura. PseInt es una herramienta visual y lógica que permite la comprensión y entendimiento de algoritmos, todo esto mediante diagramas de flujo o pseudocódigo, las características de esta herramienta nos permitieron fortalecer la lógica de programación de manera clara y sencilla. Los conocimientos adquiridos en PseInt fueron fundamentales para después trasladar esa lógica al desarrollo de ModeX mostrando que las ideas y algoritmos pueden adaptarse a diferentes entornos de programación.



3 de marzo de 2025 - Introducción al lenguaje de programación: C++

En esta fecha iniciamos con el aprendizaje del primer lenguaje de programación visto: C++, trabajando en el IDE Code::Blocks. Comprendimos que es un lenguaje multiplataforma y compilado utilizado para desarrollar aplicaciones de alto rendimiento y que su enfoque orientado a objetos permite organizar y reutilizar el código de manera eficiente, nos basamos en estas características del lenguaje C++ para los patrones de diseño que se implementaron en el desarrollo de ModeX, además de entender posteriormente la diferencia entre un lenguaje compilado a uno interpretado, logrando así detectar las optimizaciones de rendimiento más eficientes para el lenguaje de Python, aplicándolas para poder lograr un rendimiento óptimo en el videojuego.



ModeX: Bitácora del Proyecto

10 de marzo de 2025 - Primer programa en un lenguaje de programación (C++)

Realizamos nuestra primera actividad en C++, la cual consistió en un cuestionario. Esta actividad nos ayudó a comprender mejor la sintaxis y estructura de un lenguaje de programación, permitiéndonos así aplicar los conocimientos y comandos más comunes, como el output y el input, en el futuro desarrollo del proyecto ModeX, así como a transformar la lógica en código funcional. Este aprendizaje fue importante para prepararnos en el uso de otros lenguajes de programación, como Python en el futuro desarrollo de ModeX, permitiendo así el aprendizaje rápido y eficiente de la sintaxis del lenguaje.

18 de marzo de 2025 - Tipos de datos base en programación

En esta fecha aprendimos los diferentes tipos de datos en C++, comprendiendo su importancia para almacenar y manejar información dentro de un programa. El conocimiento de estos tipos de datos nos permitió entender mejor cómo se estructura la información dentro de un programa y cómo elegir correctamente el tipo según el dato que se necesite almacenar, lo que serviría para decisiones importantes de estructuras de datos rápidas en optimizaciones de rendimiento para el proyecto ModeX. Además, al haber visto conceptos similares en otros lenguajes de programación, el aprendizaje de Python, que fue el lenguaje base de nuestro proyecto, fuera más rápido y sencillo, ya que solo fue necesario asimilar la sintaxis de C++ a la de Python en lugar de comenzar desde cero.

24 de marzo de 2025 - Actividad de Suma de Valores (C++)

Para esta actividad se nos dio la instrucción de analizar la suma de dos valores. Para hacerlo debíamos escribir un algoritmo en el cuaderno que se encargará de sumar ambos y mostrar el resultado una vez que un usuario haya ingresado los dos números. Este ejercicio en particular fue la primera vez que se entendió y comprendió lo que es una operación y un operador en un lenguaje de programación, permitiéndonos en el futuro utilizar operaciones para cálculos matemáticos lo que fue esencial para campos como ángulos, velocidad y vectores de movimiento en el desarrollo de ModeX.

29 de marzo de 2025 - Actividad de Tabla de Multiplicar (C++)

En esta actividad se diseñó un algoritmo para imprimir en consola la tabla de multiplicar del 7, el ejercicio se desarrolló en el lenguaje de programación C++. Esta actividad nos permitió entender el funcionamiento de ciclos en los lenguajes de programación, que facilitan la repetición de código por medio de estructuras repetitivas en vez de copiar el mismo código la cantidad que lo desees repetir. Además de que los bucles son, se podría decir con certeza, la estructura base de cualquier aplicación

ModeX: Bitácora del Proyecto

o videojuego, y tener el dominio de esta estructura es fundamental para poder implementar la funcionalidad de ModeX.

7 de abril de 2025 - Actividad de Media Aritmética (C++)

Para esta actividad se desarrolló un programa en C++ en el que se ingresa el nombre de un alumno, sus tres materias y la respectiva calificación de cada una, el programa calcula la media aritmética de la sumatoria de las tres materias. Esta actividad en particular nos ayudó a comprender la importancia de las medidas de tendencia central en el análisis y estadísticas de datos, estas medidas se utilizarían en el futuro para la elaboración de la documentación técnica del proyecto en donde se presentan datos de rendimiento, arquitectura de código y patrones de diseño específicos.

14 de abril de 2025 – Actividad C++ (Descuentos)

En esta actividad se desarrolló un programa en C++ enfocado en el cálculo de descuentos. Se aprendió a utilizar estructuras condicionales para aplicar diferentes porcentajes según ciertas condiciones. Esta práctica fue importante porque nos ayudó a trabajar con operaciones matemáticas, validaciones y toma de decisiones dentro del código. Estas bases fueron fundamentales para la documentación técnica del proyecto, en donde se presentan datos de rendimiento que se calculan mediante porcentajes, y para comprender cómo manejar datos y resultados dentro de videojuegos.

21 de abril de 2025 – Actividad C++ (Uso de switch)

Se desarrolló un programa en C++ utilizando la estructura condicional switch, donde se preguntaban las actividades más importantes de cada mes. Esta actividad nos ayudó a comprender mejor el uso de estructuras de selección múltiple y cómo organizar opciones dentro de un menú. Este aprendizaje fue muy útil para entender el funcionamiento de un menú y de las acciones en consecuencia de la selección de una opción dentro de un videojuego.

28 de abril de 2025 – Actividad Uso del Ciclo While (C++)

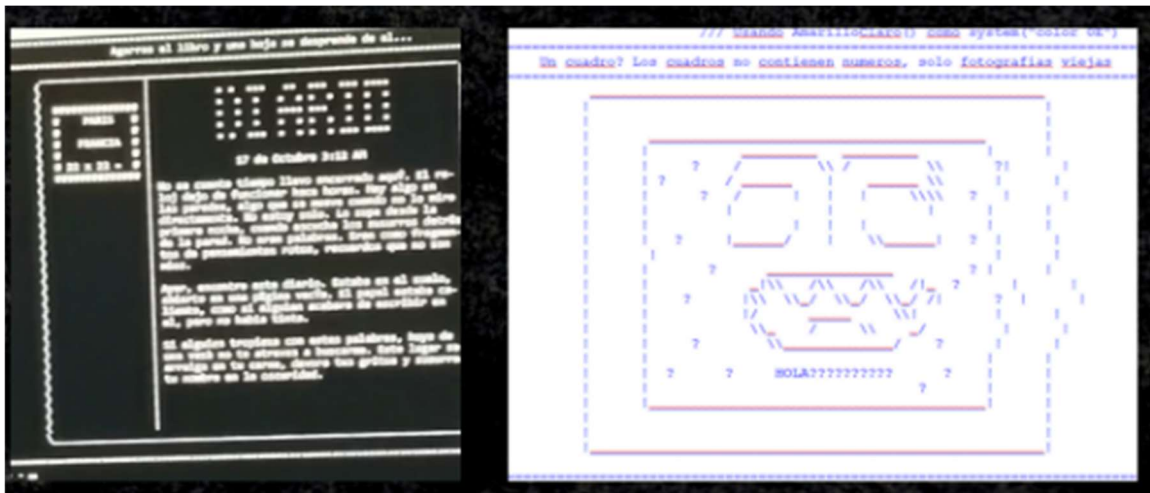
Se desarrolló un programa en C++ utilizando el ciclo while, lo que nos permitió comprender mejor los ciclos repetitivos y el control de ejecución. Se aprendió cómo repetir instrucciones hasta que se cumpliera una condición específica, cosa esencial en el desarrollo de videojuegos. En particular el uso del ciclo while es muy habitual en game loops utilizando la biblioteca Pygame, se utiliza este ciclo junto con una bandera booleana de estado del juego, por lo que para los juegos desarrollados en

ModeX: Bitácora del Proyecto

Pygame este conocimiento representó una base importante para el desarrollo de videojuegos con esta biblioteca.

29 de abril al 2 de junio de 2025 – Desarrollo del Primer Juego: Zhyniria

Durante este periodo se desarrolló el primer videojuego del equipo, Zhyniria, el cual nos permitió integrar diversos conocimientos de programación y organización. Este proyecto fue una práctica importante para aprender a estructurar un videojuego completo, manejar errores y mejorar la lógica del código. Todo este proceso sirvió como base para el desarrollo de ModeX, reafirmando que no es necesario contar con herramientas avanzadas o conocimientos muy complejos para crear proyectos funcionales.



6 de mayo de 2025 – Actividad Interpretar Código (C++)

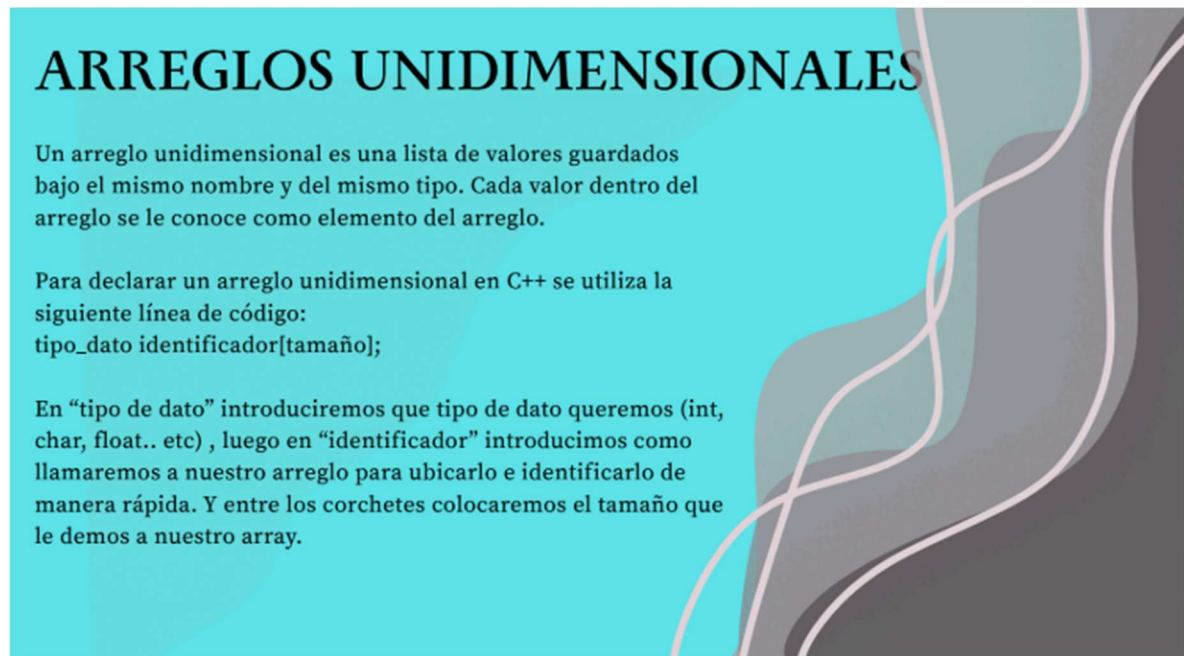
En esta actividad se analizó e interpretó un código fuente en el lenguaje de C++ proporcionado por la maestra. Esta práctica fue muy importante porque nos ayudó a desarrollar la capacidad de lectura y comprensión de programas ya estructurados. Entender código ajeno es una habilidad fundamental en programación y fue clave para mejorar nuestro nivel de comprensión y análisis antes de comenzar el desarrollo de ModeX, además de que durante el proyecto se trabajó en equipo, cada uno desarrollando diferentes partes del código, por lo que contar con esta habilidad de análisis facilitó el desarrollo rápido del videojuego.

13 de mayo de 2025 – Actividad C++ (Tipos de arreglos – Investigación)

Se realizó una investigación sobre los diferentes tipos de arreglos en C++, por lo que por primera vez se tuvo contacto con una estructura de datos y durante la investigación se aprendió cómo almacenar

ModeX: Bitácora del Proyecto

múltiples datos dentro de una sola estructura y cómo organizarlos eficientemente. Este conocimiento fortaleció nuestra comprensión sobre manejo de información y datos dentro de un programa, lo que fue indispensable para el desarrollo de ModeX.



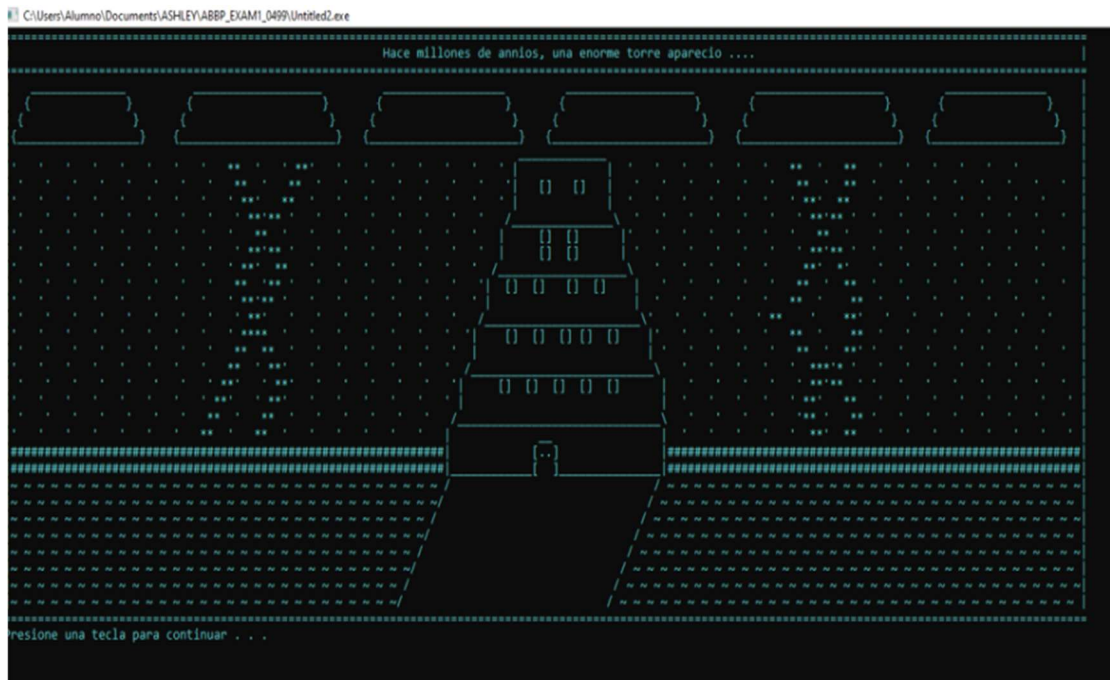
20 de mayo de 2025 – Actividad con Arreglos (C++)

Se aplicaron los conocimientos adquiridos sobre arreglos en una actividad práctica. Trabajamos con almacenamiento, recorrido y manipulación de datos dentro de estructuras organizadas. Esta práctica fue fundamental para desarrollar una mejor lógica de organización de información, habilidad que posteriormente resultó esencial en la creación del videojuego ModeX.

11 de junio de 2025 – Entrega del Proyecto: Juego “Zhyniria”

En esta fecha se realizó la entrega final del proyecto correspondiente a las fechas marcadas. Esta actividad nos permitió reflexionar sobre el proceso de aprendizaje, evaluar los resultados obtenidos, recibir feedback de playtesting y reconocer la importancia del trabajo constante. La experiencia adquirida en el desarrollo de videojuegos fue aplicada posteriormente en el desarrollo de ModeX, reforzando habilidades como la organización y estructuración de código.

ModeX: Bitácora del Proyecto



3 de septiembre de 2025 - Primera Invitación Formal al Concurso Local de Prototipos y Proyectos de Emprendimiento de la DGETI 2025

Recibimos la primera invitación para participar en el Concurso Local de Prototipos y Proyectos de Emprendimiento de la DGETI 2025, en la categoría de Prototipo de Software, lo que marcó el comienzo oficial del proyecto ModeX.

Esta convocatoria marcó el 25 de noviembre de 2025 como día del concurso en la fase local, dándonos un período inicial de aproximadamente once semanas para desarrollar, documentar y presentar el proyecto. Se decidió desarrollar un sistema arcade funcional que demostrara la viabilidad del desarrollo de videojuegos mediante Python sin el dominio de conocimientos complejos y uso de motores comerciales como proyecto para el concurso.

11 de septiembre de 2025 - Inicio del Desarrollo de ModeX Versión 1.0

Se dio inicio al desarrollo de ModeX Versión 1.0, teniendo como objetivo principal el desarrollo de un sistema de arcade que tuviera al menos dos videojuegos funcionales con mecánicas diferentes. Durante este período que duró hasta el 1 de octubre de 2025, el enfoque principal fue establecer la estructura base del proyecto para lograr escalabilidad a corto plazo y desarrollar una arquitectura de código sólida para el futuro, permitiendo evitar problemas de legibilidad o de organización con el código. Desde el principio se optó por documentar exhaustivamente con comentarios en el código que función cumple en el sistema de cada componente facilitando el entendimiento y la comprensión del código por cualquier integrante del equipo.

ModeX: Bitácora del Proyecto



12 de septiembre de 2025 - Implementación de la Metaclase Singleton

En esta fecha, se implementó la metaclase Singleton que garantiza la instancia única de las clases críticas del sistema. Esta implementación surgió como prevención al problema de que por error, en ocasiones, se instancian doble vez clases que estaban diseñadas para instanciación única.

Para poder tener el contexto de esta solución, es importante entender primero qué es el patrón Singleton. Se trata de un patrón de diseño para crear que restringe la instancia de una clase a un único objeto. En otras palabras, cuando se aplica este patrón, se asegura que sin importar cuántas veces intentemos crear un nuevo objeto de esa clase, siempre se devolverá la misma instancia. Esto es muy útil en escenarios donde necesitamos un punto de acceso global a un recurso compartido, como un administrador de configuración. En resumen, el patrón de diseño Singleton evita la duplicación de clases críticas y costosas que están diseñadas para que solo exista un objeto de esta clase.

15 de septiembre de 2025 - Integración de la Primera Fuente Tipográfica (PressStart2P)

Se descargó e integró la fuente de texto PressStart2P (diseño pixel-art retro compatible con estética arcade) como tipografía principal del sistema. El archivo de fuente se estructura en la carpeta de assets con el subdirectorio de fonts (assets/fonts), esta fuente fue el primer elemento de diseño (asset) descargado, formando así parte de la base del diseño inicial del proyecto.

Para dar más contexto, PressStart2P es una fuente tipográfica digital diseñada específicamente para emular la estética visual característica de los videojuegos arcade y consolas de 8-bits de la década

ModeX: Bitácora del Proyecto

de los ochenta. Se optó por esta fuente como la principal del sistema porque captura fielmente la apariencia pixelada de sistemas de visualización de baja resolución lo que asemeja a un arcade de décadas pasadas.

17 de septiembre de 2025 Creación de Archivo requirements.txt

Se creó el archivo requirements.txt documentando dependencias exactas del proyecto. Además, se realizó la búsqueda y obtención de imágenes de la portada y menú principal del arcade para su posterior implementación en el arcade.

El propósito de este archivo es listar las bibliotecas y paquetes externos necesarios para que el proyecto funcione correctamente, especificando las versiones exactas de cada dependencia, lo que permite a cualquier desarrollador reproducir el entorno de desarrollo idéntico simplemente ejecutando el comando `pip install -r requirements.txt`, lo cual instala automáticamente todas las librerías listadas con sus versiones precisas.

18 de septiembre de 2025 - Investigación de la API de la biblioteca Arcade Python

Se investigó extensivamente la documentación oficial de Arcade Python para familiarizarnos con su API, se analizó la información de la documentación obtenida del sitio oficial (<https://api.arcade.academy/en/stable/>) en la que encontramos información muy importante y base para lo que sería la estructura y arquitectura del proyecto en el futuro. Al analizar la documentación encontramos que la API de Arcade estaba orientada a objetos y que es muy fácil de entender y comprender por lo que ponerla en práctica no fue un impedimento para el proyecto.

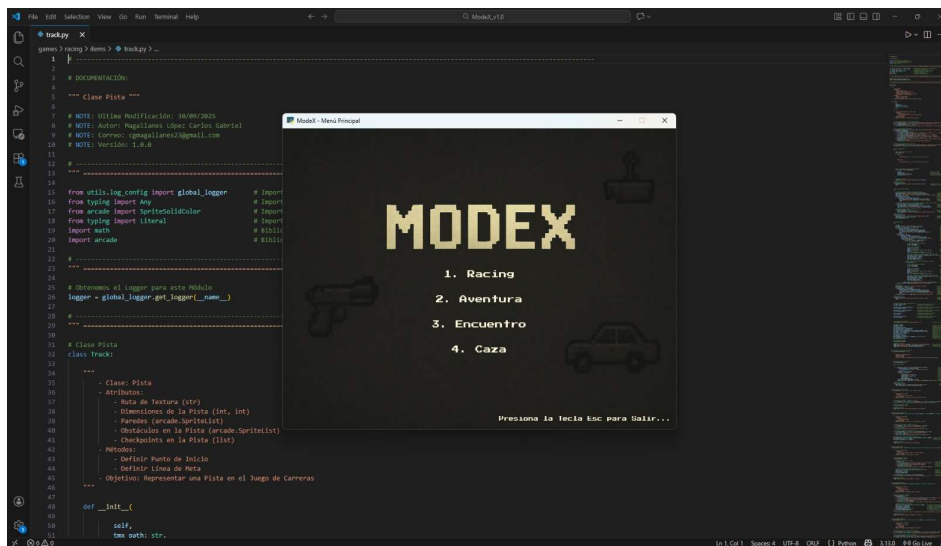
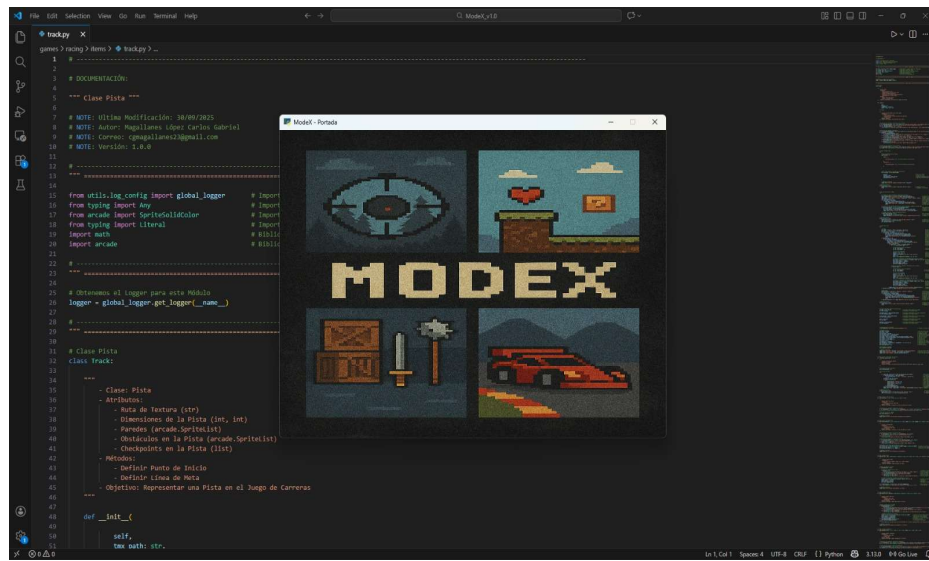
Durante la investigación se identificaron los componentes fundamentales de la biblioteca que serían críticos para nuestro desarrollo, incluyendo el sistema de Sprites y SpriteLists para gestión eficiente de entidades del juego, el manejo de física para juego de plataformas mediante la clase `PhysicsEnginePlatformer` para colisiones y gravedad realistas, el sistema de Views permite organizar el código en diferentes pantallas o estados del juego como menús y niveles, y también se prestó atención a las utilidades para carga y reproducción de sonidos y música.



ModeX: Bitácora del Proyecto

19 de septiembre de 2025 - Implementación de la Portada y Menú Principal

En esta fecha se implementó la portada y el menú principal del arcade con los assets gráficos obtenidos anteriormente. Para esta implementación se necesitó comprender los bucles que se necesitan para un videojuego, donde el método `on_draw()` se ejecuta continuamente para dibujar todos los elementos visuales en pantalla, mientras que `on_update(delta_time)` se encarga de actualizar la lógica del juego. Se implementó la clase `MenuView` heredando de `arcade.View`, organizando el código de manera modular y permitiendo transiciones fluidas entre diferentes pantallas del sistema. El evento `on_key_press()` se usó para detectar la presión de teclas específica. Este primer contacto con la biblioteca permitió establecer los conocimientos teóricos y técnicos adquiridos durante la investigación de la documentación.



ModeX: Bitácora del Proyecto

20 de septiembre de 2025 - Implementación del Sistema de Logging

Se implementó un sistema de logging estructurado para rastreo de eventos durante la ejecución del videojuego, el sistema cuenta con las siguientes características:

- Sistema creado a partir del módulo logging estándar de Python
- Cuenta con niveles jerárquicos:
 - DEBUG: Información detallada de Desarrollo.
 - INFO: Eventos normales.
 - WARNING: Situaciones no-críticas.
 - ERROR: Fallos recuperables.
 - CRITICAL: Errores fatales.
- Formato de mensaje incluyendo:
 - o Timestamp preciso
 - o Nivel de severidad
 - o Nombre de módulo origen
 - o Mensaje descriptivo

Además, el Sistema de logging cuenta con rotación automática de archivos cuando el tamaño excede 5 MB previniendo crecimiento indefinido.

21 de septiembre de 2025 - Implementación del Módulo de Utilidades

Se implementó el módulo de utilidades ubicado en el directorio `utils/` proporcionando funcionalidades extras para el sistema de arcade. Específicamente se desarrolló el módulo `imgs.py` que proporciona utilidades de dibujo como `draw_neon_text` para el dibujo de textos neón.

El módulo de utilidades (`utils`) es un componente de arquitectura fundamental en el desarrollo de videojuegos por qué sirve como repositorio de funciones auxiliares, helpers y herramientas generales que son usadas a lo largo de todo el proyecto por diferentes sistemas y módulos.

23 de septiembre - Desarrollo de Juego de Carreras

The image displays two screenshots of a PyGame development environment, likely using PyCharm, showing the creation of a racing game menu.

Top Screenshot: Model - Runas

- The code editor on the left shows Python code for a PyGame project. It includes imports for pygame, sys, and random, and defines a class for a race track.
- The game window on the right, titled "Model - Runas", displays a blue car on a dark, stylized road. The text "MODEX NEON RUSH" is prominently displayed in a pink, neon-like font. Below the car, the text "PRESS ENTER TO START" is visible.

Bottom Screenshot: Model - Menu de Carreras

- The code editor on the left shows the same Python code as the top screenshot, but with the class definition for the race track.
- The game window on the right, titled "Model - Menu de Carreras", displays a red race car on a dark, stylized road. The number "1" is prominently displayed in a pink, neon-like font above the car.

ModeX: Bitácora del Proyecto

25 de septiembre de 2025 - Ampliación del Módulo de Utilidades

Se amplió el módulo de utilidades, con las siguientes mejoras:

- Desarrollo del módulo `utils/paths.py` con función `get_image_path()` resolviendo rutas absolutas a relativas mediante `os.path.abspath(os.path.join(base_dir, 'assets', 'images', relative_path))` donde `base_dir` se determina dinámicamente como directorio del archivo principal ejecutable evitando problemas cuando el usuario ejecuta el videojuego desde diferentes directorios de trabajo.
 - Desarrollo del módulo `detect_nose_angle.py` calculando el ángulo de orientación del sprite del carro mediante el análisis de píxeles: identifica la "nariz" del vehículo buscando píxel no-transparente más alejado del centro en todas las direcciones y calcula el vector desde el centro de sprite al píxel identificado.
-

30 de septiembre de 2025 - Implementación del Módulo de Recursos Globales del Juego de Carreras

Se implementó el módulo recursos globales del juego de carreras (`games/racing/resources/globals.py`) consolidando constantes compartidas en módulo `globals.py`: definiciones de colores RGB para elementos de UI, configuraciones de dificultad y otras configuraciones extras utilizadas como argumentos para funciones o inicializadores para clases. Además, se implementó el menú de selección de carreras pensado para la selección del carro que el usuario quiera utilizar para correr en las carreras. Se aplicaron tests generales a la estructura base del juego para validar su funcionamiento correcto durante el videojuego, finalmente se creó el archivo TODO como recordatorio para las nuevas features pendientes para la siguiente versión.

El archivo TODO se creó con el propósito de servir como lista de tareas pendientes y recordatorios de trabajo futuro en diferentes versiones del proyecto, por lo que el archivo funciona como una herramienta de gestión en donde se registran features no implementadas, bugs conocidos que requieren corrección, mejoras planificadas, optimizaciones pendientes, refactorizaciones necesarias y cualquier otra tarea identificada durante el desarrollo que no puede abordarse inmediatamente debido a prioridades o limitaciones de tiempo.

1 de octubre de 2025 - Implementación de las Clases Track y Car en el Juego de Carreras

Se implementaron las clases de objetos Track y Car con funcionalidad completa:

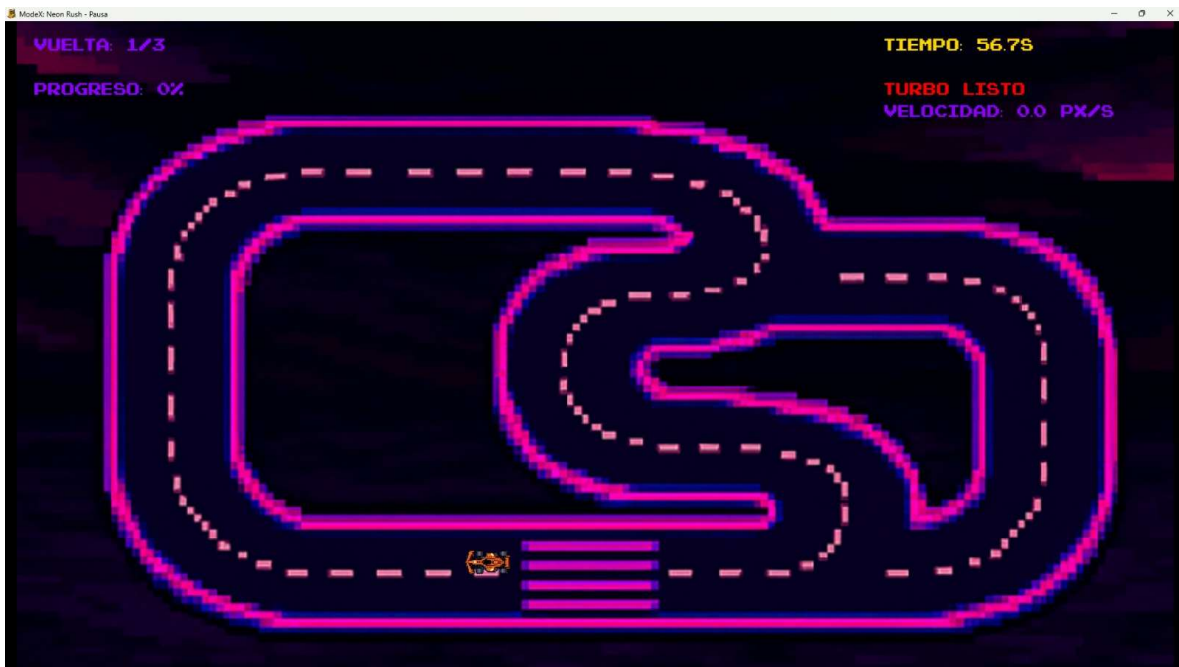
- Clase Track: esta clase representa una pista en el juego de carreras, cuenta con carga archivo TMX exportado desde Tiled Map Editor, capas de objetos de colisión (paredes como

ModeX: Bitácora del Proyecto

polígonos), atributos para posiciones de checkpoints circulares y línea de meta y detección de colisión con paredes con spatial hashing.

- Clase Car: esta clase representa un carro en el juego de carreras, implementa un modelo de física con aceleración constante al presionar tecla arriba, fricción aplicada cada frame reduciendo velocidad gradualmente, rotación proporcional a velocidad actual (simulación de inercia realista), velocidad máxima diferenciada (300 unidades normales, 450 durante turbo), sistema de turbo con impulso instantáneo al activar, duración de 2 segundos y cooldown de recarga de 5 segundos con visualización mediante texto de UI.

Además, se implementó el bucle principal del juego de carreras integrando todos los componentes: spawn de carro del jugador en posición inicial definida en pista, bucle de actualización ejecutando la lógica de física, (`car.update(delta_time)`) verificación de checkpoints y línea de meta, actualización de textos de UI (tiempo transcurrido, vuelta actual, velocidad instantánea, estado de turbo) y dibujado de todos los elementos en orden correcto (pista de fondo, carro, textos de UI).



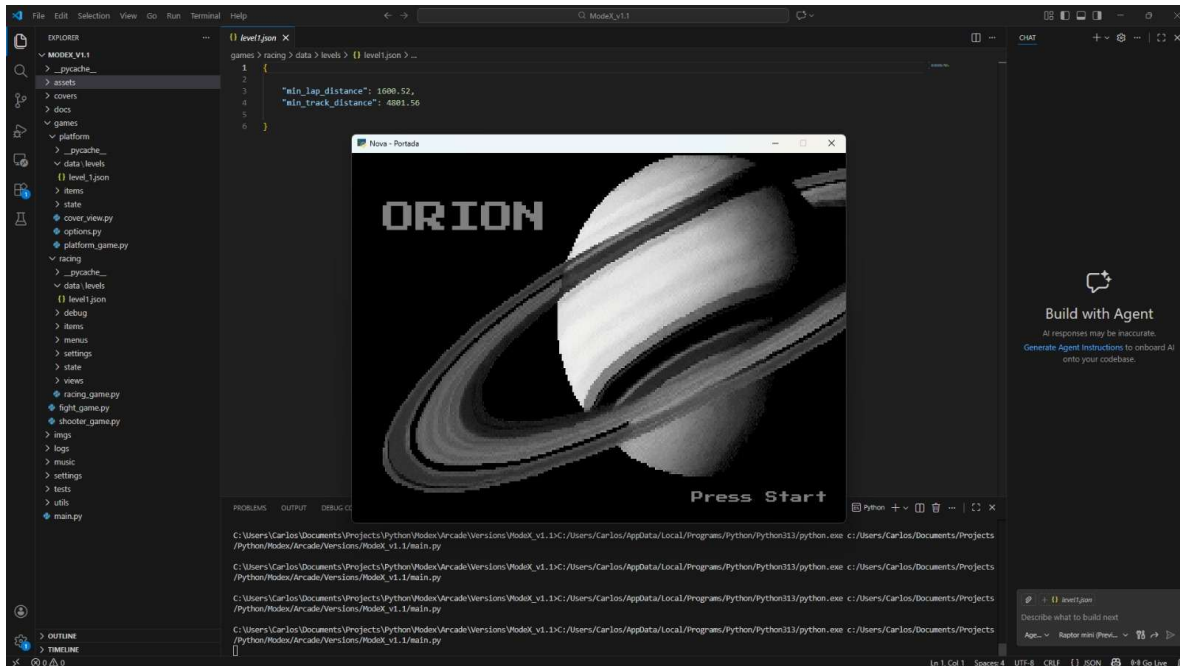
3 de octubre de 2025 - Inicio del Desarrollo de Modex Versión 1.1

Se inició con el desarrollo de ModeX Versión 1.1 ampliando el sistema con el segundo videojuego de género diferente: plataformas 2D de acción llamado "Cosmic Odyssey" con mecánicas de exploración, combate, y progresión por niveles. Este período que duró hasta el 3 de noviembre necesitó una arquitectura más compleja debido a mayor cantidad de sistemas que interactúan entre sí, como lo son:

- Física platformer con gravedad y detección de suelo.

ModeX: Bitácora del Proyecto

- Cámara que sigue al jugador con smoothing y límites del nivel.
- Gestión de múltiples niveles con carga/descarga dinámica.
- Sistema de enemigos con comportamiento de patrullaje.
- Sistema de combate con hitboxes temporales.



Entre el 3 y 6 de octubre de 2025 - Implementación de Clases Fundamentales del Juego de Plataformas

Se implementaron clases fundamentales del juego de plataformas:

- Clase Camera: representación de una cámara, usando un viewport que sigue al jugador, aplicando smoothing mediante interpolación lineal con movimiento suave sin saltos bruscos y restringiendo posición dentro de límites del nivel previniendo visualización de áreas fuera del mapa diseñado.
- Clase Level: representación de un nivel, cuenta con sus propios enemigos, plataformas y otras propiedades personalizadas como la música de fondo específica del nivel.
- Clase Player: representación del jugador heredando arcade.AnimatedWalkingSprite con sprites de animación para idle/walk/jump/attack/hurt, sistema de salud con puntos de vida actual máximos y métodos para recibir.
- Clase Wall: representación de las paredes del nivel mediante sprites invisibles posicionados con ajuste de hitbox rotada.

ModeX: Bitácora del Proyecto

Entre el 7 y 9 de octubre de 2025 - Desarrolló de Clases de Gestión de Estado en Juego de Plataformas

Se desarrollaron las clases de gestión de estado del juego de plataformas enfocándose en lógica de persistencia de datos:

- Clase GameState: mantiene el estado actual de partida en memoria.
 - Clase LevelManager: coordina las transiciones entre niveles mediante carga del siguiente nivel
 - Clase LevelCache: representando un caché de guardado de niveles para su futura carga.
-

8 de octubre de 2025 - Segunda Invitación Formal al Concurso Local de Prototipos y Proyectos de Emprendimiento de la DGETI 2025

Se recibió la segunda invitación formal para la convocatoria del concurso con recordatorio de fecha límite próxima, lo que motivó la aceleración del desarrollo de ModeX, enfocando nuestros esfuerzos en completar funcionalidad core crítica priorizando funciones visuales. Además en esta fecha se realizó el registro oficial del equipo por medio de un enlace de registro oficial del concurso.

9 de octubre de 2025 - Implementación de Dataclasses y Enums para el Juego de Plataformas

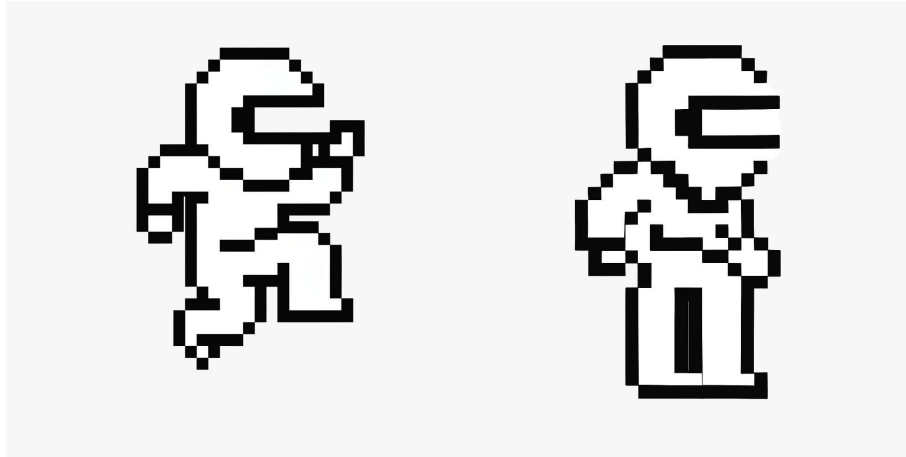
Se implementaron dataclasses y enums para el juego de plataformas utilizando decorador @dataclass de Python para estructuras de datos inmutables con validación de tipos:

- Clase LoadingStrategy: Enum definiendo estrategias de carga de niveles.
 - o LAZY: carga de nivel solo cuando accede.
 - o PRELOAD_ALL: Pre-carga de todos los niveles.
 - o HYBRID: carga del siguiente nivel durante gameplay.
 - o NONE: ninguna estrategia de carga de nivel.
 - Clase LevelStats: dataclass que almacena estadísticas de nivel (tiempo récord, muertes totales, enemigos derrotados y completado del nivel)
 - Clase PlayerProgress: dataclass guardando progresión global del jugador (niveles desbloqueados, logros obtenidos, puntuación acumulativa)
 - Clase SessionStats: dataclass agregando estadísticas de sesión actual (tiempo jugado, niveles completados en sesión, muertes en sesión).
-

ModeX: Bitácora del Proyecto

10 de octubre de 2025 - Búsqueda y Creación de Assets Gráficos para el Juego de Plataformas

Se realizó la búsqueda y creación de assets gráficos para el juego de plataformas recopilando sprites de personaje jugador con animaciones, fondos del nivel, plataformas y sprites de enemigos. No solo se obtuvieron o crearon assets gráficos, también se crearon y se buscaron assets de audio para fondo de música de niveles.



11 de octubre de 2025 - Primera Junta Formal para el Concurso Local de Prototipos y Proyectos de Emprendimiento de la DGETI 2025

El equipo asistió a la primera junta formal dedicada específicamente a la preparación para el concurso, en esta junta se habló de los requisitos para el concurso, elaboración de documentación y su respectiva estructura, cronograma detallado de actividades del concurso (25 de noviembre) y revisión de rúbrica de evaluación del concurso. Además, ese mismo día se implementaron los recursos globales y la portada del juego de plataformas, todo esto de la misma forma que con el juego de carreras.

12 de octubre de 2025 - Implementación de Clases de Estado en el Game Loop del Juego de Plataformas

Se implementaron e integraron de objetos y clases de estado al game loop del juego de plataformas:

- Inicialización de PhysicsEnginePlatformer de Arcade proporcionando detección de colisiones con plataformas y aplicación automática de gravedad.
- Instanciación de Camera con configuración de viewport inicial centrado en el spawn del jugador
- Bucle de actualización ejecutando operaciones en orden (procesamiento inputs, actualización de física, verificación de colisiones con proyectiles/ataques, actualización de animaciones, actualización de cámara, verificación de condiciones de victoria/derrota y actualización de textos de UI)

ModeX: Bitácora del Proyecto

- Bucle de renderizado dibujando capas en orden correcto (background, plataformas, enemigos, jugador, proyectiles y textos de UI).

Durante esta fecha también iniciamos una investigación de la documentación de Pygame explorando su API que se usaría en el futuro para el desarrollo de los otros videojuegos para explorar la diversidad de bibliotecas para videojuegos 2D.

13 de octubre de 2025 - Implementación de Sistema de Lectura y Carga de Archivos JSON en Juego de Plataformas

Se integró un sistema de lectura y carga de archivos JSON para configuración de niveles: estos se definen estructura mediante diccionario con las siguientes claves:

- music: nombre de archivo de música de fondo.
- scores: lista de puntuaciones del nivel.
- times: lista de tiempos del nivel.
- level_id: ID del nivel.
- dimensions: dimensiones del nivel.
- gravity: gravedad del nivel.
- player_spawn: posición del spawn del jugador.
- level_goal: posición de meta del nivel.
- backgrounds_paths: lista de rutas de los backgrounds.
- walls: datos para el inicializador de instancia de la Clase Wall.
- platforms: datos para el inicializador de instancia de la Clase Plataforma.
- enemies: datos para el inicializador de instancia de la Clase Enemies.

Además, se creó el primer archivo JSON completo para Nivel 1 del juego de plataformas definiéndolo con un nivel medio para permitir al usuario practicar y desarrollar sus habilidades. Al mismo tiempo se inició una búsqueda de assets gráficos para el tercer juego del sistema: luchador 2D de combate cooperativo.

14 de octubre de 2025 - Implementación de la Clase Wall para el Juego de Plataformas

Se implementó la clase Wall para el juego de carreras representando barreras de circuito con sprites rectangulares rotados: el constructor recibe longitud, grosor, ángulo de rotación y posición central; se genera un sprite invisible con hitbox ajustada mediante `arcade.hitbox.RotatableHitBox` permitiendo detección de colisiones precisa teniendo en cuenta su rotación.

ModeX: Bitácora del Proyecto

15 de octubre de 2025 - Búsqueda de Assets Gráficos para el Juego de Carros

Se realizó una búsqueda adicional de assets gráficos para el juego de carreras: sprites de carros adicionales con diferentes estéticas (carros deportivos) y elementos de decoración de pista (patrocinadores ficticios). Además, se integró un sistema de lectura y carga JSON para carga de niveles en este juego.



Entre el 15 y el 18 de octubre de 2025 - Implementación del Sistema de Carga de Assets para el Juego de Peleas

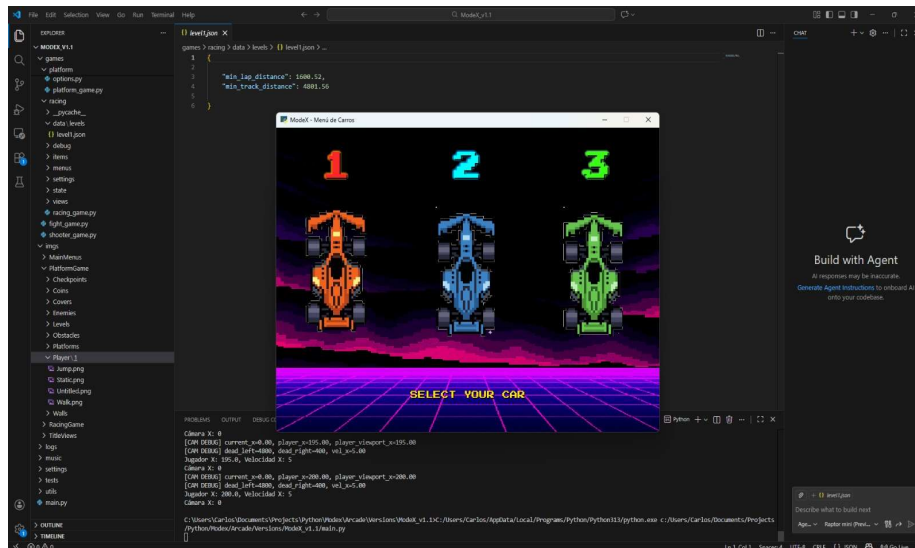
Entre estas fechas se implementó el sistema de carga de assets gráficos y de audio propio del juego de peleas. Lo que permitió la carga dinámica de recursos utilizados en el juego.



ModeX: Bitácora del Proyecto

16 de octubre de 2025 - Actualización del Menú Principal del Juego de Carreras

Se actualizó el menú principal del juego de carreras agregando los sprites de carros nuevo que se obtuvieron de la búsqueda de assets gráficos y se le agregó un texto parpadeante de instrucción, por lo que el menú quedó con 3 opciones diferentes de carros para su selección.



17 de octubre de 2025 - Implementación de la Clase Race Manager del Juego de Carreras

Se implementó la clase RaceManager que coordina lógica completa de carrera:

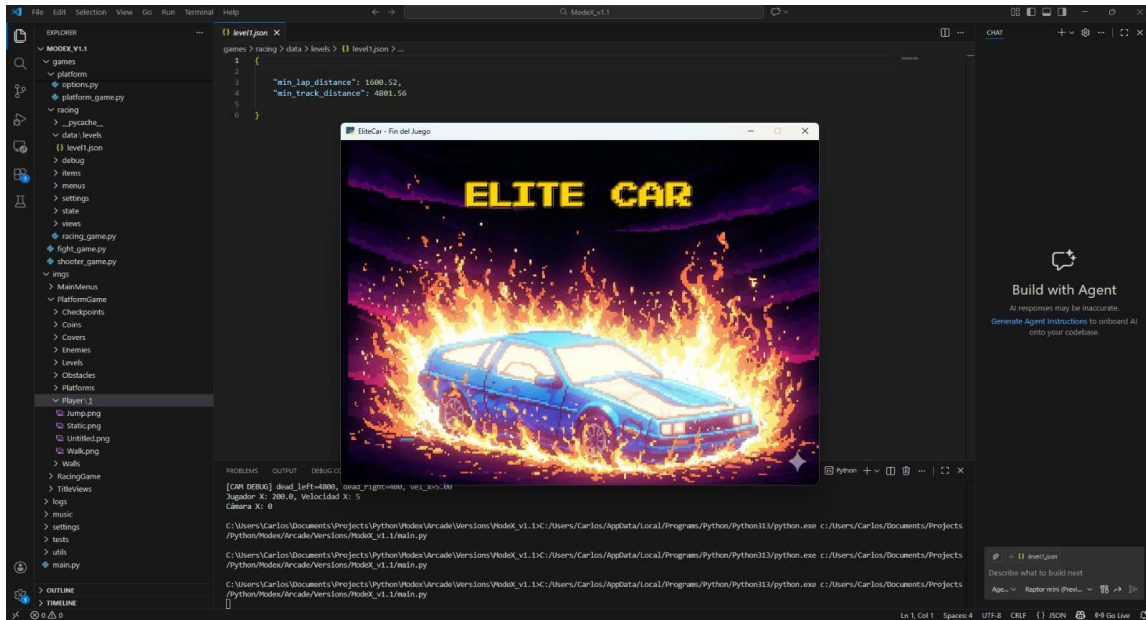
- Gestión de estados de carrera.
- Contador de vueltas: rastrea la vuelta actual y las vueltas restantes hasta completar la carrera.
- Sistema de checkpoints: verifica el paso secuencial correcto (el jugador debe pasar los checkpoints del nivel en orden específico antes de completar la vuelta lo que previene irregularidades del jugador si corta curvas para regresar al inicio).
- Temporizador de carrera: cuenta tiempo transcurrido con un sistema de límite de tiempo.
- Detección de cruce de línea de meta: considera dirección para el conteo de vuelta (el jugador debe entrar desde el lado trasero y salir por el lado delantero, para evitar activar si se cruza en dirección opuesta previniendo activaciones falsas)
- Gestión de música de fondo con inicio/pausa/resume sincronizado con estados de carrera.

Entre el 18 y 21 de octubre de 2025 - Implementación de Vistas Complementarias para el Juego de Carreras

ModeX: Bitácora del Proyecto

Se desarrollaron las vistas complementarias del juego de carreras proporcionando pantallas de transición:

- WinView: desplegada al completar totalmente el juego.
- GameOverView: mostrada al perder en una carrera.
- PauseView: accesible durante la carrera al pausar el nivel (presionar la tecla Enter).



Del 20-22 de octubre de 2025 - Actualización del Juego de Carreras para ModeX Versión 1.1

Se actualizó el juego de carreras para la Versión 1.1 integrando todos los componentes desarrollados hasta la fecha y puliendo detalles visuales y auditivos. Se implementaron transiciones suaves entre todas las pantallas, se probaron múltiples sesiones completas de carrera desde inicio hasta game over/victoria y se ajustó el balance de física del carro basándonos en feedback de playtesting interno (velocidad máxima aumentada 10% para mayor sensación de velocidad y fricción reducida 5% para derrapes más largos).

Del 22-24 de octubre de 2025 - Implementación de Clases de Objetos Fundamentales en el Juego de Peleas

Se implementaron clases de objetos fundamentales en el juego de peleas:

- Clase Actor representando personaje combatiente con animaciones, estadísticas de combate.
- Clase Player heredando Actor con manejo de inputs del jugador mapeando combinaciones de teclas a acciones de contexto e integración con sistema de cámara.

ModeX: Bitácora del Proyecto

- Clase Skeleton heredando Actor, casi misma estructura que la clase Player pero esta clase representa al jugador.

23 de octubre de 2025 - Implementación de Funciones Helper en Juego de Peleas

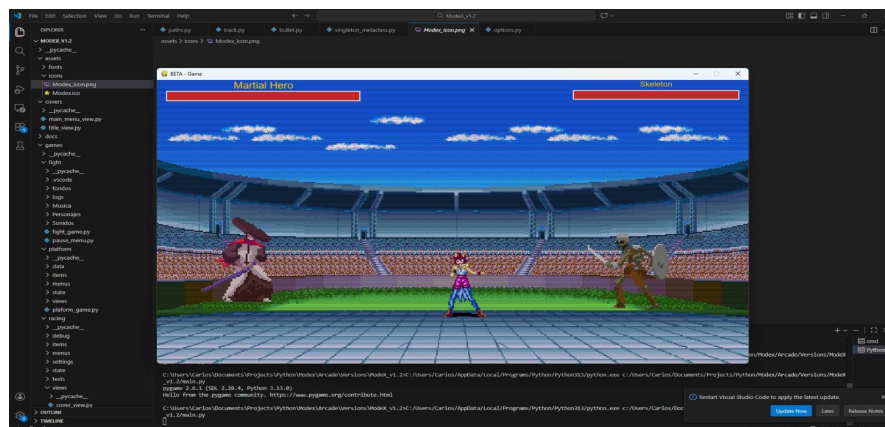
Se implementaron funciones helper para juego de peleas facilitando desarrollo y debugging:

- función `draw_debug()`: esta función dibuja información de estado de personaje como texto flotante sobre cabeza del personaje, vida restante y coordenadas de posición, toda esta información facilita el debugging.
- función `bottom_transparent_rows()`: esta función analiza el sprite del personaje identificando filas de píxeles inferiores completamente transparentes (usado en sprites que tengan la parte inferior está vacía), calculando el offset vertical para alinear los "pies" visuales del personaje con superficie de suelo previniendo apariencia de flotación o hundimiento, crítico para animaciones de idle/caminar donde contacto visual incorrecto con el suelo afecta dramáticamente la percepción de peso y presencia del personaje.

Entre el 24 y 31 de octubre de 2025 - Implementación de Bucle Principal del Juego de Peleas

Se implementó el bucle principal completo del juego de peleas integrando todos los sistemas:

- Inicialización de ventana Pygame (no Arcade, decisión técnica basada en mostrar la diversidad entre bibliotecas de desarrollo de videojuegos bidimensionales).
- Carga de stage de combate (imagen de fondo, plataformas de suelo, límites laterales del escenario)
- Instanciación de dos personajes combatientes (2 jugadores humanos).
- Inicialización de recursos de audio.
- Implementación del game loop procesando eventos de input con `pygame.event.get()` incluyendo eventos de teclado para controles de jugador y eventos de sistema, actualizando lógica de ambos combatientes.



ModeX: Bitácora del Proyecto

25 de octubre de 2025 - Integración del Sistema de Lectura y Carga de Archivos JSON en el Juego de Carreras

Se integró el sistema de lectura y carga JSON también en juego de carreras: este sistema define datos para la clase RaceManager con diccionario con claves `min_lap_distance` (distancia mínima de vuelta) y `min_track_distance` (distancia mínima para completar la pista). Además, se creó el archivo JSON de configuración para Nivel 1 con los datos ya especificados.

26-27 de octubre de 2025 - Búsqueda de Assets de Música para Juego de Plataformas y Carreras

Se realizó una búsqueda exhaustiva de música para juegos de plataformas y carreras explorando repositorios de audio libre, además de creación assets de audio propio.

28 de octubre de 2025 - Actualización del Sistema de Logs

Se actualizó el sistema de logs implementando una organización jerárquica por juego, en lugar de archivo único `modex.log` guardando información de todos los juegos dificultando el análisis por lo que se estableció una estructura de directorio `/logs` conteniendo subdirectorios `/racing`, `/platform`, `/fight`, cada uno con su propio archivo de log rotativo, permitiendo análisis independiente de datos específicos de cada juego.

29 de octubre de 2025 - Desarrollo del Módulo `utils.colors`, Colores del Sistema

Se desarrolló el módulo `utils/colors.py` que contiene opciones de colores globales del sistema, el módulo contiene constantes nombradas claras en lugar de tuplas RGB o RGBA literales esparcidas por el código base, facilitando mantenimiento de identidad visual consistente.

3 de noviembre de 2025 - Inicio del Desarrollo de ModeX Versión 1.2

Se inició con el desarrollo de ModeX Versión 1.2 con contenido adicional, y preparación específica para presentación en el concurso local programado para el 25 de noviembre (22 días restantes). Se priorizaron features con mayor impacto visual, algunas implementaciones fueron:

- Mejoras de interfaz gráfica en general del arcade, adición de animaciones en personajes de juegos y robustez ante errores de usuario.

ModeX: Bitácora del Proyecto

- Búsqueda e integración de icono oficial de ModeX representando identidad visual del proyecto.

Entre el 3 y 5 de noviembre de 2025 - Implementación de Clases de Objetos del Juego de Plataformas

Se implementaron objetos adicionales del juego de plataformas ampliando la complejidad y la variedad:

- Clase Platform: representando a una plataforma de diferentes tipos, según el tipo de Plataforma que se pase al inicializador (static, moving, breakable, disappearing y bouncy).
- Clase Enemy: representando a un enemigo con comportamiento de patrullaje, según dirección de movimiento (horizontal o vertical) y límites de patrullaje.
- Clase Boss: representando a un jefe de nivel o enemigo de alta dificultad, que cuenta con máquina de estados (IA baja), ataques, animaciones según estado y movimiento independiente.
- Clase Bullet: representando proyectil disparado por jugador con dirección y velocidad inicial, auto destruyéndose tras distancia para prevenir acumulación ilimitada y uso de la técnica de optimización de rendimiento object pooling, utilizando un pool interno de un bullet por dirección para evitar la creación de nuevas texturas durante el videojuego.

5 de noviembre de 2025 - Implementación de Enums en el Juego de Plataformas

Se implementaron enums del juego de plataformas específicos para comportamiento del Boss:

- Clase AttackPattern: enum con patrones de ataque.
 - o BASIC_COMBO
 - o AGGRESSIVE
 - o DEFENSIVE
 - o SUMMONER
 - o BERSEKER
 - TACTICAL
- Clase BossPhase: enum marcando transiciones de fase.
 - o PHASE_1: con 100-75% HP, usa ataques básicos.

ModeX: Bitácora del Proyecto

- o PHASE_2: con 75-50% HP, usa ataques combinados.
- o PHASE_3: con 50-25% HP, usa ataques
- o PHASE_FINAL: con 25-0%, usa todos los ataques y estado ENRAGE.
- Clase BossState: enum gestionando máquina de estados del boss.
 - o INACTIVE: Boss inactivo.
 - o IDLE: Boss estático.
 - o PATROL: Boss patrullando.
 - o ALERT: Boss en alerta.
 - o CHASE: Boss persiguiendo al jugador.
 - o MELEE_ATTACK: Boss atacando cuerpo a cuerpo.
 - o RANGED_ATTACK: Boss atacando a distancia.
 - o CHARGE: Boss cargando hacia el jugador.
 - o JUMP_ATTACK: Boss atacando con salto.
 - o SPIN_ATTACK: Boss realizando ataque giratorio.
 - o SUMMON: Boss invocando enemigos.
 - o SHIELD: Boss activando escudo.
 - o HURT: Boss recibiendo daño.
 - o STUNNED: Boss aturdido.
 - o ENRAGED: Boss en Modo Furia.
 - o RETREATING: Boss retrocediendo.
 - o DEATH: Boss muriendo.

ModeX: Bitácora del Proyecto

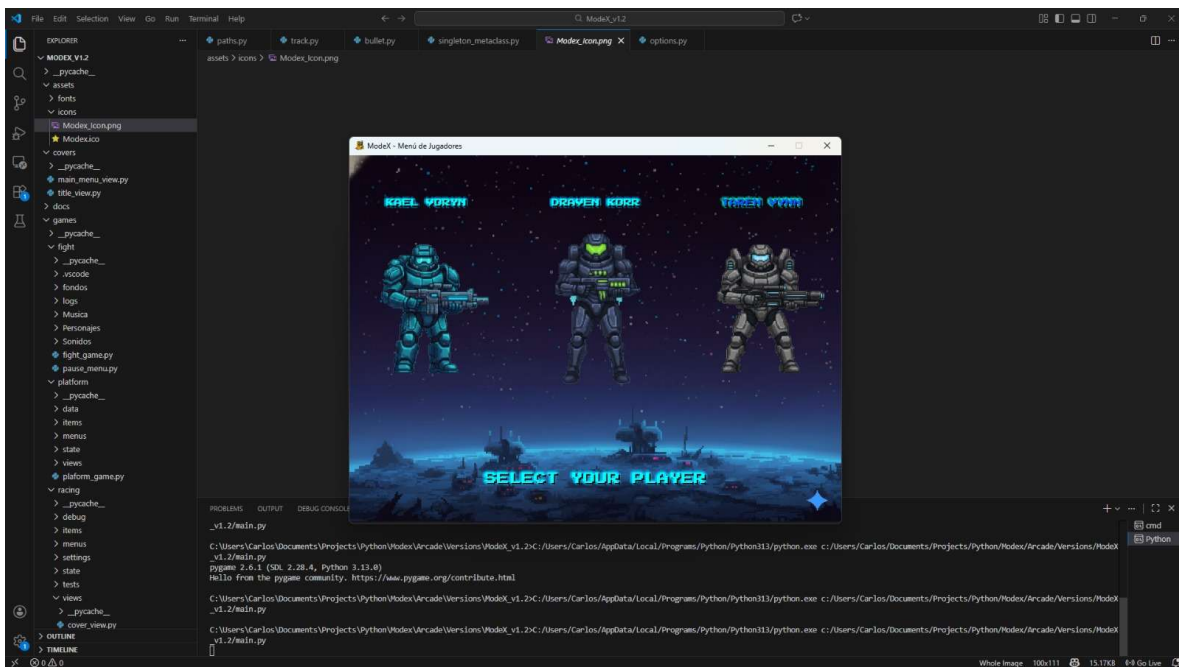
6 de noviembre de 2025 - Búsqueda de Assets Gráficos para el Juego de Plataformas

Se realizó una búsqueda complementaria de assets gráficos para juego de plataformas con enfoque en elementos visuales: sprites de escudo temporal, animaciones del Boss y elementos de UI (nombre del Boss).



7 de noviembre de 2025 - Actualización del Menú de Jugadores del Juego de Plataformas

Se actualizó el menú de jugadores del juego de plataformas agregando 2 personajes jugables al ya existente. Además, se agregó la selección de 3 jugadores completamente funcionales con animaciones y ataques diferentes al menú de jugadores.



ModeX: Bitácora del Proyecto

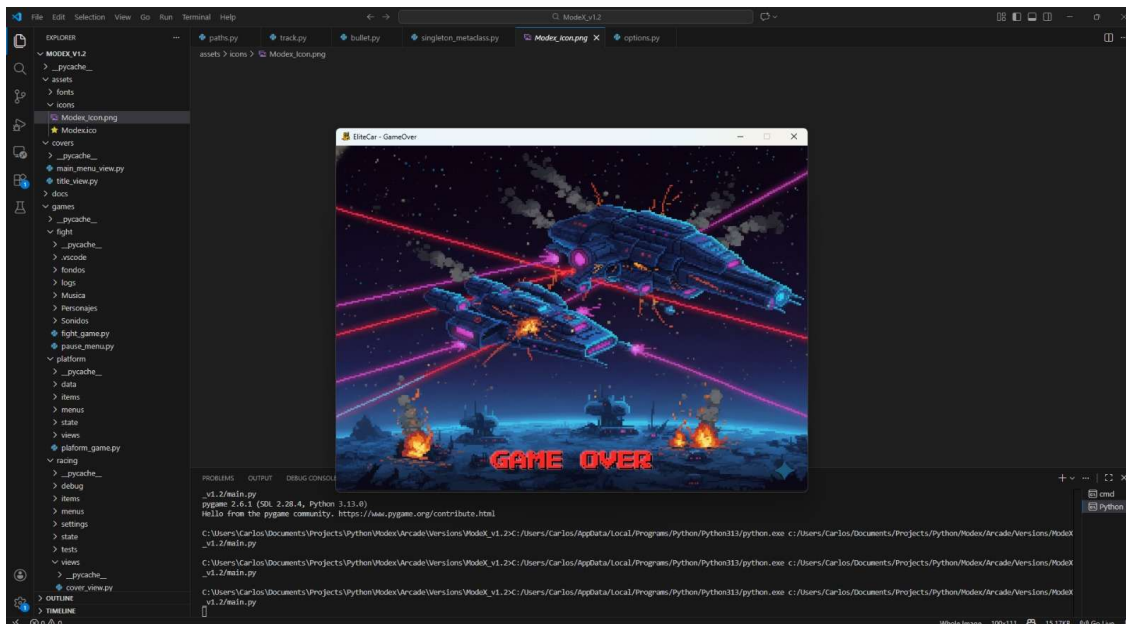
8 de noviembre de 2025 - Implementación de Nuevos Escenarios para el Juego de Peleas

Se implementaron nuevos escenarios para el menú de escenarios del juego de peleas integrando assets gráficos obtenidos de días previos. Al final de esta implementación se contó con 6 escenarios para seleccionar en el menú de escenarios del juego de peleas.

Entre el 8 y 10 de noviembre de 2025 - Desarrollo de Vistas Complementarias del Juego de Plataformas

Se desarrollaron las vistas complementarias del juego de plataformas siguiendo mismo patrón que juego de carreras:

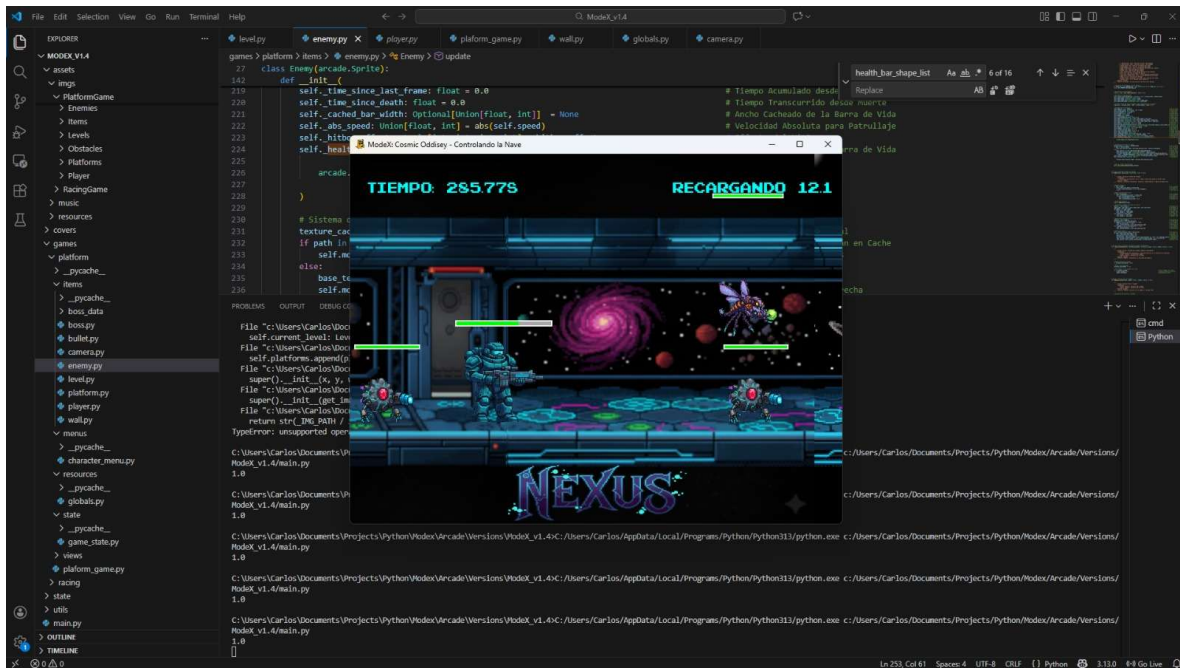
- WinView: esta vista se muestra al completar el juego en su totalidad, permite regresar al menú principal del juego o regresar al menú principal del arcade.
- GameOverView: esta vista muestra el GameOver al morir en el nivel, permite regresar al menú principal del juego o regresar al menú principal del arcade.
- PauseView: esta vista es accesible desde cualquier momento al pausar el nivel (cuando se presiona la tecla Enter) con opciones de resumir el juego, reiniciar el juego, ir al menú principal del juego de plataformas e ir al menú principal del arcade.



ModeX: Bitácora del Proyecto

Entre el 10 y 11 de noviembre de 2025 - Actualización del Juego de Plataformas para ModeX Versión 1.2

Se actualizó el juego de plataformas para la Versión 1.2 refinando mecánicas core, expandiendo contenido y puliendo presentación: se integraron nuevos métodos a la clase Player, como shoot(), melee_attack() y active_shield(), métodos de ataque y protección del Player. También se agregaron los textos de UI (tiempo transcurrido del nivel y estado del escudo del jugador) y se dibujaron fondos del nivel y textura del nombre del Boss.



12 de noviembre de 2025 - Búsqueda de Assets para el Juego de Shooter

Se realizó búsqueda de assets gráficos para el cuarto juego del sistema del arcade: un shooter cooperativo de zombies con mecánicas de defensa de oleadas. Se obtuvieron sprites de zombies con variaciones (zombie caminante básico lento, zombie corredor rápido con menos salud, zombie tanque lento pero resistente, zombie especial que explota al morir), assets de armas, elementos de escenarios de fondo y textos de UI. Al mismo tiempo se creó el archivo JSON de configuración del Boss del Nivel 1 del juego de plataformas especificando todas sus respectivas propiedades.

ModeX: Bitácora del Proyecto



Entre el 12 y 13 de noviembre de 2025 - Actualización del Juego de Peleas para la Versión 1.2

Se actualizó el juego de peleas para la Versión 1.2 integrando nuevos personajes y escenarios, implementando multijugador, con balance de personajes ajustando estadísticas personajes y pulido general del juego incluyendo transiciones suaves entre pantallas, dibujado de barras de vida de los personajes y mejora de arquitectura en el game loop.

Entre el 13 y 14 de noviembre de 2025 - Aplicación de Tests del Juego de Carreras

En esta fecha se aplicaron tests completos del juego de carreras validando todos los sistemas exhaustivamente:

- Testing funcional: verificación de cada feature (inicio de carrera, pausa/resume, paso de checkpoints, completado de vueltas, victoria, game over, transiciones entre vistas) ejecutándose correctamente sin comportamiento inesperado.
- Testing de regresión: ejecución de una suite de prueba cases re-verificando que nuevas implementaciones no rompieron funcionalidad.
- Testing de usabilidad: observando jugadores novatos interactuando con juego sin instrucción previa identificando puntos de confusión por medio de feedback.

ModeX: Bitácora del Proyecto

14 de noviembre de 2025 - Implementación de Objetos Fundamentales para el Juego de Shooter

Se implementaron todos los objetos fundamentales que conforman el juego de shooter cooperativo, las clases implementadas fueron:

- **Bullet:** representa proyectiles con trayectoria lineal, velocidad inicial, daño y tiempo de vida limitado tras el cual se autodestruyen automáticamente para liberar recursos.
- **Camera:** cuenta con viewport que se centra entre las 2 posiciones de los jugadores sin dejar a uno fuera de la pantalla, permitiendo que en modo cooperativo de dos jugadores la cámara se centre entre ambos, con sistema de smoothing para transiciones suaves y restricciones que previenen mostrar áreas fuera de los límites del mapa.
- **Particle:** representa elementos visuales efímeros de corta duración para decoración visual como humo generado por disparos o salpicaduras de sangre de zombies heridos.
- **Player:** representa al jugador que combate a los zombies en el juego, cuenta con armas, métodos de movimiento y métodos de ataque o protección.
- **PowerUp:** representa un objeto pickup temporal que otorga beneficio estratégico al jugador durante período limitado, con efectos como munición infinita por treinta segundos, multiplicador de daño doble, o incremento significativo de velocidad de movimiento.
- **PowerUpMachine:** simula a una máquina expendedora interactiva estilo arcade permitiendo a los jugadores comprar power-ups específicos gastando dinero virtual acumulado mediante la eliminación de zombies.
- **Zombie:** implementa inteligencia artificial simple pero efectiva que persigue persistentemente al jugador más cercano aplicando daño por contacto, según la dificultad los zombies se vuelven más resistentes.

Entre el 15 y 18 de noviembre de 2025 - Actualización del Juego de Carreras para ModeX Versión 1.2

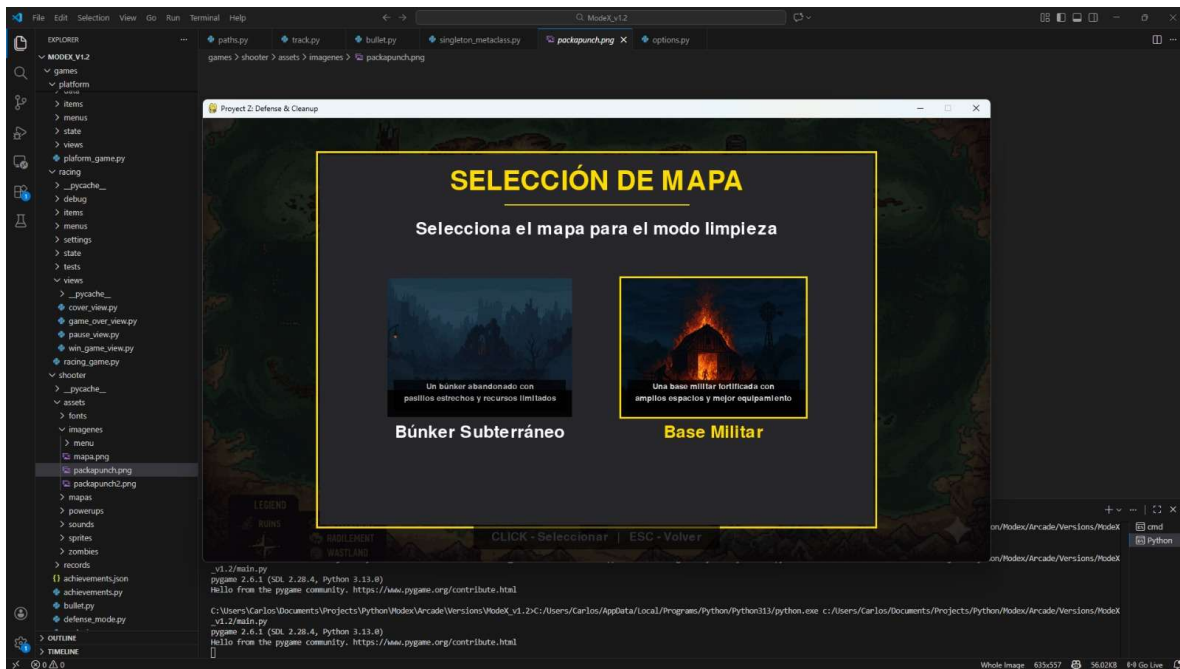
Se consolidó la actualización final del juego de carreras para la Versión 1.2 previo a la integración en el sistema arcade: se validaron nuevamente las mecánicas del juego asegurando la funcionalidad durante el gameplay y se mejoró el diseño de los textos de UI.

18 de noviembre de 2025 - Implementación de Vistas del Juego de Shooter

Se implementaron vistas del juego de shooter: menú principal con dos opciones de modo, limpieza (oleadas de zombies) y defensa (defensa de una puerta importante), portada con diferentes vistas del juego que cambian cada 5 segundos, vista de pausa con información de estado de partida (modo, ubicación, oleada actual, número de jugador, estado del jugador, HP, balas, puntaje, carga de balas),

ModeX: Bitácora del Proyecto

opciones de continuar/reiniciar/menú principal/salir. Finalmente, en esta fecha, se realizó una búsqueda adicional de assets gráficos para juego de plataformas enfocándose en bosses.



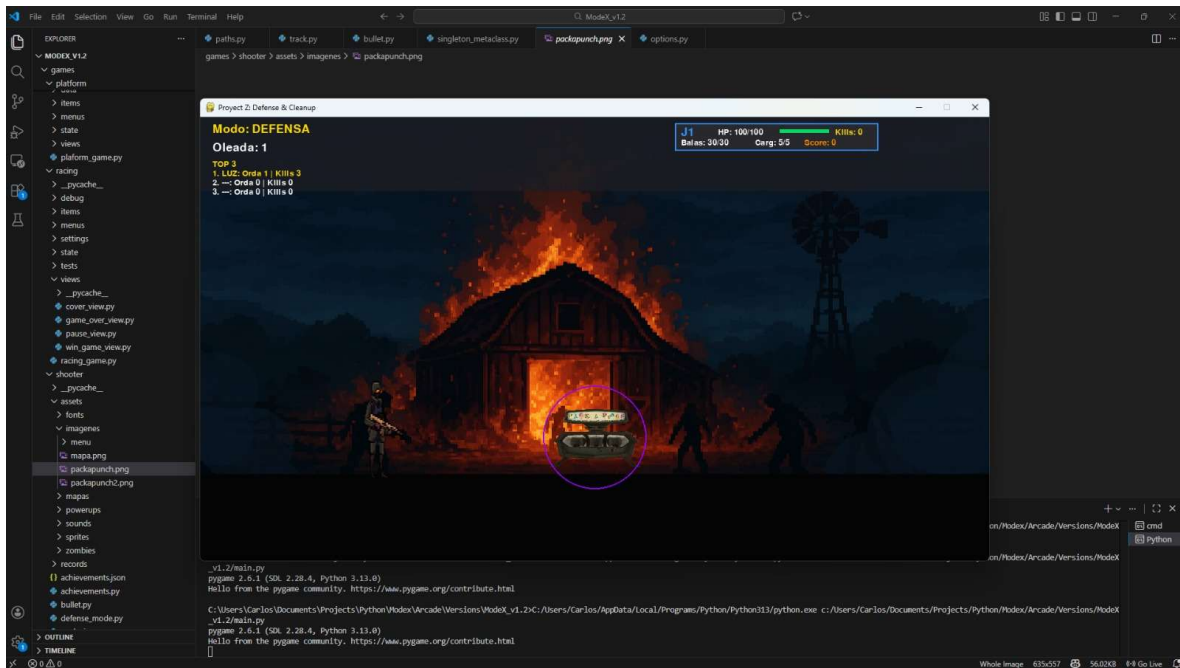
El 19 de noviembre de 2025 - Implementación del Bucle Principal del Juego de Shooter

Se implementó el bucle principal del juego de shooter coordinando los siguientes componentes y sistemas:

- Timer de oleada: cuenta tiempo total sobrevivido para registro de récords.
- Cantidad de zombies: se maneja la cantidad por oleada según la dificultad progresiva del juego.
- Tipos de zombies: introducidos progresivamente (oleadas 1-3 sólo se introducen básicos, 4-6 se introducen corredores aleatorios, 7-9 se introducen tanques, 10+ se introducen especiales).
- Spawns: son distribuidos aleatoriamente en zonas predefinidas fuera de visión de jugadores, previniendo apariciones injustas directamente visibles.
- Estado entre oleadas: se proporcionan ~30 segundos para comprar munición/power-ups, reacomodarse estratégicamente antes del siguiente asalto y notificación de inicio de oleada.

Simultáneamente diseñamos el cartel físico para presentación en concurso de prototipos mostrando screenshots de los cuatro juegos destacando variedad de géneros, lista de características principales y documentación propia del concurso.

ModeX: Bitácora del Proyecto



20 de noviembre de 2025 - Implementación del Sistema de Persistencia de Datos en el Juego de Shooter

Se implementó el sistema de persistencia de datos en juego de shooter utilizando archivos JSON almacenando récords de oleadas (tiempo de supervivencia según modo). El archivo JSON está estructurado con claves modo, nombre y tiempo facilitando lectura/escritura parcial sin cargar estructura completa y validación de esquema verificando presencia de claves requeridas y tipos correctos al cargar previniendo crashes por corrupción.

Entre el 20 y 21 de noviembre de 2025 - Integración del Juego de Peleas al Arcade

Se integró el juego de peleas al sistema arcade unificado: se agregó la opción de elegir jugar este juego en el menú principal, además de usar el módulo subprocess de Python para poder unificar las dos bibliotecas en un mismo arcade.

Entre el 21 y 22 de noviembre de 2025 - Integración del Juego de Shooter al Arcade

Se integró el juego de shooter al arcade aplicando el mismo proceso que con el juego de peleas: uso del módulo subprocess para ejecutar desde otro proceso y lograr la unificación de las dos bibliotecas de videojuegos 2D.

ModeX: Bitácora del Proyecto

23 de noviembre de 2025 - Preparativos Pre-Concurso

A dos días de presentación en concurso local, nos enfocamos en preparación de documentación requerida: elaboramos manual de usuario describiendo instalación (requisitos de sistema, pasos de instalación de Python y dependencias, prevención de problemas comunes), controles de cada juego (lista de inputs con acción correspondiente) y descripción de mecánicas principales (sin revelar soluciones completas permitiendo descubrimiento). También se elaboraron trípticos informativos para distribución en el evento resumiendo la información del proyecto. Por último, iniciamos redacción de documento de investigación formal cumpliendo con los requisitos y apartados del concurso.

24 de noviembre de 2025 - Preparación Final para la Presentación del Concurso

Día previo a la presentación, completamos la preparación final: se finalizó el documento de investigación para el concurso asegurando coherencia narrativa, lenguaje formal, formato consistente y cumplimiento de requisitos de extensión y estructura especificados en convocatoria. Además, se creó la primera versión estable utilizando un script .bat que ejecuta el punto de entrada principal del arcade, simplificando proceso de ejecución para jueces/evaluadores no técnicos eliminando necesidad de familiaridad con línea de comandos, además ejecutamos un playtesting final completando partidas de todos los juegos verificando ausencia de bugs críticos bloqueantes. Esta versión se designó ModeX V1.2 marcando la primera versión completamente estable presentable públicamente.

25 de noviembre de 2025 - Participación en el Concurso Local de Prototipos y Proyectos de Emprendimiento de la DGETI 2025

El equipo participó en el Concurso Local de Prototipos y Proyectos de Emprendimiento de la DGETI 2025, en la categoría de Prototipo de Software, presentando ModeX ante tres jueces conformado por, mayormente, profesores de programación para nuestra categoría. Realizamos una presentación oral de 15 minutos por juez, dando un total de 45 minutos de presentación, ya que, el jueceo no fue seguido, sino más bien por turnos, al presentar el proyecto describimos la motivación del proyecto (democratización de desarrollo de videojuegos), decisiones técnicas clave (elección de Python/Pygame/Arcade sobre Unity/Unreal, arquitectura modular), desafíos superados (optimización de rendimiento en hardware modesto, diseño de arquitectura extensible), y demostración en vivo de los cuatro juegos.

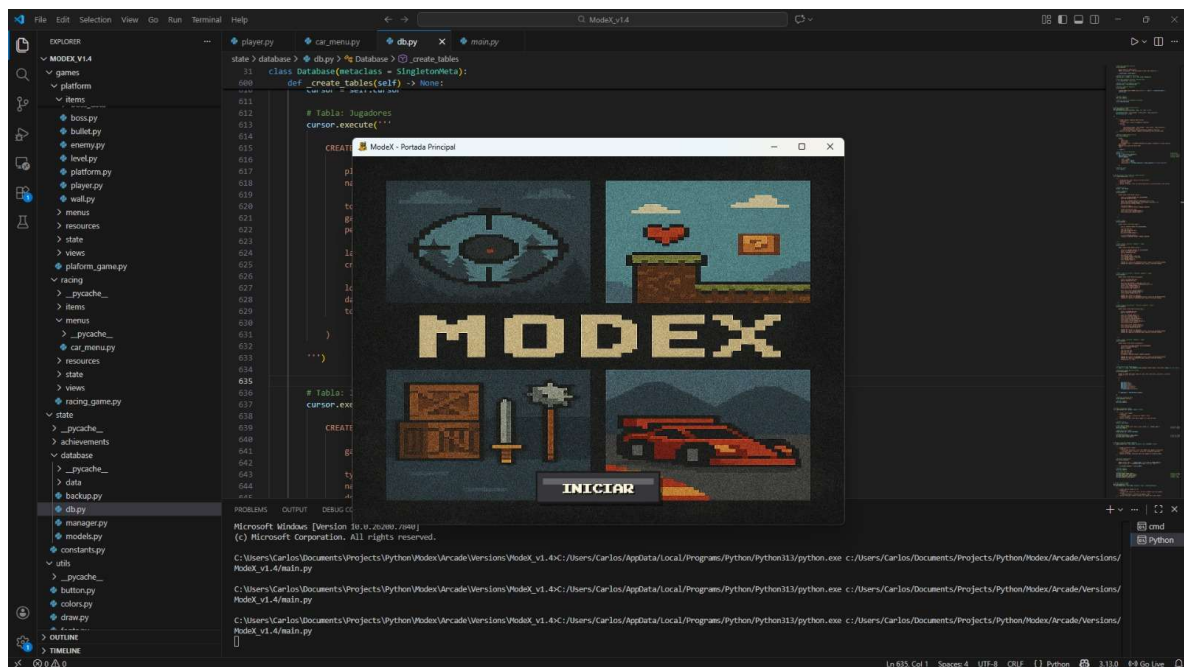
ModeX: Bitácora del Proyecto

El 26 de noviembre de 2025 - Premiación del Concurso Local de Prototipos y Proyectos de Emprendimiento de la DGETI 2025

Se anunció la premiación del concurso, otorgando el segundo lugar al equipo en la categoría de Prototipos de Software con retroalimentación positiva destacando arquitectura bien diseñada, variedad de géneros de juegos implementados demostrando versatilidad técnica y calidad del documento de investigación. Por haber obtenido el segundo lugar en el concurso local, se logró el avance a la Etapa Estatal del concurso programada para principios de marzo de 2026, dándonos tres meses adicionales para refinar y expandir el sistema. La retroalimentación de jueces identificó áreas de mejora: agregar el manejo de input por medio de botones para facilitar los controles del juego, ampliar contenido de los juegos (adición de más niveles a juegos) e inclusión de diagramas de visualización de proceso en el manual de usuario para explicar más detalles de los juegos.

Entre el 18 y 20 de diciembre de 2025 - Inicio de Desarrollo de ModeX Versión 1.3

Después del período de descanso post-concurso local, iniciamos desarrollo de ModeX Versión 1.3 enfocándonos en mejoras identificadas por jueces y features solicitados por playtesters, se actualizaron vistas de inicio (portada, menú principal) más elaboradas y transiciones más suaves. Se implementó una nueva vista `UserInputView` solicitando el nombre del jugador, pensada para almacenar el nombre del jugador en una base de datos evitando reconfiguración en cada sesión.



ModeX: Bitácora del Proyecto

Entre el 19 y 30 de diciembre de 2025 - Actualización del Juego de Carreras para ModeX Versión 1.3

Se realizó la actualización total del juego de carreras implementando sistema completamente rediseñado:

- Expansión de 1 pista a 3 pistas totalmente diferentes en diseño.
- Arquitectura de código totalmente rediseñada permitiendo configuraciones dinámicas de nivel.
- Escalado de elementos visuales según el tamaño de la ventana (resize).
- Aplicación de optimizaciones de rendimiento, permitiendo que el videojuego fuera más fluido.
- Eliminación de clases, archivos y estructuras de código inservibles en el nuevo sistema.



20 de diciembre de 2025 - Búsqueda de Assets Gráficos para el Juego de Shooter

Se realizó búsqueda de nuevos assets gráficos de zombies para el juego de shooter expandiendo la variedad visual:

- Zombies con diferentes niveles de descomposición (recién infectado con apariencia casi humana, parcialmente descompuesto con exposición ósea, severamente descompuesto casi esquelético).
- Zombies con vestimenta temática (zombie policía)
- Zombies especiales con características únicas visuales (zombie bloater con abdomen hinchado explotando al morir).

ModeX: Bitácora del Proyecto

21 de diciembre de 2025 - Actualización del Módulo de Utilidades

Se actualizó el módulo de utilidades con las siguientes acciones:

- Se implementó la clase Button que proporciona un componente UI reutilizable encapsulando lógica de botón interactivo (detección de hover mediante verificación de posición del mouse dentro de bounding box, detección de click mediante evento mouse button down mientras hovering, estados visuales diferenciados para normal/hover/pressed con colores distintos, callbacks ejecutados al hacer click y soporte para texto centrado automáticamente) reduciendo duplicación de código de botones esparcidos por múltiples vistas
 - Se implementó el módulo de fuentes (utils/fonts) consolidando carga de fuentes tipográficas mediante función helper load_all_fonts() para así solo cargar una vez las fuentes durante el videojuego.
-

Entre el 21 y 23 de diciembre de 2025 - Reorganización y Unificación del Módulo de Assets

Se unificó el módulo de assets reorganizando estructura de carpetas eliminando duplicación y archivos obsoletos:

- Se establecieron los siguientes subdirectorios (conteniendo subdirectorios para cada juego):
 - /assets/imgs/
 - /assets/music/
 - /assets/resources/
 - /assets/fonts/
 - /assets/icons
- Eliminación de imágenes, archivos y directorios no utilizados globalmente en el sistema de arcade

Esto permitió reducir el tamaño total del videojuego, lo que permite que este sea más ligero, ocupa menos espacio en disco, se mejora el rendimiento y su distribución es más fácil.

Entre el 22 y 30 de diciembre de 2025 - Implementación del Módulo de Estado del Arcade

Se implementó el módulo de estado del juego utilizando base de datos SQLite para persistencia estructurada con el siguiente desglose de tablas:

- TABLE player:
 - player_id: ID del jugador.
 - name: nombre del jugador.

ModeX: Bitácora del Proyecto

- total_score: puntuación total del jugador.
- games_completed: juegos que ha completado el jugador.
- perfect_sessions: sesiones perfectas acumuladas por el jugador.
- last_played: última fecha de juego del jugador.
- created_at: timestamp de creación del registro.
- longest_session_date: fecha de la sesión con más tiempo de duración del jugador.
- daily_login_streak: racha de días seguidos jugando del jugador,
- total_time_played: tiempo total jugado en el arcade del jugador.
- TABLE games:
 - game_id: ID de juego.
 - type: tipo de juego.
 - name: nombre del juego.
 - description: descripción del juego.
 - max_level: nivel máximo para desbloquear.
 - is_active: estado actual del juego (activo/inactivo).
 - created_at: timestamp de creación del registro.
- TABLE session
 - session_id: ID de sesión.
 - player_id: ID de jugador.
 - game_id: ID del juego.
 - date: fecha de sesión.
 - time: tiempo de duración de sesión.
 - score: puntaje de sesión.
 - wins: victorias de sesión.
 - losses : partidas perdidas de sesión.
 - FOREIGN KEY (player_id) REFERENCES players (player_id) ON DELETE CASCADE
 - FOREIGN KEY (game_id) REFERENCES games (game_id) ON DELETE CASCADE
- TABLE racing_game:
 - player_id: ID del jugador.
 - game_id : ID del juego de carreras.
 - best_time: mejor tiempo del juego de carreras.
 - total_wins: total de carreras ganadas en el juego de carreras.
 - perfect_laps: vueltas perfectas en el juego de carreras.
 - total_crashes: total de choques en el juego de carreras.
 - total_distance: distancia total recorrida en el juego de carreras.
 - daily_login_streak: racha diaria en el juego de carreras.
 - total_time_played : tiempo total jugado en el juego de carreras.
 - level_reached: máximo nivel alcanzado en el juego de carreras.
 - PRIMARY KEY (player_id, game_id)
 - FOREIGN KEY (player_id) REFERENCES players (player_id) ON DELETE CASCADE

ModeX: Bitácora del Proyecto

- FOREIGN KEY (game_id) REFERENCES games (game_id) ON DELETE CASCADE
- TABLE platform_game
 - player_id: ID del jugador en el juego de plataformas.
 - game_id: ID del juego.
 - high_score: puntuación más alta en el juego de plataformas.
 - total_score: puntaje total en el juego de plataformas.
 - best_time_level: mejor tiempo de nivel en el juego de plataformas.
 - levels_completed: número de niveles completados en el juego de plataformas.
 - total_deaths: muertes totales en el juego de plataformas.
 - total_enemies_killed: total de enemigos derrotados en el juego de plataformas.
 - total_distance: distancia total recorrida en el juego de plataformas.
 - daily_login_streak: racha diaria en el juego de plataformas.
 - total_time_played: tiempo total jugado en el juego de plataformas.
 - level_reached: máximo nivel alcanzado en el juego de plataformas.
 - PRIMARY KEY (player_id, game_id),
 - FOREIGN KEY (player_id) REFERENCES players (player_id) ON DELETE CASCADE
 - FOREIGN KEY (game_id) REFERENCES games (game_id) ON DELETE CASCADE
- TABLE achievements
 - achievement_id: ID del logro.
 - player_id: ID del jugador.
 - game_id: ID del juego.
 - code: código del logro.
 - name: nombre del logro.
 - description: descripción del logro
 - unlocked_at: timestamp de creación del registro.
 - FOREIGN KEY (player_id) REFERENCES players (player_id) ON DELETE CASCADE
 - FOREIGN KEY (game_id) REFERENCES games (game_id) ON DELETE CASCADE
 - UNIQUE(player_id, code)

1 de enero de 2026

Se inició con el desarrollo de ModeX Versión 1.4 enfocándose en optimizaciones de rendimiento globales, rediseño de arquitectura del juego de plataformas y expansión del contenido del juego de plataforma.

ModeX: Bitácora del Proyecto

Del 1 - 31 enero de 2026 - Implementación de Optimizaciones de Rendimiento Globales al Arcade

Se implementaron optimizaciones de rendimiento globales aplicando técnicas conocidas:

- Object pooling: no solo para balas sino también para partículas, enemigos, y efectos visuales reduciendo llamadas a garbage collector.
- Sprite batching: agrupando el renderizado de sprites por textura compartida reduciendo draw calls e incrementando framerate en escenas complejas.
- Culling espacial: omitiendo actualización/dibujado de entidades fuera de viewport.
- Precarga de assets: cargando en memoria texturas/sonidos compartidos entre juegos eliminando stutters durante transiciones.
- Uso de variables en vez de atributos: uso de variables locales en bucles de dibujado y actualización para eliminar acceso a atributos recurrentes.

Con estas optimizaciones el juego mejoró bastante en rendimiento, tomando en cuenta que ya contaba con el feature de ventana resizable, por lo que estas implementaciones fueron cruciales para que el videojuego fuera fluido.

14 de enero de 2026 - Implementación de Botones Interactivos en el Arcade

Se implementaron botones interactivos en casi todas las vistas del juego, facilitando navegación mediante mouse complementando controles de teclado. Las vistas previamente navegables sólo actuaban mediante teclas (arriba/abajo o 1/2/3/4 para seleccionar, enter para confirmar), ahora incluyen botones clickeables con feedback visual de hover, permitiendo usuarios preferentes de mouse interactuar sin necesidad de aprenderse controles de teclas, especialmente valioso para jugadores casuales menos familiarizados con convenciones de videojuegos. Además, integramos el soporte del mouse al arcade globalmente habilitando cursor visible dentro de la ventana del juego, procesamiento de eventos de mouse (movimiento, clicks, wheel scroll) y coordinación con sistema de input de teclado permitiendo uso intercambiable o simultáneo sin conflictos.

3 de febrero de 2026

Se inició con el desarrollo de ModeX Versión 1.5 enfocándonos en la finalización de contenido restante y preparación para el concurso estatal. Entre una de las implementaciones más importantes está el aumento de dificultad en juego de shooter de zombies mediante sistema de escalamiento progresivo.

ModeX: Bitácora del Proyecto

Entre el 6-9 de febrero de 2026

Se implementaron mejoras en la arquitectura de código del juego de peleas:

- Organización modular del juego.
 - Eliminación de features que no se ocupaban en el videojuego.
 - Adición de sprites y texturas (assets gráficos)
-

Entre el 8-10 de febrero de 2026

Se integraron los dos juegos actualizados (peleas, shooter zombies) al arcade, verificando compatibilidad con sistemas compartidos, estandarizado de interfaces y validando transiciones correctas entre vistas.

Entre el 10-14 de febrero de 2026

Se ejecutaron pruebas del videojuego con usuarios mediante playtesting, recopilando información para la documentación técnica del concurso. Se reclutaron 25 participantes (estudiantes de programación, jugadores casuales, personas sin experiencia gaming) ejecutando sesiones estructuradas de 3 minutos por juego en hardware de prueba mientras el sistema registra datos de rendimiento automáticamente (framerate, memoria, inputs por segundo).

Entre el 15-18 de febrero de 2026

Se elaboró el documento de investigación para el concurso estatal expandiendo el documento previo de la etapa local:

- Se amplió la hipótesis para que fuera más convincente.
- Se realizaron pruebas de viabilidad del proyecto.
- Se obtuvieron datos de rendimiento mediante playtesting para la documentación técnica.
- Mediante los datos recabados se pudo realizar un análisis de resultados.

ModeX: Bitácora del Proyecto

Entre el 18 - 21 de febrero de 2026

Se elaboró la bitácora para el concurso estatal documentando cronológicamente todo proceso de desarrollo, en él se describen:

- Actividades completadas con fecha de realización.
 - Desafíos encontrados.
 - Soluciones implementadas.
 - Decisiones de diseño con justificaciones.
 - Aprendizajes obtenidos.
 - Además de la inclusión de screenshots y fotos ilustrando progreso visual.
-

21 de febrero de 2026

Elaboramos el diseño para cartel del concurso estatal actualizando el diseño de la etapa local:

- Layout más profesional.
 - Información acerca del proyecto mejor organizada jerárquicamente
 - Inclusión de gráfico de rendimiento.
 - Inclusión del logo actual de ModeX.
-

22 de febrero de 2026

Elaboración del manual de instalación detallando:

- Requisitos de sistema especificando mínimos y recomendados (CPU, RAM, GPU, SO, espacio en disco, resolución de pantalla).
- Instrucciones de ejecución paso a paso con screenshots para guía visual.
- Troubleshooting exhaustivo de problemas comunes con sus respectivas soluciones.

También se elaboró la documentación técnica comprehensiva con los siguientes aspectos:

- Arquitectura del sistema mostrando herencia y relaciones.
- Diagramas de secuencia ilustrando el flujo de interacciones entre componentes.
- Análisis de rendimiento con gráficos.
- Comparativas de métricas en diferentes configuraciones de hardware.
- Técnicas de mantenibilidad del código.
- Sección de decisiones de diseño justificando elecciones arquitectónicas.

ModeX: Bitácora del Proyecto

23 de febrero de 2026

Para finalizar con la Versión 1.5 se elaboró el manual de usuario para concurso estatal refinando el manual de etapa loca y agregando estas secciones:

- Sección de controles: diagramas visuales mejorados mostrando layout de teclado con teclas resaltadas.
- Sección de estrategias: proporciona tips del videojuego sin spoilers completos.
- Sección de FAQ: respuestas a preguntas comunes basándose en feedback de playtesting.

Además, en esta fecha se creó el ejecutable .exe de ModeX Versión 1.5 empaquetando el proyecto de Python en un ejecutable standalone mediante PyInstaller: configuración especificando punto de entrada, inclusión de assets y dependencias y generación del ejecutable de archivo único simplificando su distribución.

Este cronograma detallado documenta la evolución completa del proyecto ModeX desde su inicio el 11 de septiembre de 2025 hasta su forma final para el concurso estatal en el 23 febrero de 2026, reflejando el proceso iterativo de desarrollo, refinamiento basado en feedback y preparación rigurosa para competición académica, totalizando aproximadamente 6 meses de desarrollo activo con participación de equipo multidisciplinario coordinado mediante la metodología RAD (Rapid Application Development) adaptada a un entorno educativo.