Descripción breve

Documento inicial para guiar el desarrollo del proyecto.

Documento de diseño inicial

KeyboardRunner

Logotipo

Descripción generada automáticamente

Mateo Galvis Guayana

mateogalvisgamedev@gmail.com

Contenido

[Resumen del juego 2](#_Toc180697427)

[Gráficos y visuales 2](#_Toc180697428)

[Mecánicas Principales 3](#_Toc180697429)

[Core GameLoop 4](#_Toc180697430)

[Enemigos 5](#_Toc180697431)

[UI (Propuesta) 7](#_Toc180697432)

[Equipo 9](#_Toc180697433)

[Referencias 9](#_Toc180697434)

# Resumen del juego

**Nombre del Juego:** Keyboard Runner  
**Género:** Arcade / Educativo  
**Público Objetivo:** Todas las edades, especialmente personas que deseen mejorar sus habilidades de mecanografía.  
**Plataforma:** PC (Windows, macOS, Linux), con posibilidad de adaptación a web y dispositivos móviles.  
**Motor del Juego:** Unity (Versión 2023 o superior).

**Concepto Principal:**  
*Keyboard Runner* es un juego educativo tipo arcade donde los jugadores desarrollan habilidades motrices escribiendo rápidamente las letras que aparecen en pantalla para superar obstáculos y avanzar en niveles temáticos inspirados en aventuras como Indiana Jones.

# Gráficos y visuales

**Estilo Artístico:**

* Gráficos 2D con un enfoque en animaciones fluidas y colores vibrantes.
* Inspiración en el estilo aventurero: entornos de jungla, templos antiguos y trampas visualmente atractivas.

**Personaje Principal:**

* Un aventurero estilizado similar a Indiana Jones.
* Animaciones: correr, saltar, perder (caer al hoyo o chocar con un obstáculo).

**Obstáculos y Elementos Interactivos:**

* Piedras rodantes: decorativas, simulan persecución.
* Hoyos con bordes diferenciados: verde claro para zonas de pérdida, verde oscuro como zona segura.
* Letras flotantes: representan las teclas que el jugador debe presionar.

# 

# Mecánicas Principales

**Objetivo del Jugador:**  
Presionar rápidamente las teclas correctas que aparecen en pantalla para:

* Saltar sobre obstáculos (hoyos o piedras).
* Acumular puntos al realizar acciones exitosas.
* Evitar colisiones que resulten en la pérdida de la partida.

**Sistema de Entrada:**

* El teclado será el controlador principal.
* Las teclas generadas aleatoriamente deberán presionarse dentro de un tiempo límite.

**Reglas del Juego:**

* El jugador gana puntos por cada salto exitoso.
* Si el personaje toca un obstáculo sin presionar la tecla correcta, la partida termina.
* A medida que el jugador avanza, las letras aparecen con mayor rapidez.

**Progresión:**

* El juego tiene niveles secuenciales. Cada nivel incrementa la dificultad (mayor velocidad de letras y obstáculos más complejos).
* Se planea implementar un sistema de niveles infinitos basado en puntuaciones altas.

# Core GameLoop

**Inicio:**

1. **Preparación del nivel:**
   * El nivel comienza mostrando un breve conteo regresivo (por ejemplo, "3, 2, 1... ¡Corre!").
   * El personaje inicia su carrera automática hacia adelante mientras el escenario se mueve lateralmente.

**Ciclo de Juego Activo:**

1. **Generación de letras:**
   * Letras aleatorias aparecen en pantalla.
   * Cada letra representa una tecla que el jugador debe presionar para realizar una acción, como saltar un obstáculo.
   * Las letras desaparecen si no son presionadas en el tiempo límite.
2. **Interacción del jugador:**
   * Si el jugador presiona la letra correcta:
     + El personaje salta el obstáculo.
     + Se suma un punto al marcador.
     + Se reproduce un efecto visual y sonoro satisfactorio.
   * Si el jugador no presiona la letra correcta o lo hace fuera de tiempo:
     + El personaje tropieza o choca con el obstáculo.
     + Se reproduce un efecto visual y sonoro de error.
3. **Progresión del nivel:**
   * La velocidad de generación de letras aumenta gradualmente.
   * Los obstáculos son más frecuentes y requieren reflejos más rápidos.

**Fin del Ciclo de Juego:**

1. **Condición de derrota:**
   * El jugador pierde si:
     + Toca un obstáculo (por ejemplo, un hoyo).
     + No presiona las letras requeridas dentro del tiempo límite repetidamente.
   * El juego muestra la puntuación final y un mensaje de motivación para intentar de nuevo.
2. **Condición de victoria:**
   * Si el jugador alcanza un puntaje objetivo (por ejemplo, 100 puntos) o completa un nivel predefinido:
     + Se desbloquea el siguiente nivel o se activa un modo de mayor dificultad.

**Reinicio o Continuación:**

* Tras la derrota o completar un nivel, el jugador puede:
  + Reiniciar el nivel actual.
  + Volver al menú principal para seleccionar un nivel o modo.

# Diseño de Sonido

**Música:**

* Banda sonora dinámica que acelera su tempo conforme avanza el nivel.
* Opción para que el jugador seleccione música personalizada (pendiente de implementación).

**Efectos de Sonido:**

* Sonido al presionar una tecla correcta.
* Sonido de alerta para errores o cuando el jugador pierde.
* Efectos ambientales como viento, pasos y sonidos de obstáculos

# Características Tecnicas

**Motor de Juego:** Unity (C# como lenguaje principal).  
**Componentes Clave:**

* Sistema de generación aleatoria de letras (letras seleccionadas de un array).
* Mecánicas de salto y detección de colisiones (utilizando *Rigidbody* y *Colliders*).
* Sistema de puntuación acumulativa.

**Requisitos Técnicos:**

* Resolución adaptable (mínima: 1280x720).
* Compatibilidad con teclado físico y teclados virtuales.
* Soporte para guardar puntuaciones altas localmente.

# Equipo

Mateo Galvis Guayana:

* **Rol:** Desarrollador Full Stack y Único Responsable del Proyecto.
* **Responsabilidades:** Mateo es el único encargado de todos los aspectos del desarrollo del videojuego. Esto incluye:
  + **Dirección del Proyecto:** Supervisión y toma de decisiones en todas las fases del desarrollo.
  + **Programación:** Implementación de mecánicas principales, sistemas del juego y optimización del rendimiento.
  + **Diseño de Niveles:** Creación de la estructura, diseño de los niveles y ajustes de dificultad para ofrecer una experiencia desafiante y divertida.
  + **Arte y Animación:** Diseño visual del juego, incluyendo personajes, escenarios y efectos visuales.
  + **Diseño de Sonido:** Selección, creación e integración de la música y efectos sonoros.
  + **Pruebas:** Identificación y corrección de errores, así como ajustes basados en la experiencia del usuario.
  + **Documentación:** Elaboración de la planificación, diseño de juego y cualquier documento necesario para el desarrollo y publicación.

Mateo asume todos los roles de desarrollo para garantizar una visión unificada y el cumplimiento de los objetivos del proyecto en su totalidad.