TECNOLÓGICO
NACIONAL DE MÉXICO®

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

Departamento: Sistemas

Carrera: Ingeniería en Sistemas Computacionales

Materia: Graficación

Tarea: Juego 3D: "Squash the Creeps!"

Unidad: 3

Integrantes:

- Chiu Flores Fernando. 22070307
- Hernández Martínez José Antonio. 22070302

Maestro: Jorge Peralta Escobar

Lugar: Cd. Madero, Tamaulipas.

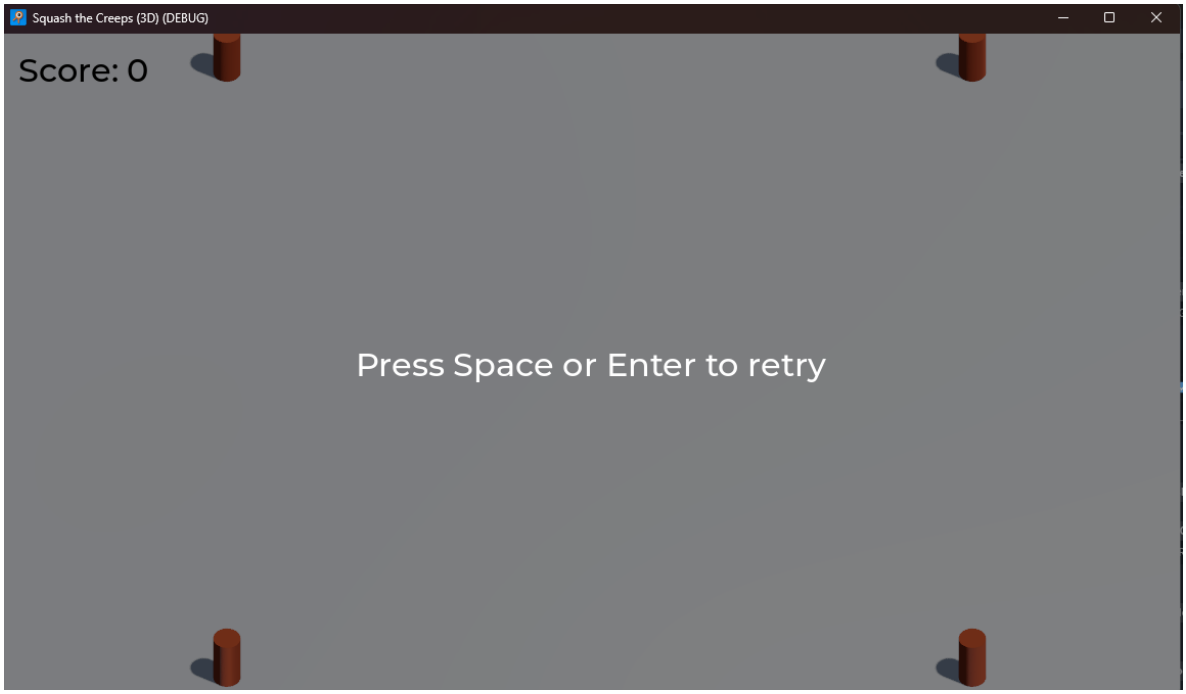
Fecha: 16 de Diciembre del 2024.

ÍNDICE

Resumen

Squash the Creeps es un juego en 3D de acción y supervivencia, desarrollado en el motor Godot Engine y con animaciones de personajes y entorno creadas en Blender. El jugador controla un personaje tridimensional en un mapa en 3D, utilizando las teclas de flechas (arriba, abajo, izquierda y derecha) para moverse por el espacio. El objetivo del juego es eliminar a los enemigos saltando sobre ellos, lo cual se consigue presionando la barra espaciadora para saltar. Cada vez que el jugador aplasta a un enemigo, su puntuación aumenta en uno. Los enemigos aparecen aleatoriamente en el mapa y el jugador debe moverse rápidamente para aplastarlos sin ser tocado por ellos. El juego tiene una atmósfera sencilla, con música de fondo constante y un objetivo claro: obtener la mayor puntuación posible antes de perder.

La partida comienza con una pantalla de inicio que muestra un mensaje pidiendo al jugador presionar la barra espaciadora para comenzar. Una vez iniciada la partida, los enemigos comienzan a aparecer aleatoriamente en el mapa, y el jugador debe usar su habilidad para desplazarse por el mapa y saltar sobre los enemigos para eliminarlos. El juego termina cuando un enemigo toca al jugador, lo que activa un reinicio automático de la partida, llevando al jugador de nuevo a la pantalla de inicio.



Objetivo del Juego

El objetivo central de **Squash the Creeps** es simple pero desafiante: el jugador debe sobrevivir el mayor tiempo posible, eliminando a los enemigos que aparecen aleatoriamente en el mapa. Cada vez que el jugador salta sobre un enemigo y lo aplasta, su puntuación aumenta en uno. La puntuación se muestra de manera continua en la parte superior de la pantalla, lo que permite al jugador monitorear su progreso mientras juega.

El desafío aumenta conforme el tiempo avanza, ya que los enemigos siguen apareciendo, y el jugador debe evitar que lo toquen. Los enemigos no tienen un patrón de movimiento predecible, lo que incrementa la dificultad, y requiere que el jugador reaccione rápidamente y con precisión.

La partida termina cuando un enemigo toca al jugador. Al perder, el juego automáticamente regresa a la pantalla de inicio, donde el jugador puede presionar la barra espaciadora nuevamente para reiniciar el juego.

En resumen, el juego pone a prueba la agilidad, precisión y capacidad de reacción del jugador, que debe sobrevivir el mayor tiempo posible aplastando enemigos y evitando ser tocado por ellos.

Género y Estilo

Género: Acción, Supervivencia, Plataformas, Arcade.

Estilo gráfico: El juego utiliza gráficos en 3D, creados con Blender, que ofrecen un entorno visual sencillo pero efectivo. Los modelos 3D del personaje y los enemigos

están diseñados para ser fácilmente identificables y para que el jugador pueda interactuar con ellos sin dificultad.

Estilo de Juego: Plataformas en 3D, con un enfoque en la agilidad del jugador para moverse y saltar sobre los enemigos. El estilo de juego se asemeja a los juegos arcade tradicionales, en los que el objetivo es superar retos mediante la habilidad de respuesta rápida y la acción constante.

Dificultad: A medida que avanza el tiempo, los enemigos aparecen más rápidamente y en mayor cantidad, aumentando así el nivel de dificultad. Aunque al principio el jugador puede sentir que el juego es fácil, conforme progresa, se vuelve más desafiante debido a la necesidad de coordinar el movimiento y el salto con rapidez.

Mecánicas del Juego

Main principal: Este código maneja la lógica para generar enemigos (mobs) y reiniciar el juego en un juego de Godot. La variable mob_scene contiene la escena del mob que se instanciará. Al iniciar (_ready), oculta el botón de reinicio de la interfaz de usuario. Cuando se presiona la tecla de aceptación (ui_accept) y el botón de reinicio es visible, recarga la escena actual.

La función _on_mob_timer_timeout se ejecuta cuando un temporizador de mobs llega a su fin, creando un nuevo mob a partir de la escena mob_scene, asignándole una ubicación aleatoria a lo largo de un camino de spawn (SpawnPath/SpawnLocation). El mob es inicializado con la posición de su spawn y la del jugador, y luego se añade a la escena. Además, se conecta la señal squashed del mob a la función que actualiza la puntuación del jugador.

Cuando el jugador es golpeado (función _on_player_hit), el temporizador de mobs se detiene y se muestra el botón de reinicio.

```
1   extends Node
2   @export var mob_scene: PackedScene
3   func _ready():
4       $UserInterface/Retry.hide()
5   func _unhandled_input(event):
6       if event.is_action_pressed("ui_accept") and $UserInterface/Retry.visible:
7           get_tree().reload_current_scene()
8   func _on_mob_timer_timeout():
9       var mob = mob_scene.instantiate()
10      var mob_spawn_location = get_node("SpawnPath/SpawnLocation")
11      mob_spawn_location.progress_ratio = randf()
12      var player_position = $Player.position
13      mob.initialize(mob_spawn_location.position, player_position)
14      add_child(mob)
15      mob.squashed.connect($UserInterface/ScoreLabel._on_Mob_squashed)
16  func _on_player_hit():
17      $MobTimer.stop()
18      $UserInterface/Retry.show()
```

Movimiento del Jugador: El jugador controla al personaje tridimensional utilizando las teclas de flecha del teclado (arriba, abajo, izquierda, derecha). El personaje se mueve por un espacio tridimensional, lo que permite a los jugadores moverse en varias direcciones, esquivar enemigos y posicionarse estratégicamente para saltar sobre ellos. El control es ágil y directo, lo que permite a los jugadores moverse rápidamente y tomar decisiones tácticas sobre dónde ir.

Este código es para un jugador en 3D en Godot que se mueve, salta y rebota al interactuar con objetos. El jugador se mueve en las direcciones definidas por las teclas de entrada (move_right, move_left, move_back, move_forward), con velocidad ajustable por la variable speed. Si el jugador está en el suelo y se presiona la tecla de salto (jump), aplica un impulso vertical (jump_impulse). También tiene un control de caída, acelerando hacia abajo con fall_acceleration. Cuando colisiona con objetos de la clase "mob", si la colisión es por arriba, rebota con un impulso (bounce_impulse). El jugador rota ligeramente dependiendo de su velocidad vertical. Si el jugador entra en contacto con un objeto etiquetado como "mob" mediante el detector, el jugador emite la señal hit, se elimina del juego (queue_free), y muere.

```gdscript
extends CharacterBody3D
signal hit
@export var speed = 14
@export var jump_impulse = 20
@export var bounce_impulse = 16
@export var fall_acceleration = 75

func _physics_process(delta):
    var direction = Vector3.ZERO
    if Input.is_action_pressed("move_right"):
        direction.x += 1
    if Input.is_action_pressed("move_left"):
        direction.x -= 1
    if Input.is_action_pressed("move_back"):
        direction.z += 1
    if Input.is_action_pressed("move_forward"):
        direction.z -= 1

    if direction != Vector3.ZERO:
        direction = direction.normalized()
        basis = Basis.looking_at(direction)
        $AnimationPlayer.speed_scale = 4
    else:
        $AnimationPlayer.speed_scale = 1
    velocity.x = direction.x * speed
    velocity.z = direction.z * speed
    if is_on_floor() and Input.is_action_just_pressed("jump"):
        velocity.y += jump_impulse
    velocity.y -= fall_acceleration * delta
    move_and_slide()
    for index in range(get_slide_collision_count()):
        var collision = get_slide_collision(index)
        if collision.get_collider().is_in_group("mob"):
            var mob = collision.get_collider()
            if Vector3.UP.dot(collision.get_normal()) > 0.1:
                mob.squash()
                velocity.y = bounce_impulse
                break
    rotation.x = PI / 6 * velocity.y / jump_impulse
func die():
    hit.emit()
    queue_free()
func _on_MobDetector_body_entered(_body):
    die()
```

Salto: La barra espaciadora se utiliza para hacer que el personaje salte. Al saltar, el jugador puede aplastar a los enemigos que se encuentren debajo de él. Esta mecánica de salto agrega un elemento de habilidad, ya que el jugador debe calcular el momento adecuado para saltar y aterrizar sobre los enemigos sin ser tocado por

ellos. El salto está diseñado para ser fácil de ejecutar, pero el timing y la ubicación son cruciales para eliminar a los enemigos y evitar el contacto con ellos.

Aparición de Enemigos: Los enemigos aparecen aleatoriamente en diferentes lugares del mapa en intervalos regulares. No hay un patrón fijo en el que los enemigos salgan, lo que mantiene al jugador constantemente alerta. Los enemigos se desplazan por el mapa de manera impredecible, y el jugador debe estar atento a su posición y movimiento para poder saltar sobre ellos o evitar que lo toquen. La aparición aleatoria de enemigos incrementa la dificultad, ya que el jugador no puede anticipar en qué parte del mapa aparecerán.

Puntuación: La puntuación es un reflejo de la habilidad del jugador para eliminar enemigos al saltar sobre ellos. Cada vez que el jugador aplasta un enemigo, su puntuación aumenta en uno. La puntuación se muestra en la parte superior de la pantalla y continúa aumentando mientras el jugador siga aplastando enemigos. El objetivo es acumular la mayor cantidad de puntos posible antes de ser tocado por un enemigo.

Este código gestiona la puntuación del jugador. La variable score inicia en 0 y se incrementa cada vez que el jugador aplasta un mob. La función _on_Mob_squashed se ejecuta cuando un mob es aplastado, aumentando la puntuación en 1 y actualizando el texto del Label para mostrar el nuevo valor de la puntuación con el formato "Score: X", donde X es el valor actual de score.

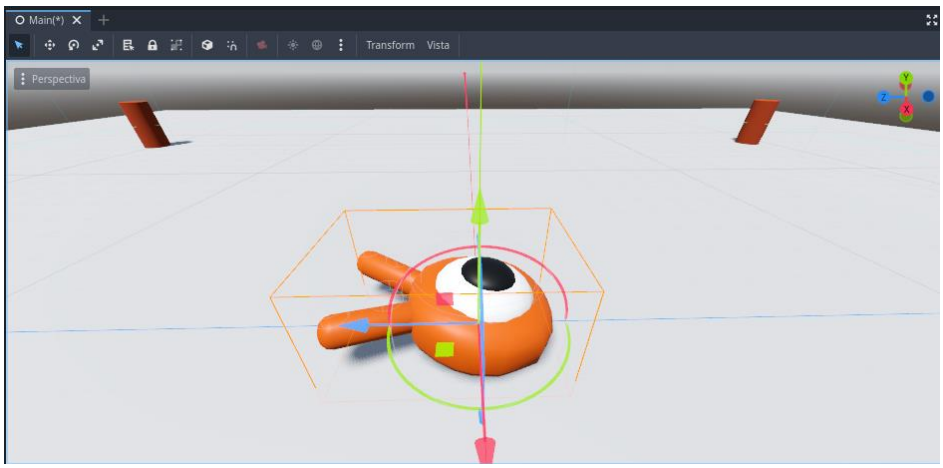
```
1   extends Label
2   var score = 0
3   func _on_Mob_squashed():
4       score += 1
5       text = "Score: %s" % score
```

Finalización del Juego: El juego termina cuando el jugador es tocado por un enemigo. En ese momento, el jugador escucha una breve animación sonora de derrota, y el juego se reinicia automáticamente, llevando al jugador de nuevo a la pantalla de inicio. Desde allí, el jugador puede presionar la barra espaciadora para comenzar una nueva partida.

Sonido: El juego está acompañado por una pista de música de fondo que suena continuamente durante la partida. La música tiene un ritmo constante que mantiene la energía alta mientras el jugador juega, ayudando a crear un ambiente entretenido. No se incluyen efectos de sonido adicionales, lo que hace que el enfoque principal sea la jugabilidad y la música.

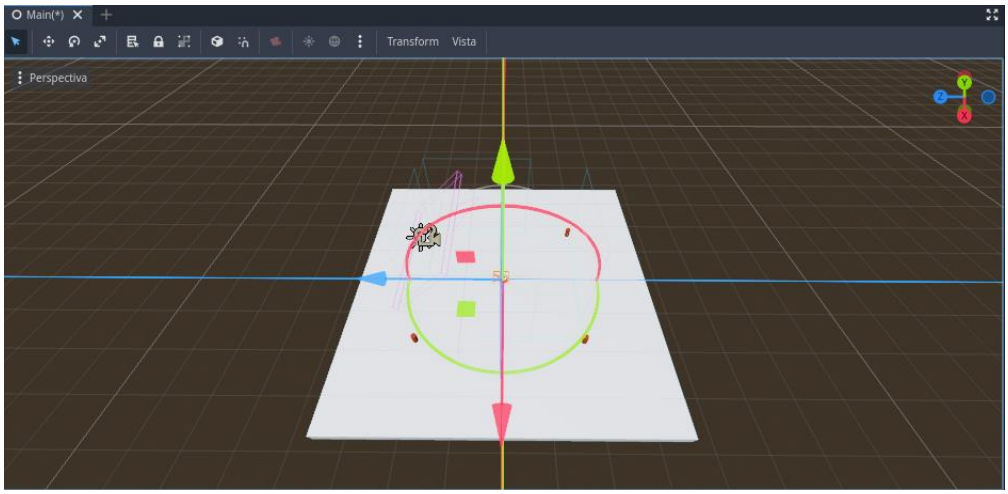
Personajes y Mundo del Juego

Personaje: El personaje principal es un modelo 3D animado, creado con Blender, que puede moverse libremente en el entorno tridimensional del juego. El diseño del personaje es simple pero efectivo, asegurando que sea fácilmente identificable en el juego. El personaje está animado para realizar el salto, lo que permite al jugador ver claramente el momento en que está listo para aplastar a un enemigo.



Enemigos: Los enemigos en Squash the Creeps también son modelos 3D, diseñados para ser fácilmente visibles en el entorno 3D. Aparecen de manera aleatoria en diferentes puntos del mapa y se mueven de forma impredecible, lo que aumenta la dificultad y el desafío del juego. Los enemigos tienen un tamaño adecuado para ser aplastados al saltar, pero el jugador debe tener buen timing para lograrlo.

Entorno y Mapas: El mapa del juego está diseñado en 3D con la ayuda de Blender. Es un espacio amplio, pero limitado, para que el jugador se mueva y trate de aplastar a los enemigos. El mapa tiene una apariencia sencilla pero efectiva, con suficiente espacio para moverse y evitar a los enemigos, pero sin demasiados obstáculos para hacer la jugabilidad más fluida y directa. El mapa está diseñado para mantener al jugador enfocado en las mecánicas de movimiento y salto, con un estilo visual simple que permite una buena visibilidad de los personajes y enemigos.



Historia

Squash the Creeps es un juego sin una narrativa profunda. El jugador simplemente asume el rol de un personaje cuya misión es sobrevivir el mayor tiempo posible aplastando a los enemigos que aparecen aleatoriamente en el mapa. La historia detrás del juego podría ser que el personaje está siendo atacado por criaturas extrañas (los "Creeps") y debe eliminar a todos los que se le crucen para seguir sobreviviendo. Sin embargo, el enfoque principal del juego es la jugabilidad rápida y desafiante, sin una historia detallada.

Interfaz de Usuario (UI)

Pantalla de Inicio: La pantalla de inicio es simple y clara. El jugador ve un mensaje que le indica presionar la barra espaciadora para comenzar la partida. Esta pantalla minimalista permite que el jugador se enfoque rápidamente en el inicio del juego.

Pantalla de Juego: Durante el juego, la puntuación se muestra de forma prominente en la parte superior de la pantalla. Esta puntuación aumenta cada vez que el jugador aplasta a un enemigo, permitiendo que el jugador siga su progreso en tiempo real.

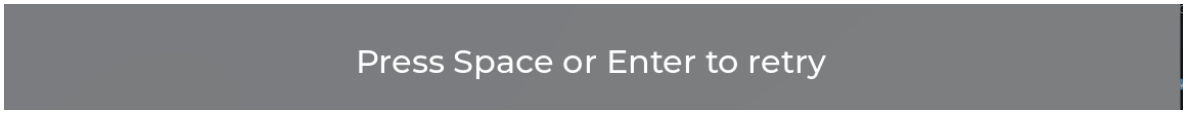
Pantalla de Fin de Juego: Cuando el jugador pierde, se muestra la pantalla de inicio nuevamente, donde podrá presionar la barra espaciadora para comenzar una nueva partida. No se muestra una pantalla de "game over" compleja, ya que el enfoque está en la acción constante del juego.

Controles

Movimiento del Personaje: El jugador usa las teclas de flecha del teclado para mover al personaje en las direcciones arriba, abajo, izquierda y derecha. El control es rápido y preciso, permitiendo al jugador moverse rápidamente en cualquier dirección dentro del espacio 3D.

Salto: La barra espaciadora se utiliza para saltar. El salto es la acción clave para eliminar a los enemigos, y el jugador debe usarlo estratégicamente para aplastarlos sin ser tocado por ellos.

Botón de Inicio: La barra espaciadora también se utiliza en la pantalla de inicio para comenzar el juego. El juego se inicia inmediatamente después de que el jugador presiona la barra espaciadora.



Gráficos y Sonido

Gráficos: El estilo gráfico 3D, creado con Blender, permite una experiencia visual simple pero efectiva. Los modelos de personajes y enemigos están diseñados de

forma clara para que el jugador pueda interactuar con ellos sin dificultad. La perspectiva en tercera persona y la iluminación del mapa contribuyen a una jugabilidad fluida.

Sonido: La música de fondo mantiene un ritmo constante durante toda la partida, contribuyendo a la atmósfera del juego. No hay efectos de sonido adicionales, lo que hace que el jugador se concentre en la música y la acción en pantalla.

Tecnologías y Herramientas Utilizadas

Motor de Juego: Godot Engine 3D, que permite crear entornos tridimensionales y manejar la lógica del juego.

Lenguaje de Programación: GDScript, utilizado para implementar la lógica de movimiento del personaje, aparición de enemigos y la puntuación.

Gráficos 3D: Blender para la creación y animación de los modelos 3D del personaje y los enemigos.

Sonido: Pistas musicales en formato compatible con Godot (como .ogg o .wav) para la música de fondo.

Desarrollo y Progreso

El juego está completamente funcional. El movimiento, salto y la aparición aleatoria de enemigos funcionan correctamente. El jugador puede acumular puntuación y el juego finaliza cuando es tocado por un enemigo.

Se pueden agregar nuevas mecánicas, como enemigos con habilidades especiales o un modo de dificultad creciente, que podría hacer que el juego se vuelva más desafiante conforme se avanza.

Conclusión

Squash the Creeps es un juego 3D que pone a prueba las habilidades de reacción y coordinación del jugador. La mecánica simple de moverse y saltar para aplastar enemigos hace que el juego sea accesible, pero lo desafiante está en manejar la creciente dificultad a medida que más enemigos aparecen. La experiencia de desarrollo en Godot y Blender me permitió aprender sobre la creación de mundos tridimensionales, la animación de personajes y la integración de sonidos, lo que ha sido una experiencia enriquecedora y educativa.