



# Tecnológico de Monterrey

Instituto Tecnológico de Estudios Superiores de  
Monterrey

## *Actividad de Diseño del Juego: Construcción de Software y Toma de Decisiones*

Campus Santa Fe

*Matías Méndez A01422885*

*Eduardo Galindo A01028846*

*Fernanda Nava Moya A01023896*

# Diseño del Juego

**Nombre del juego:** Virus Attack

**Nombre del equipo:** FEM Tech

## ***Summary***

El objetivo de este videojuego es desafiar al jugador a enfrentarse a una carrera de obstáculos ambientada dentro del cuerpo humano donde se pone a prueba conocimientos de biología básicos, a la par de habilidades cognitivas.

## ***Gameplay***

El objetivo es llegar a la meta sin perder vidas y agarrar el mayor número de puntos posible. Con base en esto se busca medir la agilidad mental y toma de decisiones del jugador mientras se enfrenta a tanto obstáculos como bonus (óxido nítrico), incrementando la dificultad del videojuego nivel a nivel.

## ***Background and style of the game***

La molécula de oxígeno, se deberá enfrentar a distintos virus a través de su recorrido por el cuerpo humano hasta llegar a la mitocondria de una célula para así poder evitar la infección de un cuerpo humano. El objetivo del héroe (en este caso la molécula o asimismo, del jugador) es **¡Salvar una vida!**

## ***Mindset***

El jugador debe sentir la presión de llegar al final del nivel obteniendo la mayor cantidad de bonus posibles, desatando emociones como nervios y obligándolo a concentrarse para mejorar sus puntuaciones.

## ***Game genre***

El juego es de tipo acción de un jugador en tercera persona. El fondo, los enemigos y el piso se desplazan con el personaje.

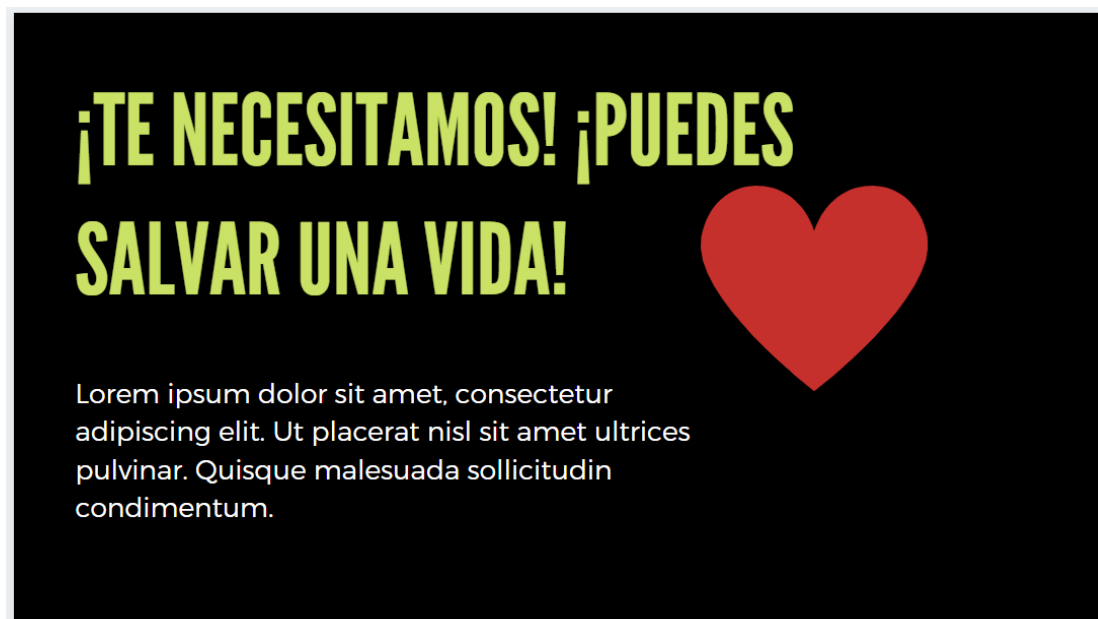
## Technical:

### Screens

- Escena de título



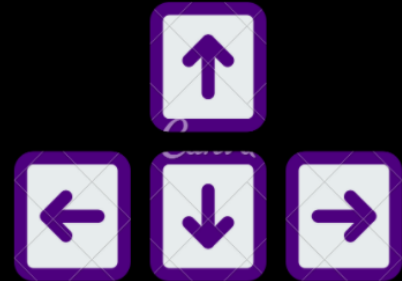
- Introducción del juego



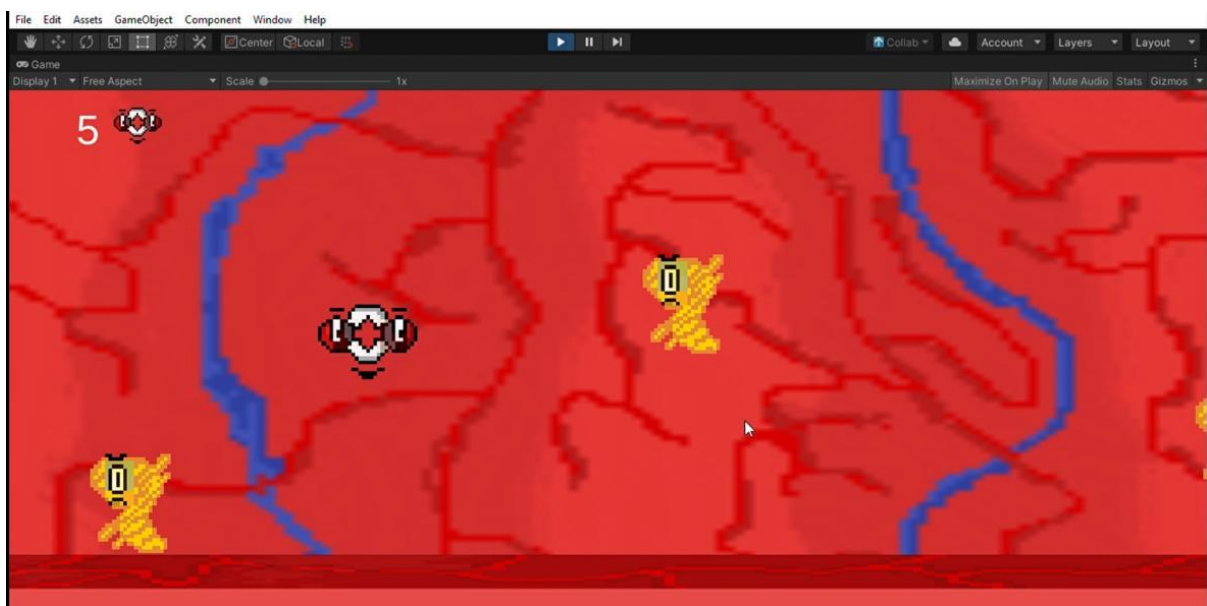
- Instrucciones

# INSTRUCCIONES

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut placerat nisl sit amet ultrices pulvinar. Quisque malesuada sollicitudin condimentum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut placerat nisl sit amet ultrices pulvinar. Quisque malesuada sollicitudin condimentum.



- Game



- Assesment/Next Level

**¡FELICIDADES!  
AVANZASTE AL  
SIGUIENTE NIVEL**

### **DESCRIPCIÓN DEL NIVEL**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut placerat nisl sit amet ultrices pulvinar. Quisque malesuada sollicitudin condimentum. Vestibulum nec dolor a magna semper consequat sit amet nec enim.

**¡UPS! MÁS SUERTE A LA PRÓXIMA**



**VUELVE A INTENTARLO**

- Puntaje y métricas

<b>PUNTAJE FINAL</b> 10 puntos	<b>NÚMERO DE PARTIDAS JUGADAS</b> 10 partidas
<b>ESTRATEGIA</b> 10 puntos	<b>PUNTAJE PROMEDIO DE TODOS LOS JUGADORES</b> 10 partidas
<b>AGILIDAD MENTAL</b> 10 puntos	

- Información de STEM

<b>CARRERAS STEM</b>	
<b>DESCRIPCIÓN</b> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut placerat nisl sit amet ultrices pulvinar. Quisque malesuada sollicitudin condimentum. Vestibulum nec dolor a magna semper consequat sit amet nec enim.	
<b>SUELDOS</b> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut placerat nisl sit amet.	<b>EMPLEABILIDAD</b> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut placerat nisl sit amet.

---

**Controls**

\_\_\_\_\_El jugador mueve al personaje principal moviéndose de arriba abajo haciendo click o presionando la tecla de espacio, debe de controlar la gravedad definida del juego para esquivar los obstáculos y alcanzar los bonus.

## ***Mechanics***

- Cuando se presiona el click o la barra espaciadora el personaje principal eleva su posición 3 usando la función `vector.up`
- Cuando el personaje principal choca contra el suelo, se le resta 1 a la variable “puntuación”. Esto se hace mediante la función “`OnCollisionEnter2D(Collision2D collision)`” que activa otra función dentro de la clase para el personaje principal disminuyendo la puntuación.
- Cuando el personaje atraviesa las áreas de incrementar puntuación, se activa una función para que suma uno a la variable puntuación dentro de la clase personaje.
- Cuando el personaje atraviesa las áreas de disminuir puntuación, se activa una función para que reste uno a la variable puntuación dentro de la clase personaje.
- La puntuación aparece en la pantalla por medio de un canvas que referencia la puntuación directamente de la clase personaje.
- Los enemigos se mueven hacia la izquierda usando la función `vector3.left`

El puntaje del jugador siempre se muestra en la parte superior del juego. Cuando el jugador colisiona con el virus se le restan puntos al puntaje total y la partícula de oxígeno (el jugador) rebota contra estos. De igual manera, cuando se tocan score extras, estas afectarán directamente a la variable de score dentro del Game object del objeto.

Se colocarán enemigos en posiciones definidas para que el juego incremente su dificultad conforme avance, de igual manera, la velocidad de los virus se incrementará conforme el usuario avance en niveles. Esto se realizará con un script que altera la posición del virus de forma constante.

## **Level design**

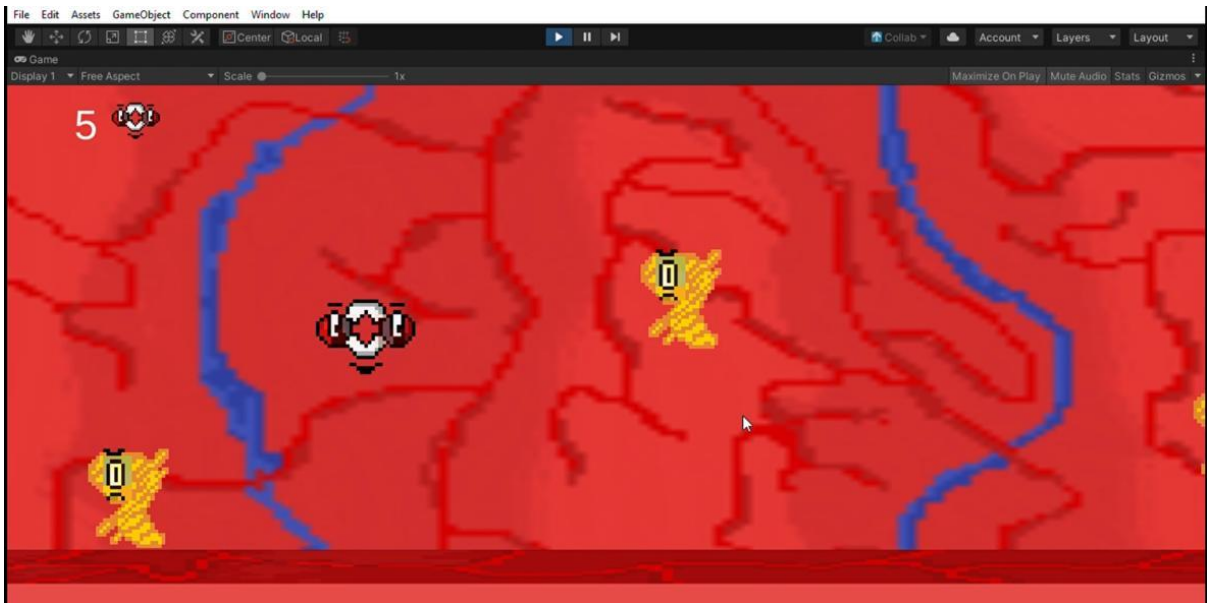
### ***Themes***

El juego está inspirado dentro del cuerpo humano, desplazándose a través de un vena con el objetivo de alcanzar la mitocondria de una célula.

1. Venas

a. Ambiente

- i. Rojo sangre, interior del cuerpo.



b. Objetos

- i. Piso, tejido de la vena



- ii. Grasa, rayo amarillo



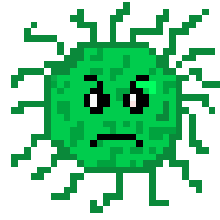
- iii. Pared, tejido de la vena

c. Interactivos

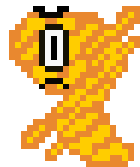
- i. Virus

1. verdes

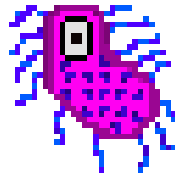




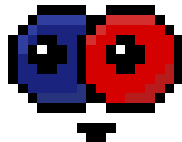
2. naranjas



3. morado



ii. Bonus, molécula de óxido nítrico



### **Game Flow**

1. Aparece el menú principal.
2. Nivel 1.
3. Nivel 2.
4. Nivel 3.
5. Puntajes.
6. Información de carreras STEM.

# Development

## ***Abstract Classes***

- logic\_character (lógica de movimiento por controles)
- logic\_obstacle (velocidad obstáculo)
- logic\_scenes (controla score y cierra escenas si pierdas)
- logic\_scoreArea (incrementa score en choque con bonus)

## **Derivated Classes**

- logic\_score (Cambia texto de marcador)
- logic\_obstacleGenerator

# Graphics

## ***Style Attributes***

El color principal que vamos a utilizar va a ser el color rojo ya que la temática es adentro de la vena. Tanto el suelo como la pared va a ser representados por los tejidos de la vena. De igual manera, la partícula de oxígeno será representada por el color rojo. Sin embargo, tanto la forma y color de los virus varía. Los colores para estos son naranja, morado y verde. Esto es para que el jugador pueda identificar de manera más rápida los "enemigos" del juego y así llegar a tomar una decisión. Todo esta diseñado con pixel art.

## ***Graphics Needed***

- Personajes
  - Molécula de oxígeno
  - Enemigos
    - Green virus
    - Yellow virus
    - Purple virus
- Escenario
  - Piso de tejido

- Fondo de tejido
- Grasa (obstáculo)

## **Sounds Needed**

1. Efectos
  - a. Cuando la partícula de oxígeno brinca
2. Retroalimentación
  - a. Cuando pierde puntos: choca con virus o piso
  - b. Cuando gana puntos: bolitas
  - c. Cuando empieza y acaba el nivel

## **Music Needed**

1. Canción de la mitocondria de fondo.

## **Schedule**

1. Diseño general de videojuego
  - a. ilustraciones
  - b. historia
  - c. inspiraciones
2. Hacer diagrama de actividades y casos de uso
3. Crear modelos en pixel art
4. Crear ambiente
5. Definir clases
  - a. Obstáculos: virus, etc,
  - b. Personaje: partícula de oxígeno
  - c. Clase de objetos que se mueven
6. Definir subclases
  - a. Virus 1, 2, 3.

7. Implementar código para cada mecánica
8. Implementar menú de inicio
9. Diseñar niveles
10. Implementación de puntos extra
11. Definir e implementar sonidos
12. Definir e implementar música