| Texto  Descripción generada automáticamente con confianza media | TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS  FACULTAD DE INGENIERÍA  Universidad Nacional de Jujuy |  |
| --- | --- | --- |



Punto 1:

| Tarjeta de Historia de Usuario | |
| --- | --- |
| Nombre de Historia de Usuario: Uso de Listas, Colisiones y DeltaTime en Snake | |
| Código de Usuario: US034 | Usuario: Jugador |
| Prioridad: Alta | Riesgo de Desarrollo: Medio |
| Estimación: 2 horas | Iteración Asignada: Iteración 2 |
| Responsable: Jimenez | |
| Descripción:  Como desarrollador de la aplicación de Snake,  Quiero implementar el uso de listas para almacenar los animales comidos, detectar colisiones y usar deltaTime para normalizar la velocidad,  Para mejorar la experiencia de juego y agregar funcionalidades adicionales. | |
| Criterios de Aceptación:   1. Almacenamiento de Animales Comidos:Acción: Crear una lista para almacenar los animales que ha comido la serpiente. 2. Detección de Colisiones:Acción: Implementar la detección de colisiones entre la cabeza de la serpiente y los animales. 3. Uso de DeltaTime:Acción: Utilizar deltaTime para normalizar la velocidad de movimiento de la serpiente. 4. Puntaje y Tiempo de Juego:Acción: Calcular el puntaje del jugador basado en los animales comidos y el tiempo de juego. 5. Uso de Imágenes:Acción: Utilizar imágenes para representar a la serpiente y los diferentes tipos de animales. 6. Encapsulamiento y Buenas Prácticas:Acción: Aplicar encapsulamiento a los atributos de las clases y seguir buenas prácticas de programación. | |

Punto 2:

| Tarjeta de Historia de Usuario | |
| --- | --- |
| Nombre de Historia de Usuario: Uso de Listas, Colisiones y DeltaTime en Tanque vs Muros | |
| Código de Usuario: US035 | Usuario: Jugador |
| Prioridad: Alta | Riesgo de Desarrollo: Medio |
| Estimación: 3 horas | Iteración Asignada: Iteración 2 |
| Responsable: Jimenez | |
| Descripción:  Como desarrollador del juego del tanque disparando a los muros,  Quiero implementar el uso de listas para almacenar los muros y balas, detectar colisiones y usar deltaTime para normalizar la velocidad,  Para crear un juego desafiante y realista. | |
| Criterios de Aceptación:   1. Almacenamiento de Muros y Balas:Acción: Crear una lista para almacenar los muros y otra lista para almacenar las balas disparadas por el tanque. 2. Detección de Colisiones:Acción: Implementar la detección de colisiones entre las balas y los muros. 3. Uso de DeltaTime:Acción: Utilizar deltaTime para normalizar la velocidad de movimiento de las balas y el tanque. 4. Movimiento del Tanque:Acción: Permitir que el tanque se mueva horizontalmente usando las teclas de flecha. 5. Disparar Balas:Acción: Permitir que el tanque dispare balas al presionar una tecla. 6. Uso de Imágenes:Acción: Utilizar imágenes para representar al tanque y los muros. 7. Encapsulamiento y Buenas Prácticas:Acción: Aplicar encapsulamiento a los atributos de las clases y seguir buenas prácticas de programación. | |

Conclusión

Párrafos de las conclusiones

Fuentes bibliográficas

Se deben enunciar las fuentes (apuntes de la materia, páginas web, videos de youtube, libro (nombre, autor)