



ESCUELA  
NACIONAL  
DE ESTUDIOS  
SUPERIORES  
  
UNIDAD MORELIA

## **Administración de proyectos**

Profesor: Héctor Alonso Guzmán Gutiérrez

### **Requerimientos**

ENESoftware

Jorge Antonio Camarena Pliego  
David Calderón Ceja  
Keshava Tonathiu Sánchez Barbosa  
Stephany Dzoara Vargas Mier

26/09/2019

**Semestre 2020-1**

Versión 0.2

# Índice

<b>Introducción</b>	<b>3</b>
<b>Descripción de Requerimientos</b>	<b>3</b>
Requisitos Funcionales	3
Requisitos No Funcionales	4
Identificación de Casos de Uso	4
Detalle de Casos de Uso	5
<b>Mockups</b>	<b>6</b>
Dentro de un Nivel del Juego	6
En el Menú de Selección de Nivel	6
<b>Capacitación en el uso de Unity</b>	<b>7</b>
Habilidades obtenidas de la capacitación	9

## Introducción

El juego, de nombre *SoundingSafe*, será un juego de plataformas en que el jugador protagonista pretende salvar a su mundo del silencio total. Tendrá a su disposición ciertos poderes y bonuses que podrá obtener a lo largo de cada nivel en la forma de notas e instrumentos musicales.

Al jugador se le presentará un resumen visual de la historia y se le pondrá en control del protagonista del juego. El protagonista buscará llegar al final del nivel a tiempo antes de que la música del escenario se apague o termine, siendo posible reanimarla con algún checkpoint o poder. Si llega al último nivel y lo completa, habrá ganado todo el juego, presentándose el final de la historia visualmente y los créditos.

## Descripción de Requerimientos

### Requisitos Funcionales

1. El jugador podrá elegir en el menú del juego de entre los niveles que haya desbloqueado el que desee jugar.
2. Los niveles se desbloquean secuencialmente conforme el jugador complete los anteriores al que vaya a desbloquearse.
3. El jugador podrá mover su personaje hacia la izquierda o derecha de la pantalla con las teclas “W” y “D” respectivamente.
4. El jugador podrá hacer que su personaje salte con la tecla de barra espaciadora.
5. Todo nivel tendrá una longitud de diez pantallas.
6. Cada nivel tendrá un solo checkpoint en algún punto del nivel.
7. Mientras esté en ejecución un nivel, se mostrará en la interfaz de usuario el tiempo restante en la esquina superior derecha.
8. Los checkpoints aumentan en cierta cantidad arbitraria el tiempo restante.
9. El jugador puede reiniciar el nivel en ejecución si lo desea, anulando todos los checkpoints desbloqueados.
10. El jugador podrá salir al menú del juego si desea salirse del nivel actual.
11. El jugador podrá resumir su progreso en un nivel a partir del checkpoint más reciente desbloqueado.
12. El jugador, al saltar, no podrá volver a saltar en el aire hasta que vuelva a tocar una plataforma.
13. Existirán objetos de salto doble que permitirán saltar en el aire una vez exclusiva tras haber obtenido alguno de estos objetos.
14. Si el tiempo restante llega a cero, el jugador perderá el nivel y deberá reiniciarlo completamente.

15. El puntaje obtenido en un nivel está determinado de manera acumulativa por cada objeto de puntaje adicional obtenido por el jugador a lo largo del nivel.
16. El juego llevará cuenta del mejor puntaje y tiempo obtenido en un nivel dado.
17. El suelo del juego tendrá agujeros peligrosos por los cuales puede caerse de la pantalla

### Requisitos No Funcionales

1. El menú principal del juego mostrará el logo de *SoundingSafe*.
2. Al iniciarse el juego, se mostrará el logo de *Unity* y el logo de *ENESoftware*.
3. Se pretende que el juego tenga ausencia de errores o bugs tanto de jugabilidad como visuales o auditivos.
4. Se pretenderá cumplir el objetivo de un producto final que abarque una cantidad igual o menor a 150MB aproximadamente en disco.
5. Las pantallas de carga del juego serán relativamente rápidas.
6. La dificultad de los niveles posteriores será progresivamente más alta rumbo al último nivel del juego.
7. La intensidad de la música viene ligada a la cantidad de tiempo restante en el nivel. Será más intensa y estará más animada conforme quede más tiempo restante.
8. El personaje tendrá una animación de muerte al fallar el nivel.
9. Cada nivel tendrá un tono neón distinto en los bordes de los objetos.
10. El final del nivel será representado visualmente como un portal.
11. El fondo de los escenarios tendrá un aspecto hipnótico o de ilusión visual.
12. Las plataformas serán representadas como bloques de ladrillos.
13. Los objetos de puntaje adicional serán representados como corcheas simples
14. Los checkpoint serán representados con claves de Sol.
15. Los objetos de salto doble serán representados como corcheas dobles.

### Identificación de Casos de Uso

ID	Funcionalidad	ID	Funcionalidad
C-1	Tiempo llega a cero	C-7	Muerte
C-2	Jugador llega al final del nivel	C-8	Finalización del nivel
C-3	Jugador cae de la pantalla		
C-4	Jugador corre a través del escenario		

C-5	Jugador salta		
C-6	Jugador llega a un checkpoint		

### **Detalle de Casos de Uso**

#### *C-1: Tiempo llega a cero*

Cuando el tiempo llega a cero, el personaje entrará en el caso de uso de muerte (C-7) y el contador de tiempo, ahora en 00:00, se tornará color rojo.

#### *C-2: Jugador llega al final de nivel*

Cuando el jugador llega al final del nivel antes de agotarse el tiempo, se reproduce un tono de sonido musical corto para indicar éxito y se entrará en el caso de uso de finalización del nivel (C-8)

#### *C-3: Jugador cae de la pantalla*

Cuando el jugador cae de la pantalla, como por un hoyo en el suelo o plataforma, se entrará en el caso de uso de muerte (C-7)

#### *C-4: Jugador corre a través del escenario*

En este caso, al personaje se le aplica una fuerza horizontal y entra en una animación de correr, así como se muestra el movimiento de todo el entorno en sincronización con el personaje.

#### *C-5: Jugador salta*

En este caso, el personaje entra en una animación de salto y se le aplica una fuerza vertical

#### *C-6: Jugador llega a un checkpoint*

En el caso de llegar a un checkpoint, se guardará como un posible punto de partida en caso de reiniciarse el nivel, así como se agregará cierta cantidad de tiempo adicional al contador de la esquina superior derecha de la pantalla.

#### *C-7: Muerte*

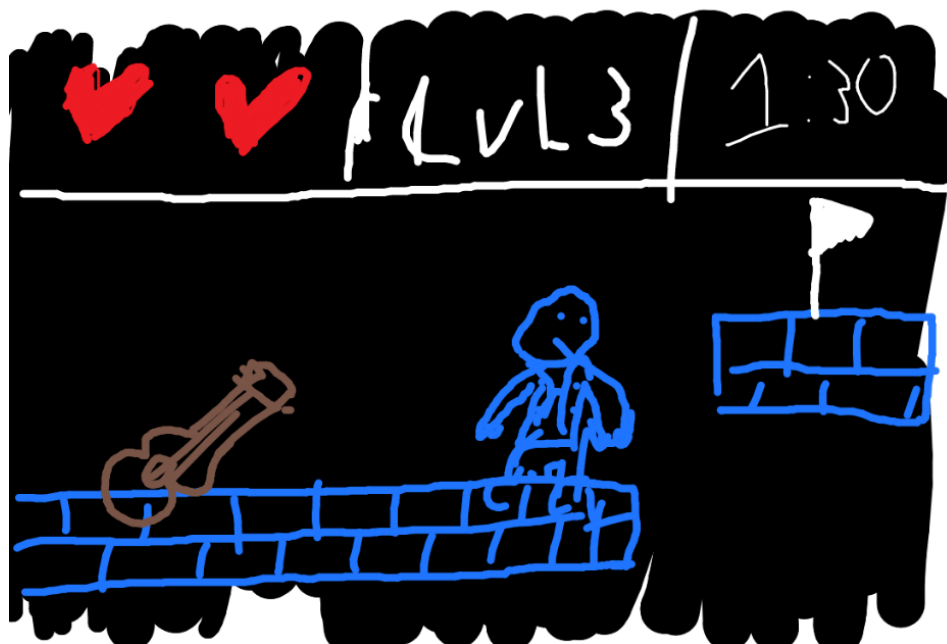
En el caso de morir el personaje, se entrará en una animación especial de muerte, así como perder una vida, reflejado como una resta de una unidad en el contador de vidas, ubicado en la parte superior izquierda de la pantalla. Este contador está reflejado en cantidad de sprites de corazones en esta sección de la pantalla.

#### *C-8: Finalización del nivel*

En el caso de finalizar el nivel, se desplegará en grande el puntaje acumulativo obtenido al momento de la finalización, así como el tiempo logrado y se mostrará un menú con las siguientes opciones: Regresar a último checkpoint, Reiniciar nivel y Regresar a pantalla principal del juego. Si el tiempo logrado es menor que el récord registrado para el nivel o este está en un “Na”, se procederá a sobrescribir este valor con el nuevo. Si el siguiente nivel consecutivo no se ha desbloqueado aún, se volverá opción disponible ahora y no tendrá marcado “locked!”, sino “Na” de que aún no se tiene registrado ningún tiempo récord para dicho nivel.

## Mockups

### Dentro de un Nivel del Juego

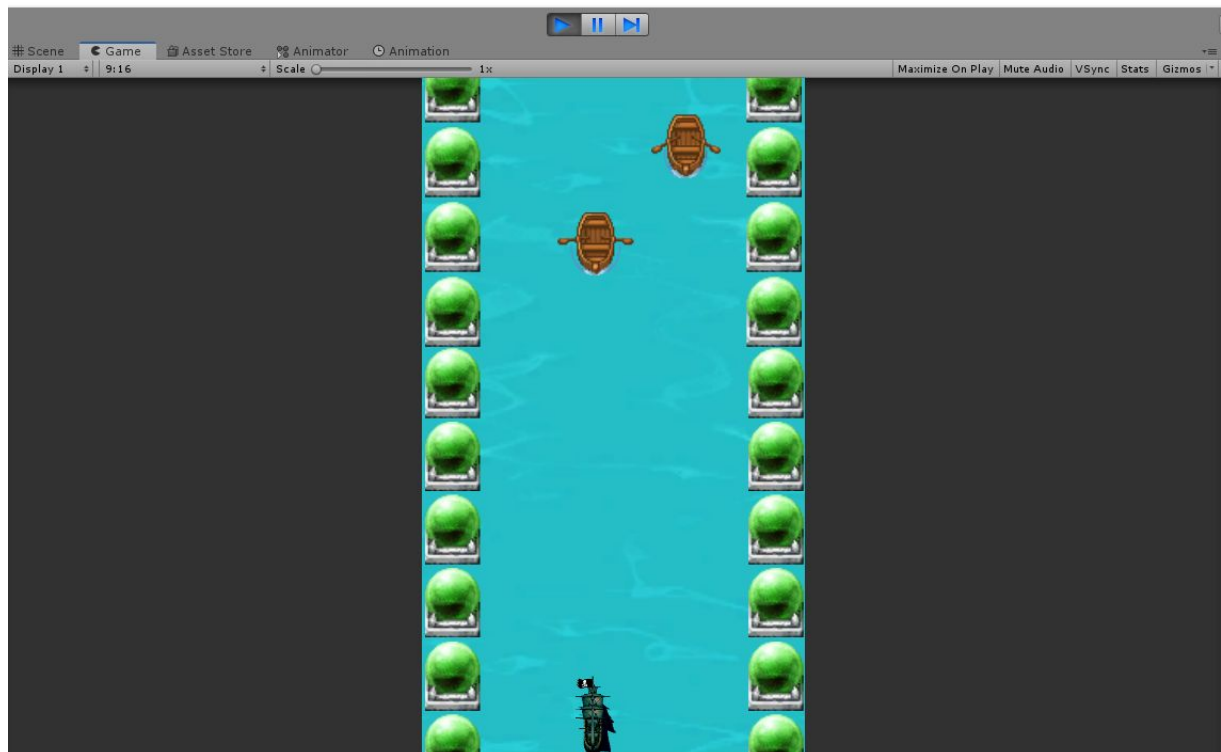
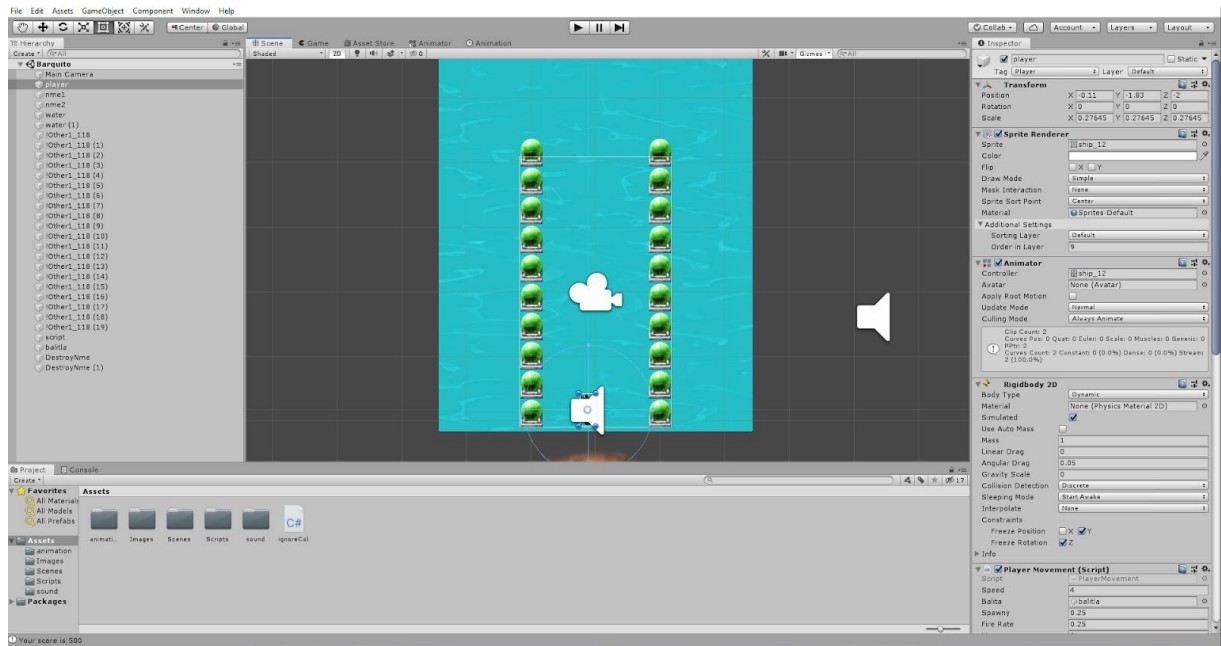


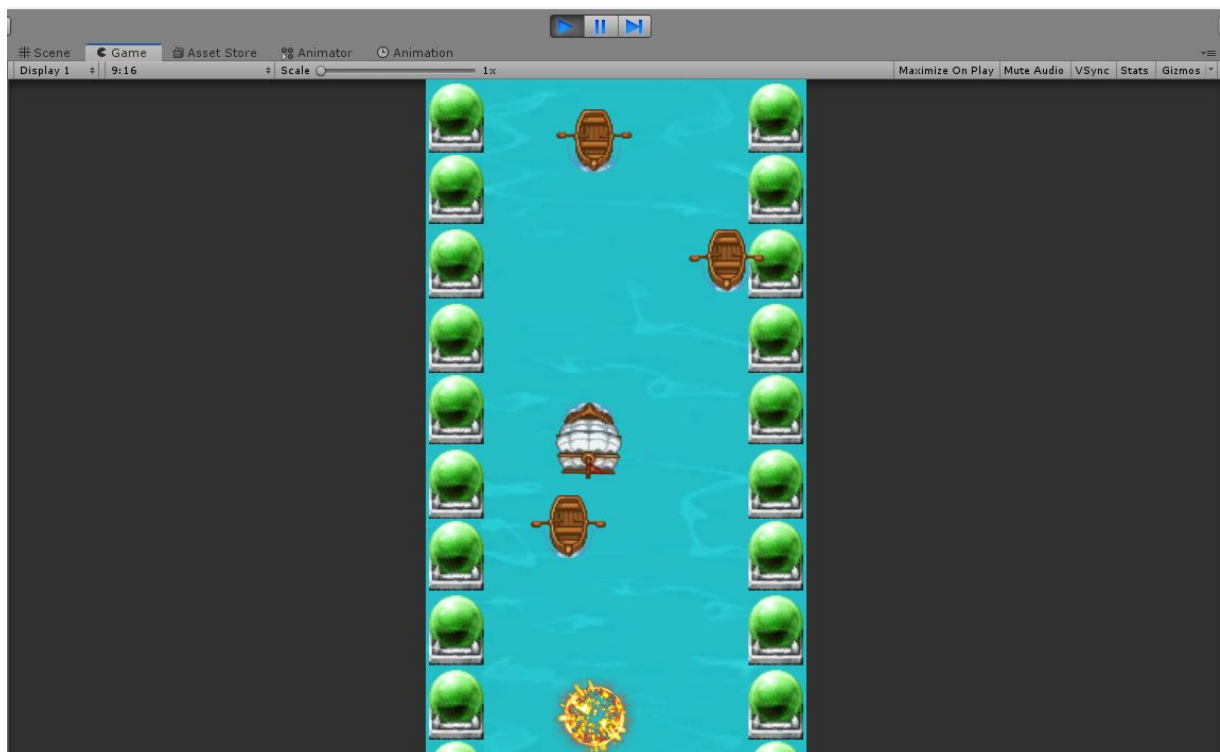
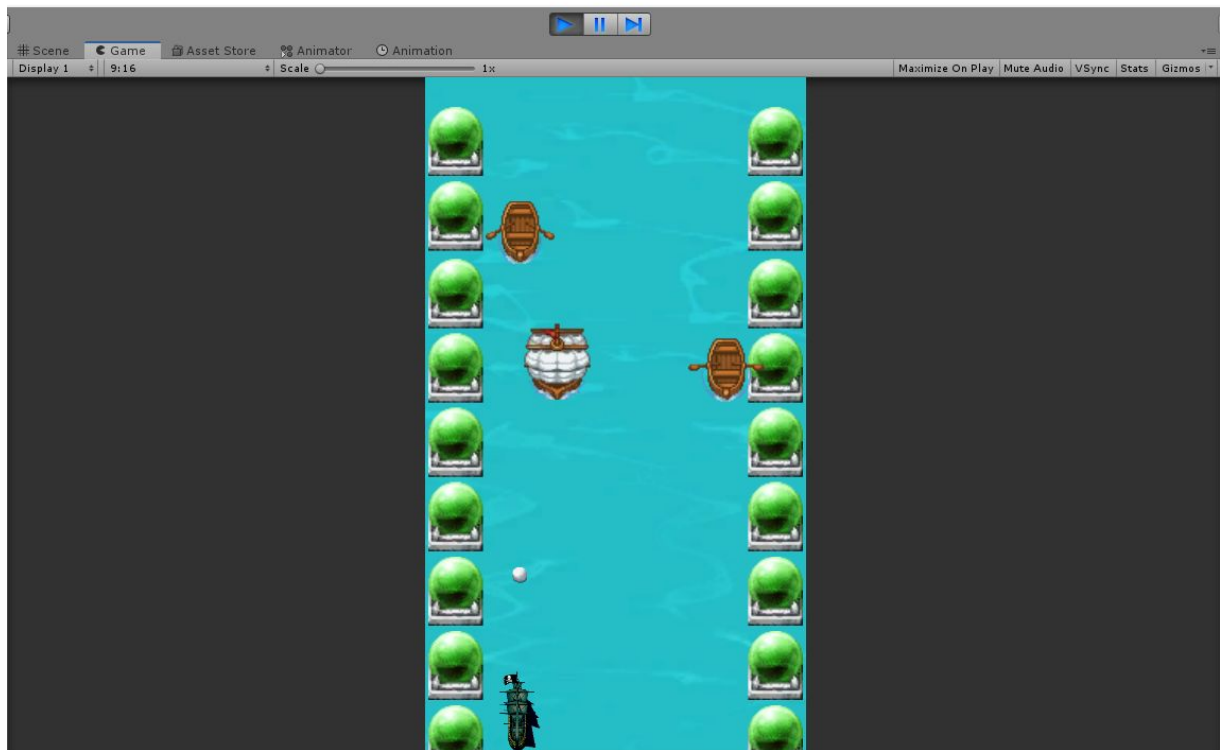
### En el Menú de Selección de Nivel



## Capacitación en el uso de Unity

Durante las últimas 2 semanas llevamos a cabo una capacitación en C# y la plataforma unity para desarrollar un juego mas simple de lo que queremos presentar en el proyecto, a continuación de muestran capturas de pantalla del juego en cuestión y posteriormente algunas breves descripciones de las utilidades que se aprendieron de la capacitación





El juego tiene como género “shoot em’ up” donde el propósito es destruir todos los enemigos posibles sin que destruyan tu barco, cuenta con 3 vidas y la capacidad de disparar.



## **Habilidades obtenidas de la capacitación**

Gracias a la capacitación se obtuvieron los siguientes conocimientos:

- Familiarización con C#
- Familiarización con Unity
- Instanciación de objetos en C# y Unity
- Técnica de “infinite scrolling”
- Técnica de duplicación e instanciación de objetos en tiempo de ejecución
- Detección de inputs de un teclado y mouse
- Reproducción de audio de múltiples fuentes
- Animación de sprites
- Recorte de sprites
- Uso de procesos en paralelo en C#