

## Dungeon Quest AR

### Crear nivel:

- Para crear de forma procedural el nivel se utilizaron varios prefabs con muros en diversas posiciones.
- La generación se hace mediante un script que recibe todos los prefabs de muros.
- Se instancian los contornos del nivel (esquinas y paredes laterales)
- De forma aleatoria se rellena el mapa para generar el resto de los muros con instancias de todas las opciones de prefabs
- Después por medio de scripts y varias reglas de posicionamiento se cambian los muros que forman callejones sin salida por puertas o por pedazos de suelo sin muro
- Una vez que se crean los muros por cada prefab que contiene el mapa se instancian de forma aleatoria los enemigos y cofres, se pueden instanciar:
  - Enemigo1
  - Enemigo2
  - Enemigo3
  - Cofre
  - Sin enemigos
- Finalmente, una de las piezas se reemplaza por la pieza de salida en una posición aleatoria.
- La pieza de salida tiene una estatua, esta a su vez tiene un trigger. Cuando el jugador está en el trigger aparecerá el botón de terminar nivel, si el jugador sale del trigger entonces el botón se desactivará.
- No se consideró la gravedad ya que no se utilizarán acciones como el salto así que no son necesarios colliders en el suelo.



Figura 1. Prefabs que conforman el mapa

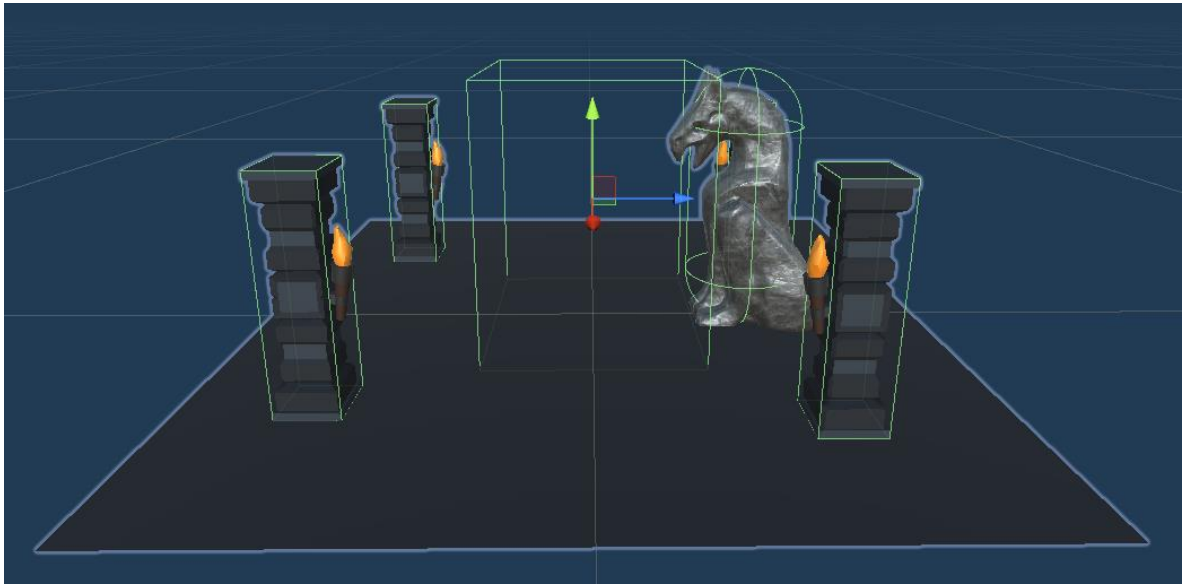


Figura 2. Trigger de la estatua y colliders en el prefab exit.

## GUI:

- Para la barra de vida y de stamina se crearon dos sliders, estos sliders mediante código se conectan a la variable currentHP y currentStamina en el script de PlayerMovement, además de hacer el calculo de porcentaje a mostrar dependiendo de la vida actual y el stamina actual.
- Para crear el Joystick se utilizó un asset gratuito en Unity llamado Joystick Pack, para implementarlo en el código simplemente se crea una variable Joystick que estará asociada al Joystick de la GUI.
- Para el botón de defensa (Y) se creó un script nuevo de botón y con la función OnPointerDown conseguimos poder ejecutar el código mientras el botón este presionado y con OnPointerUp ejecutamos otra función cuando el botón deje de presionarse.
- Se pusieron otros 3 botones normales para atacar, usar poción de stamina y poción de vida.
- Los botones de poción van conectados a un diccionario que funciona como el inventario del jugador.



Figura 2. Elementos de la GUI

## AR

- Para implementar la tecnología de realidad aumentada se utilizó vuforia y Unity
- Para poder utilizar Vuforia se debe descargar el paquete directamente desde la página de Vuforia, además se debe hacer una cuenta y crear una licencia gratuita, el número de licencia de esta licencia se debe copiar y pegar en la carpeta Resources que se crea en los assets cuando se importa Vuforia
- Se crea una base de datos con las imágenes que se van a utilizar, esto en la misma página de Vuforia y también ahí mismo te dan una calificación según la facilidad con la que se puede reconocer la imagen.
- Para utilizar Vuforia en Unity se borra la cámara principal y se reemplaza por un ARCamera en la pestaña de Vuforia en creación de objetos.
- Para detectar la imagen (en este caso el logo de la fi) se crea un Image Target también desde la pestaña de Vuforia y para que el objeto aparezca cuando se detecte simplemente se debe colocar como hijo de este GameObject.
- En este proyecto se utilizó la función OnTargetFound() del Image Target para activar el Plane Finder, de esta forma solo podremos instanciar el mapa una vez que hayamos escaneado la imagen y de esta forma saber que temática va a tener el nivel que se va a generar.
- Los prefabs se instancian como hijos de un Ground Plane Stage que se coloca junto con el Plane Finder.

### License Manager

Get Basic

Buy Premium

Buy Cloud Add On

[Learn more](#) about licensing.  
Create a license key for your application.

Search

Name	Primary UUID <sup>①</sup>	Type	Status <sup>▼</sup>	Date Modified
practica8	N/A	Basic	Active	Nov 19, 2023
o	N/A	Basic	Active	Nov 06, 2023

Figura 3. Licencias de Vuforia.

**Target Manager** Add Database

Use the Target Manager to create and manage databases and targets.

Search

Database	Type	Targets	Date Modified
cuaderno	Device	1	Nov 07, 2023
practica8	Device	1	Nov 19, 2023

Figura 4. Bases de datos de imágenes en Vuforia.

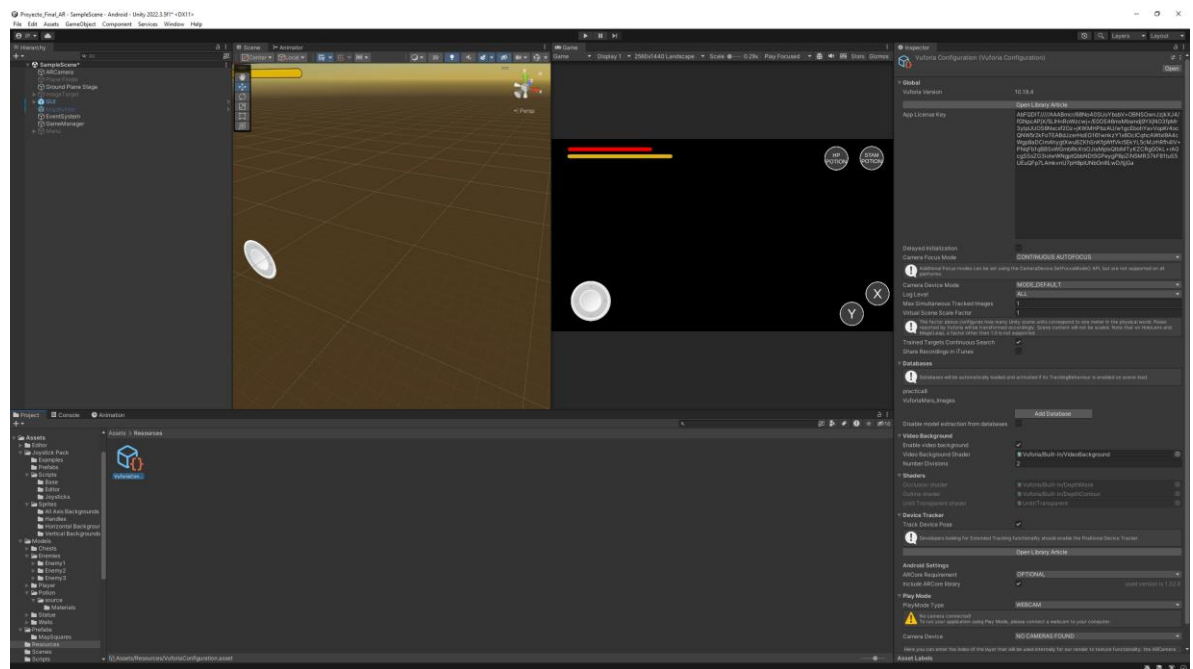


Figura 5. Configuración de Vuforia en Unity

## Enemigos

- Los enemigos cuentan con una variable de vida y dependiendo del arma entre dos y tres triggers.
- Para activarse, el jugador deberá entrar a un trigger cuadrado que abarca toda una casilla de suelo, una vez que se activa el enemigo seguirá la posición del jugador y cuando se encuentre a cierta distancia realizará un ataque.
- Los enemigos tienen un tiempo de espera entre cada ataque y este irá variando dependiendo del enemigo.

- Los triggers de las armas del enemigo quedan desactivadas hasta que se ejecuta la animación de atacar.
- Los enemigos cuentan con animación de ataque, muerte, idle y correr.

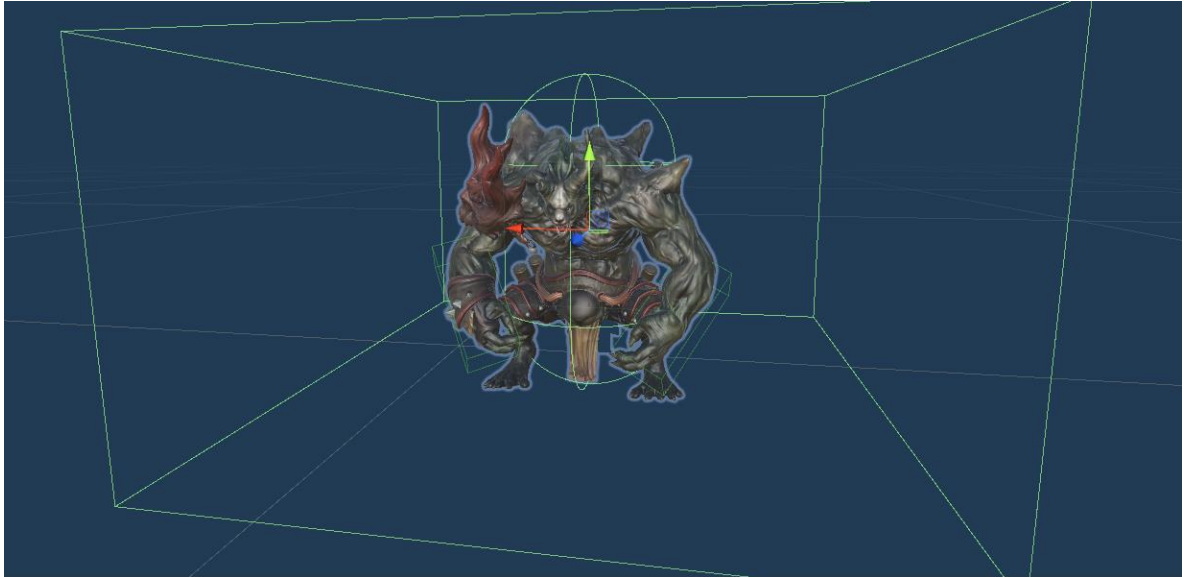


Figura 6. Triggers y collider de enemigo 3.

## Jugador

- El jugador tiene una variable para la vida y una variable para la stamina
- Cuenta con las acciones de atacar, recoger objetos, defenderse, correr e idle
- Además, cuenta con un diccionario que actúa como el inventario del jugador
- En el script del inventario se controlan las funciones de los objetos, en este caso están las funciones de las pociones para recuperar vida y stamina.
- Utiliza varias funciones para: detectar muerte del jugador, recibir daño, cambiar los estados de atacar, defender y recoger objetos
- El jugador contará con una pointlight que ayudará al jugador a avanzar por el escenario oscuro.
- El jugador tiene un collider que le impedirá pasar a través de las paredes y enemigos, su espada contará con un trigger que se activará cuando realice la acción de atacar.
- Varias animaciones descargadas de mixamo fueron duplicadas para poder llamar funciones cuando empiece o termine la animación. Además la animación de caminar y atacar fueron modificadas para que el personaje se mantuviera estático mientras se ejecutan estas animaciones.
- El GameObject que controla el ataque del jugador es el GameManager y este a su vez está conectado con el botón X de la GUI



Figura 7. Triggers y collider del jugador.

## Objetos y Cofres

- Mediante un trigger y script el cofre soltará un objeto aleatorio una vez que el jugador ingresa al trigger
- Los objetos tienen de igual forma un trigger y si el jugador se pone sobre este en vez de atacar se realizará acción de recoger.
- Con un booleano se detectará si el jugador recoge el objeto, una vez que se activa aumenta el inventario del jugador y después se destruye del mapa
- Las pociones cuentan con una pointlight para que sea más fácil su identificación, además el color de esta luz irá variando dependiendo del tipo de poción.
- La poción solo cuenta con un trigger ya que queremos que el jugador pueda posicionarse encima del ítem.

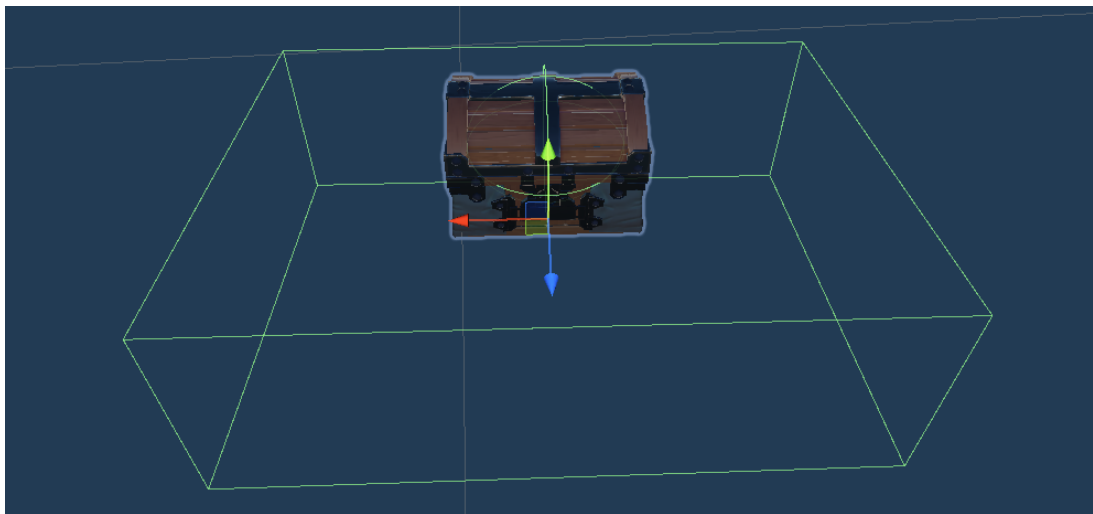


Figura 8. Trigger y collider del cofre.

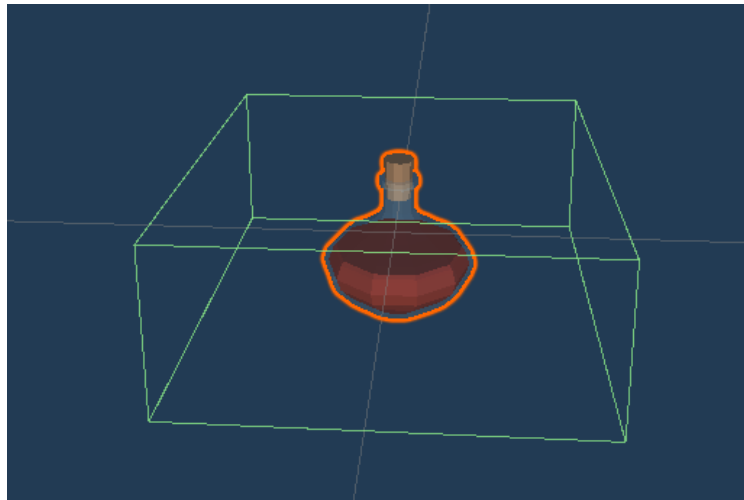


Figura 9. Trigger de la poción.

## Cronograma

### Original

	Oct 5 - 15	Oct 15 - 25	Oct 25 - Nov 5	Nov 5 - 15	Nov 15 - 25
Planeación					
Elementos graficos					
modelo personaje					
modelo enemigos					
modelos escenografía					
modelos de consumibles					
texturas					
efectos de sonido					
Programación					
investigación AR					
funciones de jugador					
funciones de enemigos					
generación de niveles					
Implementación					
Programación					
código jugador					
código enemigo 1					
código enemigo 2					
código enemigo 3					
código genrador de mapa					
importación de modelos					
Animaciones					
animaciones jugador					
animaciones enemigo 1					
animaciones enemigo 2					
animaciones enemigo 3					
animacion objetos					
Implementación AR					
Testing					

## Actualizado

1		Oct 5 -15	Oct 15 - 25	Oct 25 - Nov 5	Nov 5 - 15	Nov 15 - 29
2	Planeación					
3	Elementos graficos					
4	modelo personaje					
5	modelo enemigos					
6	modelos escenografia					
7	modelos de consumibles					
8	Programación					
9	investigación AR					
10	funciones de jugador					
11	funciones de enemigos					
12	generación de niveles					
13	Implementación					
14	Programación					
15	código AR					
16	código enemigos					
17	códigos de jugador					
18	código de armas					
19	código de GUI					
20	código de objetos					
21	código de generación de mapa					
22	código de cofres					
23	código Game Manager					
24	importación de modelos					
25	Animaciones					
26	animaciones jugador					
27	animaciones enemigo 1					
28	animaciones enemigo 2					
29	animaciones enemigo 3					
30	animacion objetos					
31	Implementación AR					
32	Testing					

## Costos

Costo (2 meses de desarrollo):

*Software Utilizado:*

Unity Pro: \$185 USD mensual

Photoshop: \$898 MXN mensual

3ds Max: \$5188 MXN mensual

Total al 29/11/2023: \$18,600 MXN

*Hardware Utilizado:*

Nvidia RTX 4070: \$14,544 MXN

32 GB de RAM DDR5: \$2,143 MXN

Intel Core i7 12<sup>th</sup> Gen: \$6,805 MXN

Tiempo de vida del Hardware: aproximadamente 3 años.

Costo de renta mensual: \$652 MXN

Costo para el proyecto: \$1305 MXN

*Sueldos*

Costo de implementación: \$350/hora

Costo para el proyecto: \$61,600 (4 horas diarias y 44 días hábiles)



#### *Otros Gastos:*

##### **Luz:**

Computadora: 700W/hora

Costo de Kilowatt por hora (considerando consumo intermedio de energía según la CFE): \$1.153 MXN

Costo de luz por hora: \$0.80 MXN

Estimado de luz mensual: \$140.8 MXN

Costo para el proyecto: \$281.6 MXN

**Internet 300 MEGAS:** \$859 al mes

Costo para el proyecto: \$1718 MXN

Costo total al 29/11/2023: \$83,223 MXN

El costo bruto de la aplicación es de \$83,223 MXN, considerando un 40% de ganancias el precio del proyecto a la fecha actual es de: \$116,523.

Este costo irá incrementando dependiendo del tiempo que se siga trabajando en la aplicación además de variar debido al precio del dólar.

## Referencias

<https://www.totalplay.com.mx/paquetes>

<https://store.unity.com/configure-plan/unity-pro?slug=unity-pro-1-year-prepaid-2020>

<https://www.adobe.com/mx/products/photoshop/compare-plans.html#:~:text=Puedes%20comprarlo%20como%20aplicaci%C3%B3n%20independiente,anual%20a%20MXN%20%244%2C788.00%2Fa%C3%B1o.>

<https://www.cfe.mx/hogar/tarifas/Pages/Acuerdosdetarifasant.aspx#:~:text=Consumo%20b%C3%A1sico%3A%20%240.595%20por%20cada,hora%20adicional%20a%20los%20anteriores.>