

Primer Parcial

Hacer un clon del videojuego Bomberman original de NES:

<https://www.youtube.com/watch?v=CZ9Pu9Usk5o>

Fecha de entrega: Lunes 20 de Mayo.

Debe entregarse el proyecto y código fuente:

- Link a repositorio git en aula virtual.

La build:

- Mediante Git releases en github.com

Version de Unity: 2018.3.x

El proyecto no debe contener errores de Unity al compilar, correr en editor o buildear.

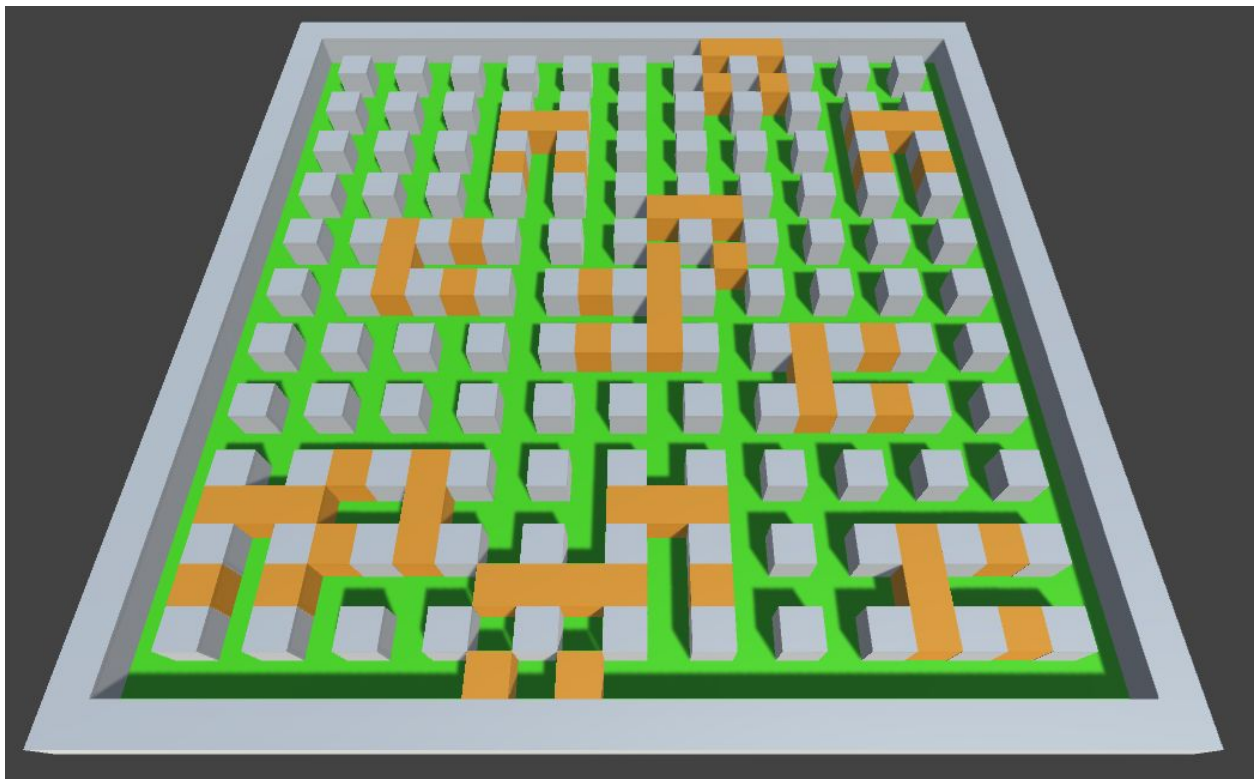
Juego Básico

El juego debe tener los siguientes puntos perfectamente funcionales para estar aprobado:

Generación del nivel

El nivel tiene una estructura previa generada desde el editor de Unity. Esto es: el piso, los bordes y las columnas principales (lo gris y verde en la imagen).

Deben generarse columnas destruibles (en marrón) randomicamente en el nivel. Estas deben situarse entre las columnas no destruibles.



Player

El jugador debe moverse con las flechas del teclado en el eje X y Z.

El jugador puede poner bombas con la barra espaciadora o click izquierdo del mouse (solo puede haber una bomba a la vez).

El player tiene 2 vidas. Al ser tocado por un enemigo o la explosión de una bomba este muere y aparece al inicio del nivel.

Prestar especial atención al “snapeo” del player al doblar en las esquinas teniendo en cuenta el concepto de cuadrilla como se puede ver en el video provisto (en criollo: que el player no baile entre las paredes).

Cámara

La cámara debe seguir al player a una distancia y rotación configurable desde el inspector en Unity.



Bombas

El player pone la bomba en la posición en la que está, la bomba se centra en la cuadrilla. Una vez que el player se alejó de la bomba este ya no puede atravesarla.

Las bombas tienen X tiempo de vida (modificable desde el inspector) hasta que explotan.

Explotan en cruz como se puede ver en el juego original.

La explosión de la bomba destruye las columnas destruibles, mata al jugador o a cualquier enemigo.

Enemigos (usando FSM)

El juego empieza con una cantidad de enemigos fijada desde el editor previo a correr el juego.

Estos enemigos (rojos) se moverán randomicamente.

No atraviesan columnas.

Tocan al player y lo matan.

Al chocar con otro enemigo o bomba se dirigen en dirección opuesta a donde se dirigían.

Condición de victoria y fin de juego

El juego termina cuando muere el player y no tiene más vidas o cuando el player sale por la puerta de la salida del nivel.

La puerta de salida del nivel se encuentra escondida en una columna destruible. Al hacerla explotar aparece la puerta.

La puerta permanece cerrada mientras haya enemigos vivos en el nivel.

Pantallas del juego

- Pantalla Inicial: Logo del juego, logo de la empresa (Cachuflito Games Inc.) y botón para empezar el juego.
- Pantalla de Loading: Loading screen con su porcentaje del siguiente nivel cargado.
- Pantalla de Juego: el juego propiamente dicho.
- Pantalla Final: esta pantalla mostrará datos de puntaje e información pertinente dependiendo de cada feature implementado.

Esta pantalla mostrará el puntaje más alto logrado entre partidas (mini Highscores).

Código y Proyecto de Unity

Se espera el uso de prácticas comunes vistas en clase sobre, entre ellas:

Estructura y claridad de código (legibilidad, nombre de scripts, variables y métodos)

Coding Standards de C#

Estructura de carpetas y archivos en el proyecto.

Uso de delegates y clases no MonoBehaviour donde sea pertinente.

Assets y scripts no utilizados no deben estar en el proyecto.

Se deberán tener en cuenta los warnings de Unity (salvo específicas excepciones no debería haberlos).

Puntos adicionales

UI

El juego tiene una UI completa. Cada número con su ícono.

La UI debe mostrar, durante y al finalizar la partida:

- Cantidad de vidas
- Puntaje
- Enemigos destruidos
- Tiempo de juego
- Cantidad de bombas simultáneas.
- Distancia de la explosión de las bombas.
- Cualquier otro valor que crean pertinente mostrar.

FPS

Apretando los botones 1 y 2 del teclado la cámara del juego pasa del juego original a cámara de un FPS, respectivamente. En modo FPS el movimiento y controles del jugador son los esperados en un juego FPS. La mecánica básica del juego se mantiene. El jugador no puede saltar.

Power Ups

Las columnas destruibles tienen chance de dejar un ítem de power up en el piso.

- **Explosión más grande:** este ítem incrementa en 1 la distancia de la explosión de la bomba.
- **Bombas simultáneas:** este ítem incrementa en 1 la cantidad de bombas que se pueden tener en todo momento (la explosión de una bomba hará explotar otra bomba si la toca)
- **Vida:** este ítem nos incrementa en 1 la cantidad de vidas.

Mas enemigos

Además del enemigo rojo están los enemigos violeta y amarillo:

- **Violeta:** Este enemigo puede atravesar columnas destruibles y bombas.
- **Amarillo:** Este enemigo tendrá alguna funcionalidad **especial** decidido por el alumno (se debe especificar la funcionalidad esperada mediante un comentario en el código).

Sistemas de Partículas

Se espera el uso de partículas donde sea pertinente.

Pulido

Pulido de las features y su coherencia (tanto funcional como artística) en el juego.

Cualquier cosa que no haya sido especificada en este documento que sume a la experiencia del juego.

Notas importantes

La decisión de arte del juego es libre a interpretación. Los colores estipulados son una sugerencia.

Se recomienda hacer algo simple pero pulido (minimalista) debido a la limitación de tiempo.

Los puntos evaluados serán tomados como un todo, esto es: la parte de código en conjunto con su funcionalidad en el juego y manejo de los conceptos de Unity.

Si se utilizan assets de terceros se debe mencionarlos en una pantalla de créditos.

Se tendrá muy en cuenta a la hora de evaluar el parcial la comprensión y seguimiento del enunciado.

El parcial es **individual**. Código similar al de sus compañeros influirá en la nota final.