

# TP3: Juego Terminado

*Taller de Programación I - Cátedra Ing. Diego Azcurra - 2do cuatrimestre 2014*

## Objetivo

El objetivo de este trabajo práctico es extender el programa desarrollado en el TP2 para completar la funcionalidad de un juego multiplayer.

## Desarrollo

Podrán participar de 1 a 4 jugadores simultáneamente en red. Para unirse a un juego, los jugadores se conectan a un servidor central que contiene el nivel en el que desean jugar.

Cada nivel tiene una determinada cantidad necesaria de jugadores que participan en él. Cada jugador que entra al escenario debe esperar a que se encuentren presentes la totalidad de los participantes necesarios en el nivel para comenzar el juego. Hasta entonces se le muestra una pantalla que indica que se encuentra en período de espera.

El juego es colaborativo entre los participantes; todos deben atacar y tratar de eliminar a todos los enemigos del nivel. Cuando eso ocurre se pasa al siguiente nivel. El juego es en tiempo real. Cada jugador podrá mover a su personaje por todo el escenario atacando y eliminando los enemigos de la pantalla.

Cada jugador tiene una cantidad determinada de vidas. Cuando es alcanzado por un enemigo pierde una vida y vuelve a la posición inicial en la que apareció en ese nivel. Existe un período de invulnerabilidad en el cual el jugador se muestra parpadeando y es inmune a los enemigos. Cuando el jugador pierde todas sus vidas ya no vuelve a aparecer en la pantalla y el nivel debe seguir siendo jugado por los otros participantes. Si todos los jugadores pierden todas sus vidas el nivel termina con "Game Over".

Después de terminar un juego (porque se pasaron todos los niveles o se perdió) y si hay quorum de jugadores (los que cada nivel requiera) el nivel reinicia.

## Plataformas atravesables

Algunas plataformas se podrán configurar como atravesables. Estas plataformas pueden ser atravesadas si se avanza hacia ellas en un sentido pero no pueden ser atravesadas en el otro. Esto trata de emular las plataformas de los juegos de este género donde el jugador puede saltar a través de ellas pero al descender cae sobre la misma sin atravesarla.

## Sonido

El juego debe tener música y efectos de sonido. Los mismos quedan a criterio de los alumnos, pero se espera que cada vez que ocurra un evento en el juego (explosión, golpes, etc.) el mismo sea indicado por un sonido.

## Enemigos y forma de ataque

Existirán enemigos autónomos. Por autónomo se entiende uno que se mueve solo y ataca a los jugadores. Este ataque puede ser algo planeado mediante una sofisticada inteligencia artificial o meramente producto del movimiento aleatorio del enemigo.

Para eliminar a los enemigos se trata de emular la dinámica del juego Snow Bros. Los personajes pueden arrojar nieve a los enemigos. La nieve primero inmoviliza al enemigo y después lo cubre completamente formando una bola. El personaje puede empujar esta bola. Si mientras la está empujando la patea, la bola sale despedida a gran velocidad arrastrando otros enemigos, bolas de nieve y jugadores que se crucen en su camino. Después de un momento la bola se destruye.

Los jugadores que son atrapados por una bola de nieve son arrastrados hasta el lugar donde la misma se destruye. Los enemigos y otras bolas de nieve son destruidos por el impacto inicial.

### Controles

Ataque: Barra espaciadora

Patear bola de nieve: Barra espaciadora

## Interfaz gráfica

La pantalla de juego debe contar con una interfaz gráfica que debe contar con los siguientes elementos como mínimo:

- El nombre del jugador propio y de los compañeros.
- El puntaje de cada uno de ellos
- La cantidad de vidas restantes
- El nivel

## Enemigos

**Estándar (obligatorio):** Se mueven aleatoriamente y pueden saltar de una plataforma a la otra. También se dejan caer desde una plataforma a otra inferior

**Tira fuego (obligatorio):** Se mueve aleatoriamente pero puede arrojar fuego. El fuego se mueve en línea recta.

**Remolino (2 punto):** Eventualmente se transforma en un remolino que lo hace inmune a los ataques. Cuando es un remolino puede flotar y desplazarse en línea recta atravesando plataformas.

**Calabaza macabra (1 punto).** Aparecen varias si se tarda mucho en terminar el nivel.

**Luchador de sumo (2 puntos):** Se mueve poco pero puede arrojar bolas de fuego hacia los niveles inferiores. El fuego puede traspasar todas las plataformas pero si toca a los personajes los mismos son eliminados.

**Bola de fuego (2 puntos):** Puede arrojar una bola de fuego al escenario (la arroja como cualquier objeto real, describiendo una trayectoria parabólica). La misma al

tocar el suelo genera un pequeño incendio en la zona del impacto que puede dañar a los personajes que lo toquen. Este incendio se apaga solo despues de un momento.

**Murciélago (3 puntos).** Puede volar por el escenario. Puede traspasar plataformas y posarse sobre las mismas.

**Boss (4 puntos).** Es el enemigo grande e intimidante que aparece en el último nivel y debe ser eliminado para seguir adelante (o ganar el juego). Debe tener un comportamiento específico y debe ser difícil de eliminar. También debe generar (de alguna u otra forma) otros enemigos. Se puede atacar mediante los disparos comunes de los jugadores o arrojándole enemigos transformados en bolas de nieve.

## Bonus

Los bonus aparecen ocasionalmente al eliminar enemigos. Los mismos se representan con un ícono (una botellita por ejemplo) que el personaje debe tocar para obtener el bonus. **Es obligatorio implementar como mínimo 2.**

- Vida extra
- Correr más rapido
- Tirar nieve más lejos
- Nieve más potente (con menos disparos se transforma en bola a un enemigo) (1 punto)

## Otras características opcionales

**Plataforma de hielo (1 punto).** Son plataformas con fricción mínima.

**Bombas (3 puntos).** Al obtener este bonus el personaje obtiene un cierto número de bombas que puede arrojar (con otra tecla). Arrojar la bomba implica dejarla en el lugar donde se encuentra en ese momento. Después de un momento la bomba explota y daña los enemigos y personajes que tiene cerca. Además la onda expansiva de la bomba arroja los elementos móviles que se encuentren cerca.

**Agujeros en el piso (2 puntos).** Son agujeros en el nivel inferior del escenario que tienen el extraño efecto de hacer que las cosas que caen por ellos reaparecen por la parte superior del nivel.

**Agua (3 puntos).** Es posible que en el nivel haya “lagunas”. Cuando un jugador o enemigo se mete en la laguna sus movimientos son más lentos y la gravedad es menor. El jugador puede saltar y eso provoca que se eleve en el agua (de esta manera si esta saltando dentro del agua y vuelve a saltar se eleva más, cosa que no puede ocurrir cuando esta en el aire). Además el color del personaje se modifica cuando se sumerge en el agua.

**Portales (4 puntos).** Al obtener este bonus el jugador tiene dos disparos del [arma de portales](#). Con dicha arma puede disparar una bola de antimateria contra una pared (vertical) lo que tiene como consecuencia, naturalmente, la generación de un portal interdimensional. El segundo disparo, contra otra pared, genera un portal que es la salida del primero. Esto significa que todos los personajes, enemigos y bolas de nieve que entran por uno reaparecen por el otro.

**Pre-entrega (3 puntos sólo en caso de aprobarla):** Existirá una pre-entrega definida en las fechas al final de este enunciado. El objetivo de la preentrega es

tener completamente implementado (no necesariamente en red) **el enemigo estándar y el ataque** (todo lo descrito en la sección “Enemigos y forma de ataque”). La preentrega no tiene segunda oportunidad y su desaprobación no tiene un efecto perjudicial en la nota final.

## Condiciones mínimas de aprobación

Es necesario implementar los 2 enemigos obligatorios, 2 bonus y obtener un mínimo de 5 puntos en items opcionales.

## Bonificaciones

Superados los 5 puntos mínimos de aprobación se pueden obtener bonificaciones en la nota final si se alcanzan:

Un total de **10 puntos** en items opcionales otorga **1 punto** adicional a la nota final.

Un total de **15 puntos** en items opcionales otorga **2 puntos** adicional a la nota final.

## Restricciones

- Para implementar las comunicaciones se deberá utilizar la biblioteca de sockets provista por el sistema operativo.
- Para implementar multithreading se puede utilizar la biblioteca provista por el sistema operativo, C++ 11, o bien la abstracción provista por SDL 2.0.
- Para la representación gráfica se deberá utilizar la biblioteca SDL 2.0 (<http://www.libsdl.org>).
- Para la simulación física debe utilizarse la biblioteca Box2D (<http://box2d.org>).
- Para la lectura y escritura de archivos JSON debe utilizarse, preferiblemente, la biblioteca jsoncpp (<http://jsoncpp.sourceforge.net>). La utilización de otras bibliotecas no está prohibida pero debe ser consultada. **No se permite la utilización de un parser propio.**
- Respetar las condiciones de entrega especificadas en el sitio web de la materia.
- Todo el código debe ser desarrollado íntegramente por cada grupo. No se permite la reutilización de código de cuatrimestres anteriores o de otras materias. Ante cualquier duda se deberá consultar con los docentes. La reutilización de código sin consulta previa será condición suficiente para la desaprobación de la materia.

**Este enunciado no es definitivo. Si se realizan cambios en clase se respetarán y evaluarán los mismos.**

## Fechas

Semana #	Fecha	Tema
11	29 de octubre	Entrega TP2. Presentación enunciado TP 3
12	5 de noviembre	Primer recuperatorio TP2. Consultas TP 3
13	12 de noviembre	<b>Pre-entrega opcional del TP 3.</b> Segundo recuperatorio TP2. Consultas TP 3
14	19 de noviembre	Consultas
<b>15</b>	<b>26 de noviembre</b>	<b>Entrega TP 3</b>
16	3 de diciembre	Primer recuperatorio TP 3
17	10 de diciembre	Segundo recuperatorio TP 3