

Universidad Tecnológica Nacional - Facultad Regional Mendoza -
Tecnicatura Superior en Programación

Metodología de la Investigación:

Proyecto de desarrollo de software

Equipo de trabajo:

Fedeli, Andres

Ferraro, Giuliana

Lattuca, Andres Nicolas

Perez, Tomás

Rodriguez Dei Rossi, Agustin

ÍNDICE

Marco teórico	3
Antecedentes	3
Scrum backlogs	4
PRODUCT BACKLOG	5
SPRINT BACKLOG	5
Actas de reuniones	6
Xtreme Tower Defense	9
REGLAS DEL JUEGO	9
ESPECIFICACIONES	10
LIMITACIONES	10
Bibliografía	11
Anexo	12

Marco teórico

Para el proyecto de Metodología de la Investigación, se optó por el desarrollo de un software de videojuego del género *tower defense*. Se seguirá el marco de desarrollo ágil [SCRUM](#), adaptado al contexto de un proyecto universitario. Se trabajará de modo colaborativo presencial y remotamente, con ayuda del servicio de versionado GitHub.

El [tower defense](#) es un subgénero de los juegos de estrategia. El objetivo de los juegos tower defense consiste en evitar que los enemigos lleguen a la base/torre que se desea proteger. Si esto sucediera, el jugador pierde. Para lograr defender la base/torre se colocan unidades de defensa a lo largo del recorrido de los enemigos. Estas pueden disparar proyectiles, lasers, etc. A lo largo del juego las unidades enemigas comienzan a volverse más fuertes por lo que es necesario mejorar el armamento de las unidades de defensa. Mientras más enemigos se destruyen, más recursos se obtienen. Estos recursos son utilizados para mejorar el armamento, obtener mejores unidades, etc.

Se eligió desarrollar en este subgénero debido a la versatilidad del mismo, pues da la posibilidad de crear un juego sencillo pero que mantenga el dinamismo de los videojuegos de estrategia. Además, los [antecedentes del género](#) indican que es apropiado como proyecto para equipos reducidos y desarrolladores independientes. Incorporando los elementos básicos que definen al tower defense, se espera desarrollar un software simple, sin gran detalle gráfico, pero que encuadre en el género.

Antecedentes

Los primeros juegos del género *tower defense* datan de la década del 1980. Se considera que los precursores fueron los juegos de arcade Space Invaders (1978) y Missile Command (1980). El clásico de estrategia Rampart, del año 1990, es uno de los primeros juegos de arcade que se ajusta al estilo de los tower defense modernos. **Rampart**, que fue diseñado por John Salwitz y lanzado por Atari Games, combina los géneros *shooter* y *puzzle*, al dar al jugador la oportunidad de construir un castillo y muros de defensa con piezas al estilo del Tetris.

No obstante el éxito de este lanzamiento, no surgieron muchos otros juegos del género hasta la adopción masiva del mouse. Desde entonces la popularidad de los tower defense incrementó considerablemente, sobre todo impulsada por desarrolladores independientes.

Entre los años 2007 y 2008, con la implementación de Adobe Flash, los juegos tower defense se convirtieron en un fenómeno de los juegos RTS (de estrategia en tiempo real) para PC. Dos ejemplos dignos de mención de esta época son Flash Element Tower Defense y Desktop Tower Defense. **Flash Element TD** comenzó como un proyecto personal de David Scott, quien se inspiró WarCraft III Element TD y decidió practicar sus habilidades de programación creando su propio tower defense simplificado para Flash.

Asimismo, **Desktop Tower Defense** fue desarrollado por Paul Preece, un conocido de David Scott, quien quiso crear un tower defense para navegador que permitiera erigir torres en forma de laberinto para bloquear el avance del enemigo. Ambos proyectos son emblemáticos del desarrollo independiente de videojuegos de baja complejidad gráfica.



[Rampart](#): ejemplo de un tower defense con gráficas 2D sencillas y vista *top-down*.

El éxito de estos títulos produjo que el género siguiera evolucionando, tanto en PC como en consolas, y adaptándose a las nuevas plataformas y tecnologías móviles y táctiles. Actualmente los tower defense son muy variados en su complejidad y en sus características gráficas, temáticas y formales (modalidad de juego *single-player/multi-player*, en tiempo real/por turnos, cámara, punto de vista), etc.

Scrum backlogs

Uno de los requisitos de proyecto es que debe desarrollarse para Linux. Por lo tanto se ha elegido un entorno multiplataforma para el lenguaje C++. Se desarrollará en el IDE AppGameKit, que facilita el manejo de recursos gráficos y cuenta con un pseudocódigo claramente documentado para dicho lenguaje.

Se ha especificado que la parte gráfica del juego no es una prioridad y que, por el contrario, lo más importante es la funcionalidad lógica y la jugabilidad del proyecto. Estas especificaciones son la base de los backlogs que se presentan a continuación.

Referencias:

AR: Agustin Rodriguez Dei Rossi

NL: Andres Nicolas Lattuca

TP: Tomás Perez

AF: Andres Fedeli

GF: Giuliana Ferraro

Metodología de la Investigación
Proyecto de desarrollo de software

PRODUCT BACKLOG			
ID	REQUISITO	PRIORIDAD	ESTIMACIÓN
1	Que el usuario pueda jugar una partida de principio a fin	Muy alta	63
2	Que se pueda usar el software para el juego sin mayores inconvenientes	Alta	14
3	Que haya una evolución que permita el avance del juego	Alta	16
4	Que el usuario pueda interactuar fácilmente con el software	Media	24
5	Que la experiencia de juego sea coherente	Media	6
6	Que se pueda consultar material de referencia	Media	5

SPRINT BACKLOG Inicio del proyecto: 05/04/2017 Fin del proyecto: 21/06/2017				
ID / REQUISITO	TAREA	RESPONSABLE	ESTADO ¹	ESTIMACIÓN
5 Experiencia de juego coherente	Conceptualización	GF, AR, TP, AF, NL	Completa	6
4 Interacción del usuario con el software	Diseño del sistema: menú e interfaz de usuario	AR, AF	Completa	12
	Diseño del sistema: activos gráficos	GF, NL, AF	Completa	12
1 Posibilidad de jugar	Codificación: respuesta al puntero	GF	Completa	2
	Codificación: configuración del escenario	GF	Completa	7
	Codificación: dinámica y movimiento	TP, AF	Completa	8
	Codificación: personaje y	TP, AF	Completa	16

¹ Detalles e información actualizada sobre el estado de cada tarea en la hoja de cálculos "Backlog Alfa".

Metodología de la Investigación
Proyecto de desarrollo de software

	defensa			
	Codificación: enemigos	TP, AF	Completa	10
	Codificación: reglas y aspectos lógicos	TP, AF, AR, GF, NL	Completa	20
3 Avance del juego	Codificación: puntaje	AR	Completa	8
	Codificación: tiempo de juego	NL	Completa	8
2 Jugabilidad adecuada	Pruebas	NL, AF, TP	Completa	12
	Puesta en marcha	AR	Completa	2
6 Acceso a material de referencia	Documentación	GF	Completa	5

Actas de reuniones

5 de abril de 2017

Integrantes: AF, GF, NL, TP, AR

- Constitución del grupo y planteo de posibles temas. Definición de roles del modelo SCRUM:

- Scrum Master: Agustin Rodriguez Dei Rossi
- Product Owner: Giuliana Ferraro
- Scrum Team: Andres Fedeli; Andres Nicolas Lattuca, Tomas Perez

14 de abril de 2017

Integrantes: AF, GF, NL, TP, AR

- Preparación del product backlog: objetivos y prioridades del cliente.
 - Definición del género del juego (defensa de torre) y del entorno de desarrollo. IDE AppGameKit compatible con Windows y Linux, con pseudocódigo para C++. Nombre provisorio: "Xtreme Tower Defense".

19 de abril de 2017

Integrantes: AF, GF, NL, TP, AR

- Investigación de antecedentes.
 - Avance en la lectura de la documentación del IDE y del lenguaje.
 - Búsqueda de posibles recursos gráficos para la interfaz del juego.

26 de abril de 2017

Integrantes: AF, GF, NL, TP, AR

Metodología de la Investigación
Proyecto de desarrollo de software

- Planificación del sprint: objetivo del sprint y tareas necesarias para conseguirlo.
- Planteo de un sprint backlog provisorio (ver anexo).

03 de mayo de 2017

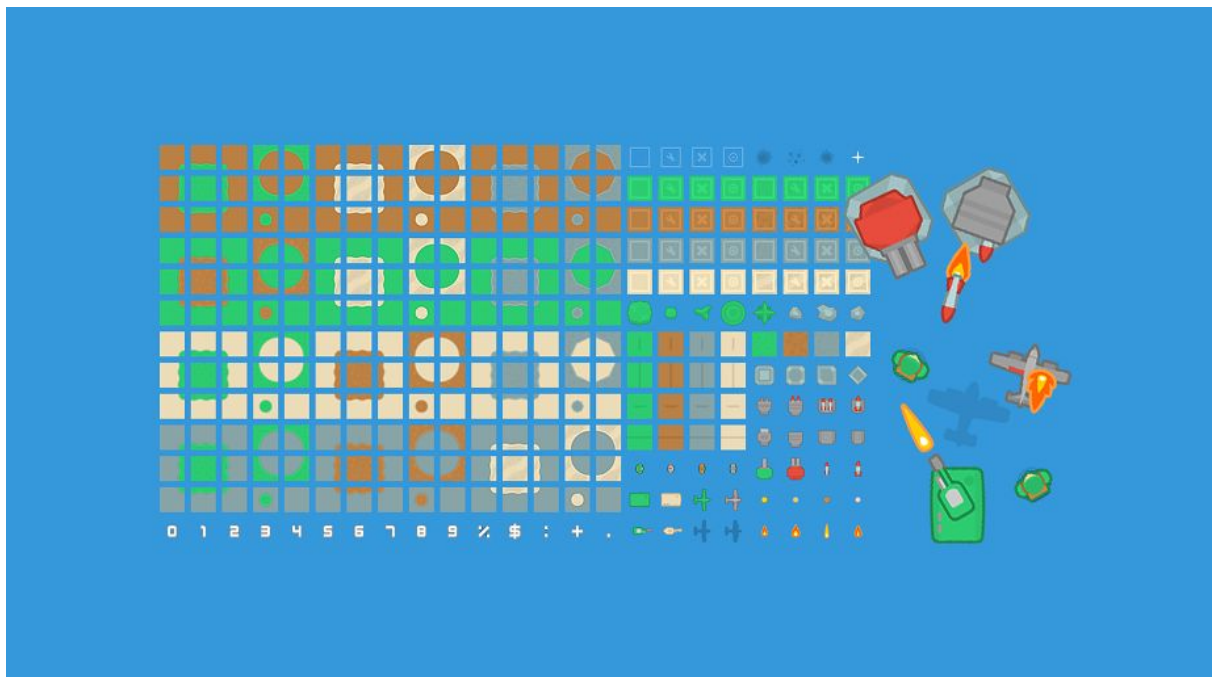
Integrantes: GF, NL, AR

- Creación de repositorios en GitHub: <https://github.com/Slayerr1991/mdiTowers>.
Introducción al uso de la plataforma.

12 de mayo de 2017

Integrantes: AF, GF, NL, TP, AR

- Últimos detalles a la conceptualización del juego.
- Definición de los activos gráficos. Paquete de assets 2d “Tower Defense (top-down) Pack” diseñado por Kenney Vleugels, de dominio público y uso libre bajo licencia Creative Commons Zero, CC0.
- Codificación del menú principal y la pantalla de inicio.



Vista previa del paquete *top-town* para tower defense de [Kenney Vleugels](#).

18 de mayo de 2017

Integrantes: GF, NL, AR

- Finalización del menú de inicio
- Diseño de imágenes para el escenario de fondo, la base y una torre
- Inclusión del temporizador

25 de mayo de 2017

Integrantes: AF, GF, NL, TP, AR

- Codificación del movimiento del enemigo

Metodología de la Investigación
Proyecto de desarrollo de software

- Codificación de botonera para colocar las torres
- Diseño de pantalla de finalización del juego
- Diseño y codificación preliminar del sistema de puntos y monedas

1 de junio de 2017

Integrantes: AF, GF, NL, TP, AR

- Codificación del movimiento del enemigo en un bucle
- Inclusión en el código de funciones para monedas, vidas, puntaje
- Implementación de pantalla de finalización del juego
- Codificación de opciones más amigables para colocar las torres
- Diseño de un segundo modelo de torre e imagen para la pantalla de posicionamiento de las torres

7 de junio de 2017

Integrantes: AF, GF, NL, TP, AR

- Codificación de limitaciones para las torres, para evitar que se superpongan o se coloquen fuera del área especificada
- Corrección del ángulo de ubicación de las torres para que apunten al camino
- Corrección a las coordenadas para el desplazamiento del enemigo
- Creación de balas para el primer tipo de torre
- Diseño e incorporación de imágenes de fondo mejoradas para la zona de colocación de torres

14 de junio de 2017

Integrantes: GF, NL, TP, AR

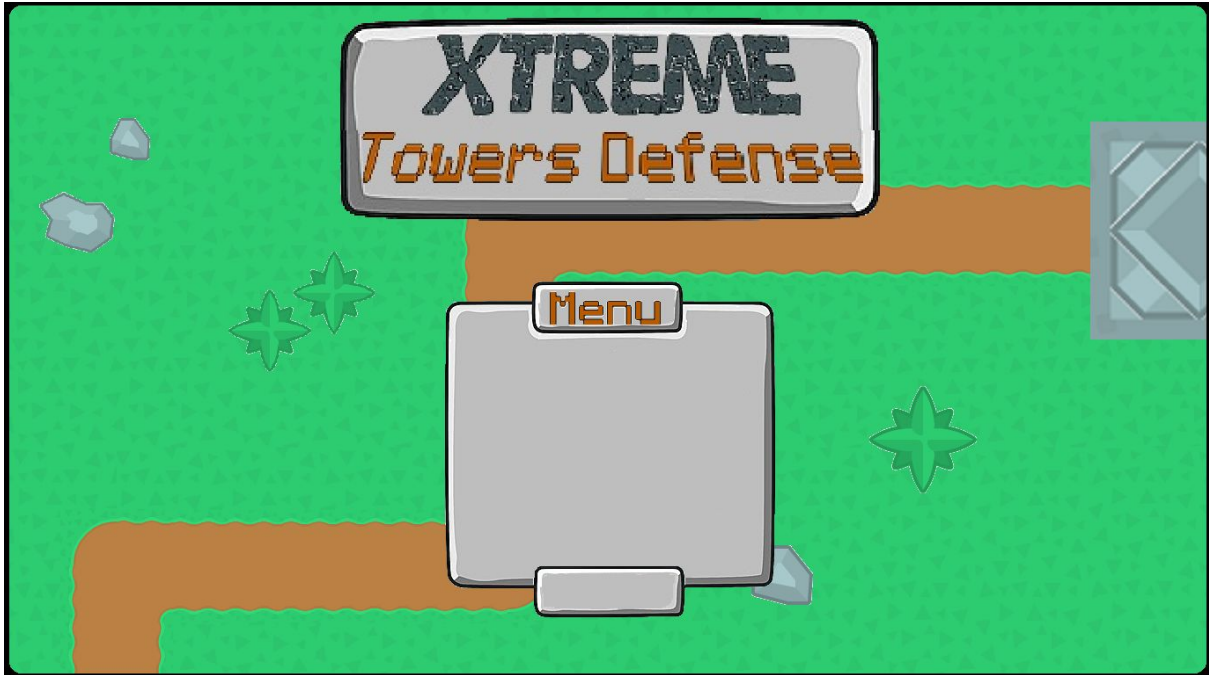
- Corrección de errores en la trayectoria del primer tipo de balas
- Codificación de un segundo tipo de balas
- Implementación del disparo automático
- Pruebas y corrección de errores en la posición de las torres

20 de junio de 2017

Integrantes: AF, GF, NL, TP, AR

- Codificación de la resistencia del enemigo y un contador de enemigos aniquilados
- Corrección en las balas para que todas las torres puedan disparar a la vez
- Diferenciación del comportamiento de los dos modelos de torres
- Finalización de la codificación de reglas para el puntaje y los casos de ganar/perder el juego
- Ajuste en los indicadores para una mejor experiencia de usuario
- Diseño e implementación de pantalla para juego ganado
- Pruebas y corrección de errores de desempeño

Xtreme Tower Defense



Vista preliminar del menú de inicio del juego.

REGLAS DEL JUEGO

El objetivo del juego es defender mediante torres contra el ataque de enemigos a la base situada al final del camino. Si algún enemigo llega a la base, se pierde el juego. Si se consigue aniquilar a 15 enemigos de creciente resistencia a las balas, se gana el juego.

A medida que se exterminan enemigos, se obtienen puntos que desbloquean más monedas para adquirir torres y defender la base. Se inicia con 0 puntos y 10 monedas para invertir en defensa. Luego de eliminado el primer enemigo, se consiguen 100 puntos y 10 monedas adicionales, y así sucesivamente.

Hay dos variedades de torres, de precio, eficacia e intensidad de disparo diferentes. Las torres verdes son más económicas pero disparan misiles menos confiables y de bajo impacto. Las torres rojas, más costosas, disparan con mayor eficiencia misiles blindados que generan gran daño a los enemigos.

Las torres disparan automáticamente a los enemigos que detectan en su rango, por lo que la zona de colocación de torres está restringida a los alrededores del camino transitado por los enemigos. Es posible apilar torres de distinto tipo si se lo considera necesario para combinar las virtudes de los dos tipos de munición.

Solo se podrán colocar 7 torres como máximo: 4 verdes y 3 rojas. Según se indicó, la posibilidad de acceder a estas 7 torres depende de las monedas y puntos obtenidos al eliminar enemigos. La selección y distribución de las torres en el mapa es el factor de estrategia más importante para lograr ganar el juego. Es necesario considerar la velocidad de desplazamiento del enemigo, la velocidad y eficacia de los misiles, el ángulo y rango de las torres, etc.

ESPECIFICACIONES

El resultado final del juego está disponible en GitHub para su ejecución en Linux. También puede consultarse el código fuente en el repositorio citado anteriormente. Si se desea acceder al compilador o entorno de desarrollo utilizado, estos podrán solicitarse a cualquiera de los miembros del equipo.

LIMITACIONES

Dada la escasez de recursos y el limitado tiempo disponibles para completar el proyecto, el desarrollo obtenido hasta el momento presenta limitaciones y áreas susceptibles de mejoras en el futuro. Se podría considerar que el resultado del presente proyecto es un prototipo o punto de partida que sin duda podría escalarse de varias formas como las que se describen a continuación.

Actualmente el jugador solo cuenta con una opción de mapa, que en niveles subsiguientes podría modificarse para dar lugar a mayores grados de dificultad. Podrían añadirse caminos, áreas de mayor alcance para las torres o escondites para menor vulnerabilidad de los enemigos.

El diseño del juego también permite agregar más modelos de torres, con distintas propiedades de ataque, así como incorporar nuevas especies de enemigos que sean más o menos susceptibles a estas torres. Por ejemplo, se podrían añadir enemigos que presentaran combinaciones únicas de resistencia a las balas y velocidad de traslado, ofreciendo dificultades cada vez mayores para erradicarlos.

Asimismo, podrían seguirse desbloqueando opciones de defensa a medida que se avanza en el juego, como la opción de paralizar a los enemigos por distintos lapsos de tiempo, de detonar una bomba para eliminar instantáneamente a todo enemigo restante o de disparar proyectiles con objetivo a distancia. Cabe destacar que la lógica planteada e implementada hasta ahora hace viables todas estas proyecciones a futuro.

Bibliografía

- Avery, P., Togelius, J., Alistar, E. y Pieter van Leeuwen, R. (2011). Computational Intelligence and Tower Defence Games. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, 1084-1091. Disponible en: <http://julian.togelius.com/Avery2011Computational.pdf>
- Desktop Tower Defense (2017). En *Wikipedia*. Recuperado de: https://en.wikipedia.org/wiki/Desktop_Tower_Defense
- Flash Element TD (2016). En *Wikipedia*. Recuperado de: https://en.wikipedia.org/wiki/Flash_Element_TD
- López, G., Menzinsky, A. y Palacio, J. (2016). *Scrum Manager: Guía de formación [e-book]*. Disponible en http://www.scrummanager.net/files/sm_proyecto.pdf
- Manrubia Pereira, A. M. (2014). El proceso productivo del videojuego: fases de producción. *Historia y Comunicación Social*, 19. N.º Esp. Marzo, 791-805. ISSN: 1137-0734. Disponible en: <https://revistas.ucm.es/index.php/HICS/article/viewFile/45178/42539>
- Peralta, A. (2013). Metodología SCRUM (Cátedra de Ingeniería de Software) [en línea]. Universidad ORT Uruguay. Disponible en: <http://fi.ort.edu.uy/innovaportal/file/2021/1/scrum.pdf>
- Rampart (s. f.). En *The International Arcade Museum*. Recuperado de: https://www.arcade-museum.com/game_detail.php?game_id=9263
- Ruiz, P. (2013). Making a Hit Tower Defense Game: A Top App Dev Interview. *Ray Wenderlich*. Disponible en: <https://www.raywenderlich.com/30636/top-developer-interviews-ironhide-game-studio>
- Schwaber, K. y Sutherland, J. (2016). *La Guía Definitiva de Scrum: Las Reglas del Juego [e-book]*. Disponible en: <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf>
- Simmonds, N. (2010). Tower Defense – A brief History. *My Gaming*. Disponible en: <https://mygaming.co.za/news/features/6341-tower-defense-a-brief-history.html>
- Vleugels, K (s. f.). Tower Defense (top-down). *Kenney*. Disponible en: <http://kenney.nl/assets/tower-defense-top-down>

Anexo

Backlog preliminar obtenido en el primera reunión de planificación del sprint

ID	TAREA	REQUISITO	PRIORIDAD	ESTADO	ESTIMACIÓN
1	Conceptualización	Experiencia de juego coherente	Media	En progreso	6
2	Diseño del sistema: menú e interfaz de usuario	Interacción del usuario con el software	Media	En progreso	12
3	Diseño del sistema: activos gráficos	Interacción del usuario con el software	Media	En progreso	12
4	Codificación: respuesta al puntero	Posibilidad de jugar	Muy alta	No iniciada	2
5	Codificación: configuración del escenario	Posibilidad de jugar	Muy alta	No iniciada	7
6	Codificación: dinámica y movimiento	Posibilidad de jugar	Muy alta	No iniciada	8
7	Codificación: personaje y defensa	Posibilidad de jugar	Muy alta	No iniciada	16
8	Codificación: enemigos	Posibilidad de jugar	Muy alta	No iniciada	10
9	Codificación: reglas y aspectos lógicos	Posibilidad de jugar	Muy alta	No iniciada	20
10	Codificación: puntaje	Avance del personaje en el juego	Alta	No iniciada	8
11	Codificación: tiempo de juego	Avance del personaje en el juego	Alta	No iniciada	8
12	Pruebas	Jugabilidad adecuada	Alta	No iniciada	12
13	Puesta en marcha	Jugabilidad adecuada	Alta	No iniciada	2
14	Documentación	Acceso a material de referencia	Media	En progreso	5