



Universidad Nacional del Sur  
Departamento de Ciencias e Ingeniería de la Computación  
**Tecnología de Programación**  
(2018)



## **Galaxian**

Proyecto

Las plataformas de juegos móviles son muy populares hoy en día y dado que la “nostalgia” por juegos clásicos está muy de moda, la empresa “Midway” ha decidido lanzar una nueva versión de su conocido



juego “Galaxian”. Este juego es del tipo “vertical shooter” donde el jugador controla una nave con un movimiento limitado solamente al eje horizontal en la base de la pantalla, disparando hacia la parte superior de la misma, y los enemigos son “extraterrestres” que, inicialmente, se encuentran en formación y progresivamente descienden para atacar. El objetivo del juego es acabar con todas las naves extraterrestres sin que las mismas destruyan la nave del jugador.

En esta versión modificada, algunos de los enemigos, al ser destruidos, entregarán “power ups” a modo de premios que modificarán distintos aspectos de la nave del jugador.

### **El mapa**

El mapa donde se juega es un escenario vertical estático donde en la parte superior se encuentran los enemigos en formación y en la parte inferior la nave del jugador.

Los enemigos atacan hacia abajo al jugador de dos formas, con armas o “kamikaze” arrojándose hacia el. Si un enemigo pasa la línea inferior de la pantalla, vuelve a aparecer por arriba o (opcionalmente) en su posición original.

### **El nivel**

El nivel está definido por el conjunto de enemigos que el mismo presenta variando en dificultad conforme avanza el juego.

La cantidad de enemigos en el nivel es fija para cada uno de ellos y todos los enemigos aparecen de manera conjunta al iniciar un nivel.

Alternativamente, algunos niveles podrán presentar además “defensas” que simplemente serán obstáculos a ser destruidos por el jugador o por los enemigos sirviendo de “guarida” para cualquiera de ellos.

### **Los personajes y objetos**

Todos los personajes y objetos (premios u obstáculos) del juego, tienen una determinada cantidad de puntos de vida. Al terminarse estos puntos, el objeto o personaje se destruye.

Por otra parte, los personajes (todos ellos) tienen además una fuerza de impacto “kamikaze” y, pueden tener o no, un arma con determinados puntos de “golpe” por disparo y un “alcance”. Estos parámetros determinan cuánta vida consume al atacar y alcance tienen.

Los enemigos tienen además una velocidad que determina qué tan rápido se mueve por el mapa, un puntaje.

Los objetos que aparecen en el juego tienen determinada cantidad de puntos de vida (tanto los obstáculos como los premios).

## Los enemigos y las armas

Los enemigos pueden ser tener arma o no. Aquellos que no tienen un arma atacan al jugador arrojándose sobre él. Aquellos que tienen arma pueden dispararle al jugador pero algunos de ellos la pierden al llegar al 20% de su vida, a partir de ahí atacan en modo “kamikaze”.

Este juego deberá tener al menos tres enemigos solo “kamikaze” con distintas inteligencias, una que busque al jugador y otro que se mueva aleatoriamente. El tercer enemigo se moverá buscando al jugador hasta que llegue al 50% de su vida, a partir de ahí entrará en modo “mareado” y se moverá aleatoriamente.

Por otra parte, al menos dos enemigos deberán tener armas, uno de ellos la debe poder “perder”.

Los enemigos pueden o no colisionar entre sí y puede o no haber “friendly fire”, usted lo decide.

## Los obstáculos

Los obstáculos serán de dos tipos, “destruibles por todos” o “barricadas para enemigos”.

Los del primer tipo pueden ser destruidos tanto por disparos del jugador, como por disparos enemigos o un “kamikaze”. Los del tipo “barricadas para enemigos” son más que nada una “molestia” para el jugador y solo pueden ser destruidos por sus disparos.

## Los premios

Al ser destruidos, algunos de los enemigos dejarán un premio que “cae” en dirección al jugador con movimientos horizontales aleatorios. El jugador deberá interceptar el premio antes de que el mismo cruce la línea de la pantalla porque ahí el premio desaparece.

Los premios caen a distintas velocidades.

Estos premios pueden ser de dos tipos:

- Magia temporal: Este tipo de premios otorga efectos temporales al jugador o los enemigos. Uno de los efectos deberá ser un “congela tiempo” que detenga a todos los enemigos y un “super misil” que otorgue un poder de golpe multiplicado al arma del jugador por un intervalo de tiempo.
- Objetos preciosos: Este tipo de premios le confiere al jugador poderes especiales para su nave o su arma. Uno de los efectos deberá ser una suerte de “campo de protección” que destruya al siguiente enemigo “kamikaze” que lo toque, una suerte “poción” que recupere la vida de la nave del jugador y otros dos deberán ser mejoras para el arma.

## Lineamientos generales de implementación y diseño

Como se ha mencionado anteriormente, usted deberá diseñar de manera completa el juego pudiendo elegir la temática que desee.

Los requerimientos establecidos para el diseño son:

- Debe respetar el tipo de juego.
- Debe respetar los tipos mínimos de enemigos que se describieron antes.
- Debe respetar los tipos mínimos de premios que se describieron antes.
- Debe respetar los tipos mínimos de obstáculos que se describieron antes.
- Deberán aparecer al azar los premios al destruir los enemigos.
- Deberán existir al menos dos niveles diferentes al menos uno teniendo obstáculos.

## Donde conseguir recursos gráficos

Se les recomienda a los alumnos, a la hora de buscar recursos gráficos, el recurrir a los llamados “sprites” de juegos viejos. Estos son “plantillas” con todos los personajes de un juego en todas las posiciones posibles simplificando así la búsqueda de imágenes.

En el siguiente sitio web hay algunos recursos de este tipo:

- [https://www.sprisers-resource.com/game\\_boy\\_advance/namcomuseum/sheet/22730/](https://www.sprisers-resource.com/game_boy_advance/namcomuseum/sheet/22730/)

Existen muchos sitios de este tipo dedicados a “sprites” y recursos para juegos. Se les recomienda explorar las posibilidades y ser creativos a la hora de elegir.

Por otra parte, dado que pueden utilizarse a la hora de implementar los llamados “gif” manteniendo la animación, es posible, partiendo de las partes de los “sprites” generar gif animados en sitios como el siguiente:

- <http://gifmaker.me/>

Para ello, inicialmente deberán cortar las distintas partes del “sprite” por ejemplo usando “gimp” o “photoshop” para mantener la transparencia.

# Método de trabajo

Este proyecto se realizará en grupos de 3 alumnos. Habrá una entrega final con opción de re-entrega.

Durante todo este proyecto se utilizará un repositorio "Github" por equipo y se realizarán las entregas pertinentes a través del mismo.

Cada 15 días corridos habrá un "sprint". Un sprint, en el desarrollo de software, es una reunión corta donde los involucrados en los distintos niveles de diseño y desarrollo se juntan. En este caso, en cada sprint, el ayudante responsable de la corrección del proyecto se reunirá con los alumnos para que le informen el estado y el avance del proyecto. Estas reuniones serán de 5 minutos donde el equipo muestra sus avances y habrá 5 minutos más para que el ayudante pregunte. El objetivo de estas reuniones es ayudar al grupo a planificar mejor el desarrollo del proyecto.

Como resultado de cada una de las reuniones, los integrantes de la comisión deberán redactar una minuta (correspondiente a la plantilla) que contenga lo expuesto por el grupo en la reunión y subirla al repositorio.

A continuación se detalla una lista tentativa de los objetivos a cada sprint:

## Primer Sprint

- Diseño del juego
- Diseño UML del juego contemplando todos los aspectos generales del sistema: jugador, enemigos, obstáculos y power-up.
- Diseñar en pseudocódigo (general) la generación del mapa.
- Diseñar el movimiento del jugador y enemigos (pseudocódigo o diagrama de interacción).
- Tener pensado como interactuar entre el jugador, los enemigos y el mapa. Diseñar un diagrama de interacción borrador.
- Tener funcionando el GitHub con los archivos generados.

## Segundo Sprint

- Implementar gráficamente el mapa (sin colisiones).
- Implementar el jugador de manera básica.
- Implementar gráficamente el jugador (sin colisiones).
- Implementar un enemigo básico.
- Implementar gráficamente el enemigo (sin colisiones ni movimiento).
- Diseñar el disparo.

## Tercer Sprint

- Implementar el disparo (sin interacción).
- Incorporar patrones de diseño al juego.
- Incorporar detección colisiones al juego (implementar).
- Diseñar la interacción entre jugador/enemigos y disparos.
- Diseñar la interacción entre el jugador y los power up.

## Cuarto Sprint

- Implementar el disparo con interacción y gráficamente.
- Implementar la interacción entre jugador/enemigos y disparos.
- Implementar la interacción entre el jugador y los power-up.

## Quinto Sprint

- Implementar todos los enemigos.
- Implementar todos los power-up.
- Implementar el ganar/perder.
- Implementar todo gráficamente.

## MINUTA DE FIN DE SPRINT

Comisión TDP	
Docente TDP	
Fecha	
Sprint que se evalúa	
Objetivos cumplidos (máximo de 500 caracteres)	
Objetivos pendientes (máximo de 500 caracteres)	
Correcciones a realizar (máximo de 500 caracteres)	