



## **Informe Tancovich**

### **Programación 3**

#### **Integrantes:**

**Jeremias Burgueño**

**Rodrigo Fanjul**

**Mauro Lagueruela**

**Ignacio Huerta**

## **Resumen de la aplicación**

Tancovich es un juego multijugador local sobre tanques de guerra.

El objetivo del juego es destruir al/los oponente/s a través de las armas que posee el tanque.

Cada tanque posee 2 armas, misiles o minas de tierra. Al lanzar los misiles, estos pueden rebotar 4 veces sobre cualquier superficie antes de explotar, cuando un misil alcanza a un enemigo o a uno mismo explota inmediatamente provocando daño.

En cambio las minas, se dejan caer sobre la tierra, cuando un tanque pasa sobre alguna de ellas genera una explosión provocando daño al tanque que pasó sobre ella.

Cada tanque posee una barra de vida y 3 corazones, al provocar daño con las armas, la barra de vida va disminuyendo, cuando esta llega a 0 se resta un corazón. Cuando se llega a 0 corazones el tanque pierde la partida.

## **Proceso de desarrollo**

Teníamos claro que íbamos a hacer un juego, decidimos que iba a ser de pelea de tanques por lo que empezamos a codear e investigar que debía tener un juego. Con eso en mente, comenzamos a diagramar las primeras clases y de a poco a implementar Swing aprendiendo a medida que íbamos codeando. Nuestro primer paso fue imprimir sobre un fondo negro la imagen del tanque y lograr que se ejecute constantemente para lograr un movimiento. A medida que íbamos agregando las funciones que necesitábamos para lograr nuestro objetivo, fuimos agregando clases y métodos.

Una vez que logramos que el tanque se mueva y dispare, para ir probando esas funciones creamos una clase Enemy para testear el choque de los misiles con otro objeto.

Nuestro siguiente paso fue mejorar el movimiento de la imagen a cuando el tanque se mueve, logrando cambiar la rotación de la misma y utilizando el mismo método para desde donde salen los misiles. Cuando fuimos probando ese método de misiles, descubrimos cómo hacer que el tanque deje minas así que creamos la clase y logramos agregar la clase.

Luego, creamos cajas para hacer que las balas reboten donde queríamos, acá fue donde tuvimos nuestras mayores dificultades calculando las físicas de las balas y logrando que estas reboten donde queríamos sin que haya ningún inconveniente.

Una vez que tuvimos todo esto funcionando, nos adentramos en poder agregar un player 2 por lo que tuvimos que agregar las funciones correspondientes a sus interacciones.

Cuando tuvimos esto finalizado, arrancamos a crear el menú y agregar métodos para hacer el juego mucho más completo (barras de vida, número de vidas) y fuimos puliendo el código para hacerlo más eficiente ya que teníamos atributos incluso sin getters y setters.

Nuestro último paso fue el crear el menú. Para lograrlo investigamos el uso de enum para ir cambiando el estado del juego y que se imprima en pantalla lo que queríamos.

Hubo cosas que dejamos en el “tintero” como por ejemplo poner música y sonido a los sprites. Pudimos crear la clase pero no tuvimos tiempo a implementarla. Otra cosa que queríamos hacer era agregar mas players y agregar distintos tipos de misiles pero por la misma razón que el sonido, no pudimos hacerlo.

## **Informe técnico**

El sistema consta de trece clases:

Por un lado tenemos la clase llamada Sprite la cual es clase padre de todos los “actores” del juego. Posee métodos de movimiento, carga de imágenes de los mismos, Testeo de colisiones, y también de efecto de explosión.

Las cuatro clases que heredan de Sprite e implementan Entity son:

- Tank
- Mine
- Missile
- Hearts

Estos son los “actores” del juego es decir los objetos que producen acciones e interactúan en el juego.

Poseen métodos de inicialización (inicializa los movimientos de cada clase) de actualización (clase update, actualiza los movimientos en tiempo real a su vez del estado de las clases a través del método isVisible). Estos últimos dos métodos son implementados a través de la interfaz de Entity. También posee un método de check Controls que lee las teclas y las traduce en acciones.

A su vez la clase Tank contiene a las otras clases ya que es quien las utiliza y las “activa” a través de la lectura del teclado.

La clase más importante del sistema es Board. Extiende de JPanel (una clase de Swing\*) e implements ActionListener.\*

Se encarga del bucle del sistema, a través de la clase Timer que es manejada por threads. Lee los inputs del teclado, del mouse con KeyListener\* y MouseListener\* (contenida en la clase MouseInput).

Con enum de estado, maneja los cambios de renders entre los diferentes menús y también la ejecución del juego.

A través del método ActionPerformed ejecuta continuamente todo lo que tiene que ver en el juego, los tanques, las minas, los misiles, las vidas y las colisiones entre los sprites.

A través de los métodos de JPanel, la clase Board dibuja los objetos (tanques, misiles, minas, vidas) además de que en esta se crean distintos objetos que interactúan con el mapa (barras de vida, cajas).

## **Manual del Usuario**

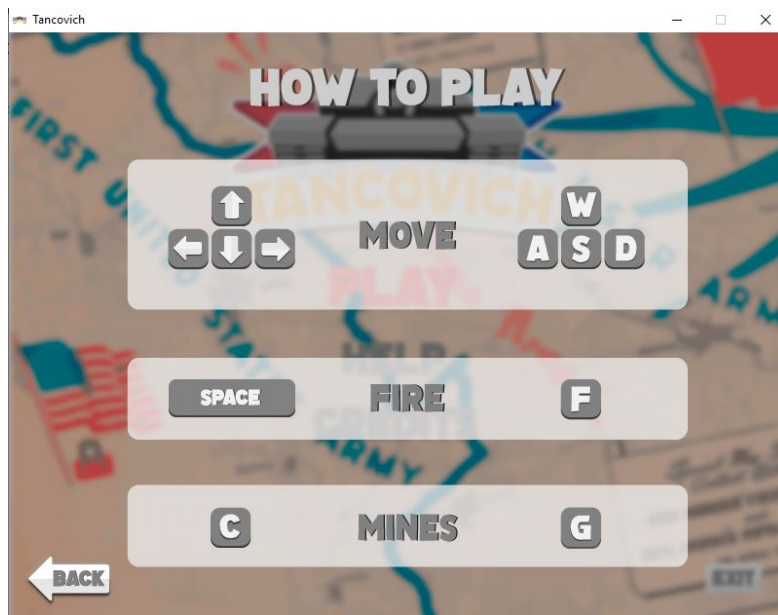
1-Apenas se ejecuta el programa, la primera interfaz que se observa es el inicio del juego. Presionando la tecla enter o haciendo click sobre el botón se puede avanzar a la siguiente pantalla:



2-Luego de presionar “ENTER” la siguiente pantalla es el menú principal desde donde se puede iniciar el juego, ver los controles del juego, ver los créditos y salir. Si se presiona el botón “BACK” se vuelve a la pantalla anterior, si presiona el botón “EXIT” cierra el programa:



3-Si se presiona el botón “HELP” se mostrará la pantalla de controles:

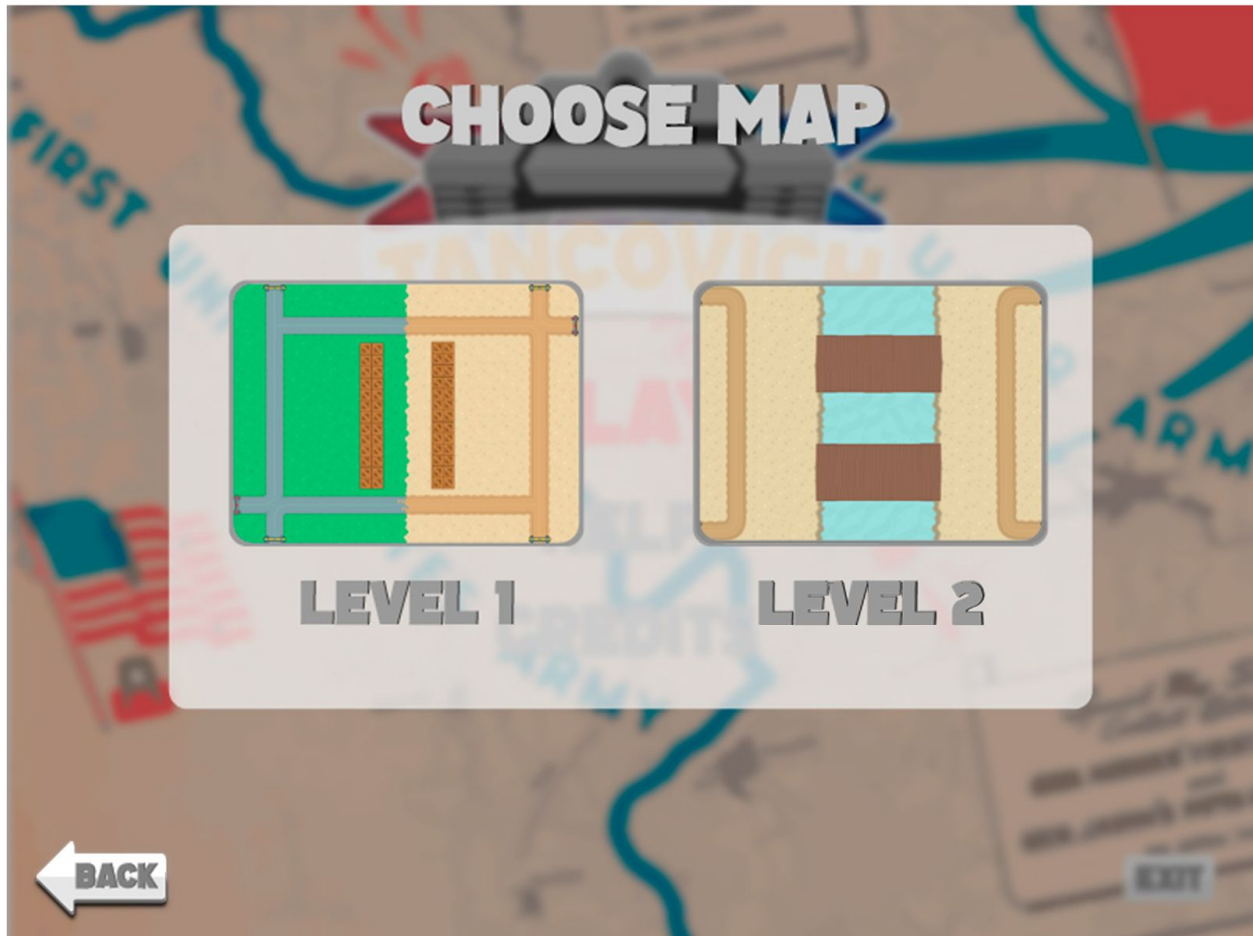


4-Si presiona el boton "CREDITS" se mostrara la pantalla de los credits:



5-Si presiona el boton "PLAY" se pasara a la pantalla de elegir nivel:





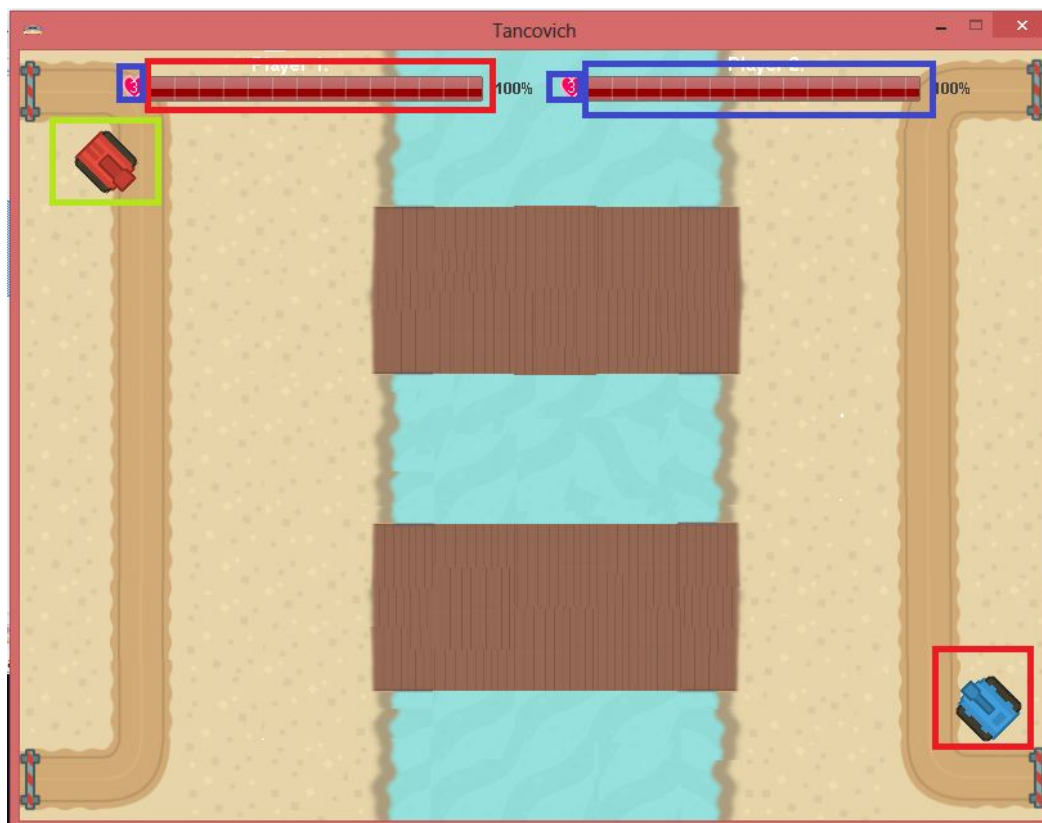
6-Luego se iniciara el juego con la respectiva cantidad de tanques y con el nivel elegido:

Level 1:





Level 2:

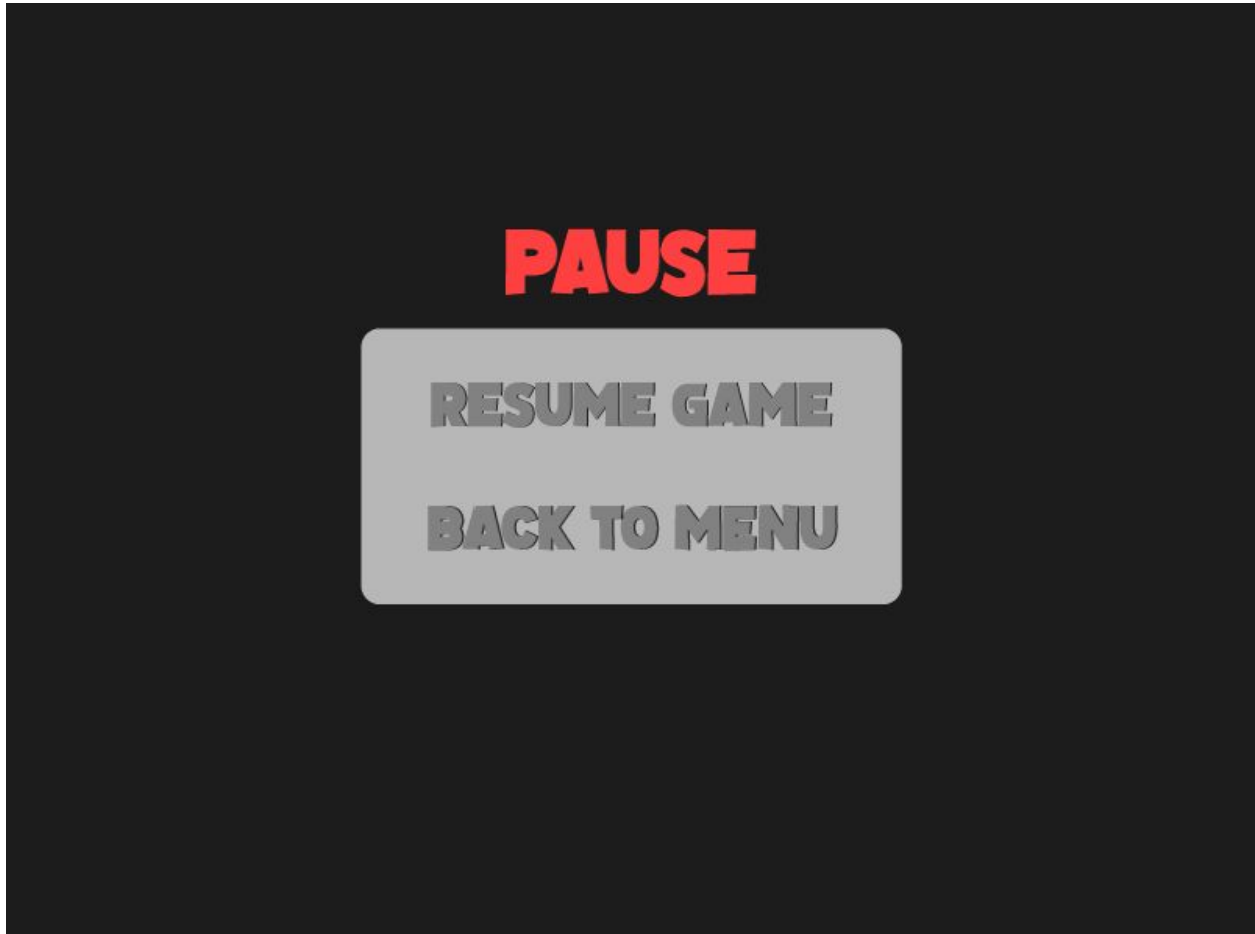


-Barras de vida

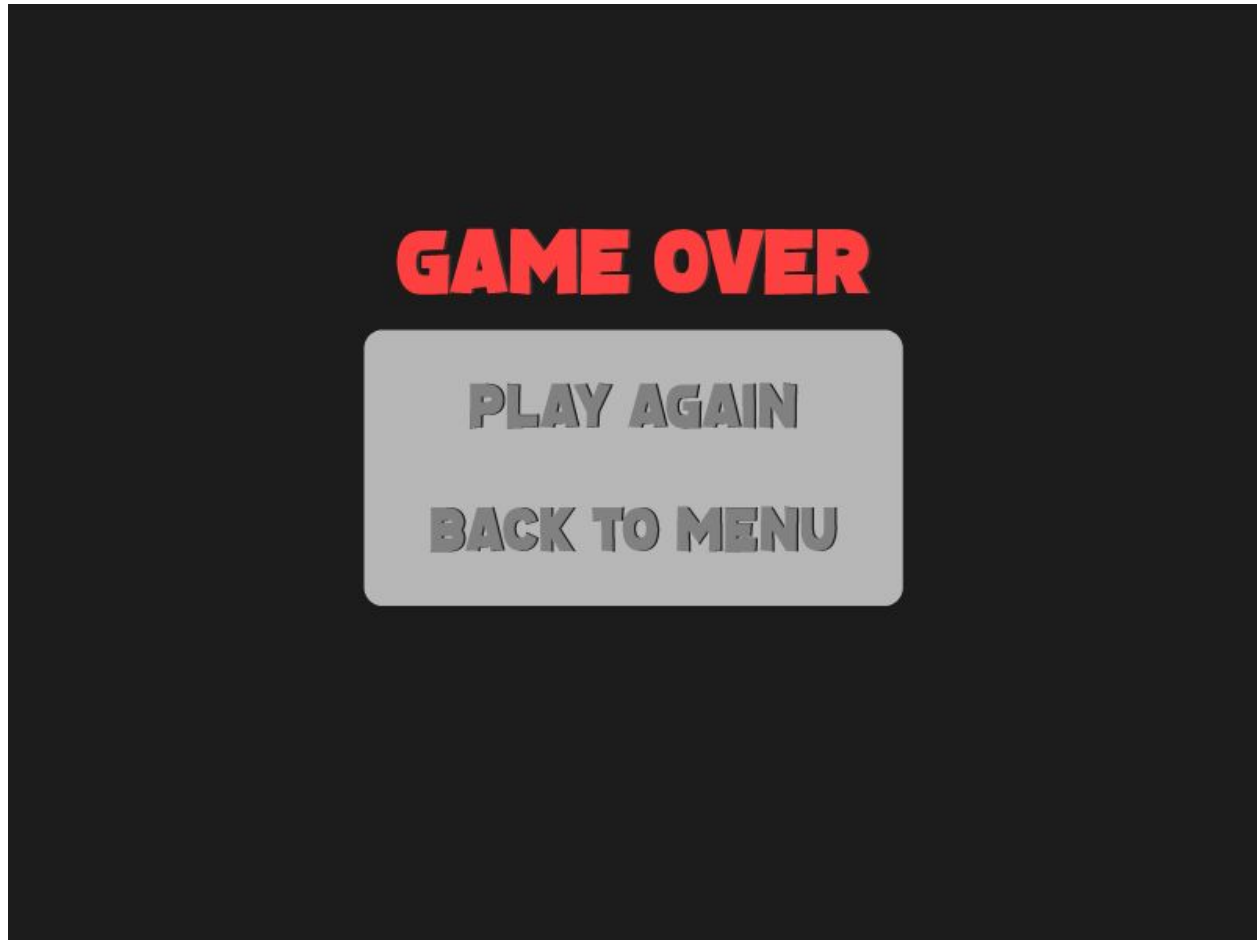
-Corazones

-Tanques

7.1- Durante el juego podrás poner pausa y elegir entre estas dos opciones:



8-Al quedar un solo tanque vivo el juego finaliza mostrando un cartel de "Game over".



### **Bibliografía consultada**

- <https://opengameart.org> Sprites.
- <https://www.stackoverflow.com> Dudas diversas.
- <https://www.youtube.com/watch?v=FZWX5WoGW00> Base para el menú.