Universidad Autónoma de Yucatán

Facultad de matemáticas

Licenciatura en Ingeniería de Software

Programación estructurada

**Segunda entrega**

Integrantes:

William Cetina

Gener Echeverría

Jesús Estrella

15/05/2020

Índice

[**Descripción del problema**](#_nwvc38hlx664) **3**

[**Requerimientos funcionales**](#_5kapqo9h75ec) **4**

[**Requerimientos no funcionales**](#_joceusi7excl) **7**

[**Casos de uso**](#_mw487wp1a3vb) **8**

[**Plan de trabajo**](#_x20sm0sysxd1) **16**

# Descripción del problema

El proyecto consiste en un juego llamado “Quicker” cuya mecánica principal es obtener el mayor puntaje de minijuegos completados en una sucesión rápida. Dichos minijuegos van aumentando de dificultad a medida que se completan y, en caso de no ser completados, el jugador pierde una vida. Finalmente el juego termina cuando el jugador pierde 3 vidas.

El juego tendrá 4 minijuegos:

* **Pong:** simulará un tenis donde el jugador tenga control de una paleta que se mueve verticalmente en la parte izquierda de la pantalla y competirá contra otra paleta del lado derecho. El jugador utiliza la raqueta para pegarle a la pelota y devolverla al otro lado. El jugador pierde si no logra devolver la pelota.
* **Preguntas:** presentará una pregunta sobre operaciones matemáticas en el cual el jugador deberá escribir la respuesta correcta para conseguir un punto. Pierde una vida si no responde en 10 segundos o si su respuesta es errónea. Disminuye el tiempo límite para responder la pregunta conforme el jugador completa minijuegos.
* **Asteroids:** El jugador controla de manera horizontal y vertical una nave cuyo objetivo es esquivar los asteroides que irán apareciendo en la parte superior de la pantalla con dirección descendente. El jugador obtiene un punto si logra esquivar los asteroides durante 10 segundos, de lo contrario, pierde una vida.
* **Gato:** también conocido como tres en línea o Tic Tac Toe, consiste en un tablero de 3x3 en donde el jugador selecciona la casilla donde colocará su símbolo (X), el jugador obtiene un punto si logra realizar una línea recta o diagonal de su símbolo,y pierde una vida si la computadora logra dicha recta o diagonal. En caso de surgir un empate, la vida y el puntaje no cambian.

Al terminar cada minijuego, se desplegará una pantalla de 2 segundos de duración que muestra la vida y el puntaje del jugador, y pasará a otro minijuego, mientras el jugador tenga vidas.

# 

# Requerimientos funcionales

**Generales:**

Al iniciar el programa, se despliega el menú principal del juego, el cual contiene la opción **Empezar** y **Salir**,

El programa reconocerá las teclas **W, A, S y D** para el movimiento (**W** = arriba**, A** = izquierda**, S** = abajo**, D**  = derecha) y **ENTER** para la confirmación de acciones. En el minijuego de preguntas también reconocerá las teclas de números.

La opción **Empezar** ejecutará la función **Selec** y la opción **Salir del juego** ocasionará que programa finalice.

El programa inicializa el puntaje del jugador en cero y el número de vidas en tres.

Al ejecutar la función **Selec**, el programa comenzará un ciclo donde se mostrará la pantalla de puntaje y se escogerá de manera aleatoria el minijuego que se ejecutará.

El programa aumentará en uno el puntaje del jugador cuando complete exitosamente un minijuego.

El programa disminuirá en uno el número de vidas del jugador cuando falle en completar un minijuego.

El programa cambiará el color de los caracteres en pantalla a amarillo si el número de vidas es de dos.

El programa cambiará el color de los caracteres en pantalla a rojo si el número de vidas es de uno.

Si la vida del jugador llega a cero, el programa detendrá la función Selec y desplegará la pantalla de game over y volverá al menú principal.

La pantalla de game over mostrará el puntaje total de minijuegos completados.

Función **Pong**:

El programa imprime ambas paletas y establece el límite de la cancha.

El jugador moverá su raqueta verticalmente utilizando las teclas **W** y **S**.

El programa moverá la raqueta contraria de arriba hacia abajo.

El programa moverá la pelota de un lado a otro del campo, cada vez que una raqueta la toque.

El programa termina el minijuego cuando la pelota choque con el límite del campo posterior a cada raqueta, quita una vida si es del lado del jugador y aumenta en uno el puntaje si es del contrario.

Función **Asteroid:**

El programa limita el campo de juego y crea inicialmente varios asteroides en la parte superior del campo y una nave en la parte inferior.

El programa desciende los asteroides, los cuales desaparecen al tocar el límite inferior del campo y reaparecen en la parte superior

El jugador moverá la nave utilizando las teclas **W,** **S, A, D**.

El programa termina el minijuego cuando la nave choque con un asteroide o si se mantiene sin chocar por 10 segundos. En el primer se resta una vida al jugador, en el segundo, aumenta en uno el puntaje.

Función **Preguntas:**

El programa muestra de manera aleatoria una de 10 preguntas.

Cada pregunta tiene una única respuesta que debe ingresarse por teclado.

El programa disminuye en uno la vida del jugador si su respuesta es errónea o si no contesta en 10 segundos.

El programa aumenta en uno el puntaje del jugador si su respuesta es correcta.

Función **Gato:**

El programa imprime un tablero 3x3 que se actualiza después de cada turno.

El programa coloca la ficha ´X´en la casilla disponible que el jugador ha escogido.

El programa coloca aleatoriamente la ficha ´O´ en alguna de las casillas disponibles.

El programa evalúa en cada turno si se ha formado alguna recta o diagonal con la misma ficha.

El programa aumenta en uno el puntaje del jugador si logra formar una recta o diagonal con el símbolo ‘X’.

El programa disminuye en uno la vida del jugador si se forma una recta o diagonal con el símbolo ‘O’.

El programa mantiene el puntaje y la vida del jugador previo al minijuego si no se forma ninguna recta o diagonal del mismo símbolo.

# 

# Requerimientos no funcionales

El programa podrá ser ejecutado en equipos con compiladores de lenguaje C.

El programa únicamente podrá funcionar si el compilador cuenta con todas las librerías incluidas en el código.

El sistema es desarrollado con las herramientas que la programación en C ofrece.

El sistema debe estar disponible para todo aquel interesado en experimentar su contenido.

Los textos que aparezcan en la pantalla deberán ser totalmente legibles y entendibles para el usuario que esté ejecutando el programa.

El programa se encuentra en idioma español.

# Casos de uso

Este programa permite ejecutar 4 minijuegos distintos los cuales serán escogidos aleatoriamente. Dependiendo del resultado y vidas del jugador, el programa determina el ciclo de ejecucion de juegos.

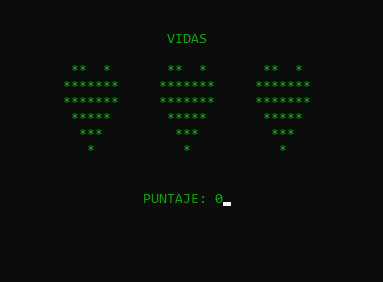
**Flujo de eventos**

|  |  |  |  |
| --- | --- | --- | --- |
| Paso | Actor | Descripción | Evento del sistema |
| 1 | Usuario | Selecciona la opción **Empezar juego** | Se despliega la pantalla principal y el juego pasa a la función Selec.  **Ver pantalla 1** |
| 2 | Máquina | Pantalla de puntaje | La función Selec despliega la función puntaje con valores iniciales.  **Ver pantalla 2** |
| 3 | Máquina | Selecciona el juego que ejecutara | Mediante un algoritmo, la función **Selec** escoge aleatoriamente uno de los cuatros minijuegos disponible y lo ejecuta para que el usuario juegue. |
| 4 | Usuario | Corre el juego **Asteroid** y mueve las nave con las teclas **A, W, S** y **D** | El programa ejecuta la funciòn **Asteroid,** y actualiza en pantalla la posicion de la nave y de los asteroides (descienden). SI el jugador no choca con un asteroide durante 10 segundos, el programa aumenta en uno su puntaje, si no, pierde una vida.  **Ver pantalla 4.1.1,4.1.2,4.1.3** |
| 4 | Usuario | Corre el juego **Pong** y mueve la raqueta con las teclas **W** y **S** | El programa ejecuta la funciòn **Pong**. Si el usuario no logra darle a la pelota, pierde una vida. Si la raqueta opuesta no logra devolverla, aumenta en uno el puntaje del jugador.  **Ver pantalla 4.2** |
| 4 | Usuario | Corre el juego **Preguntas** e ingresa por teclado la respuesta | El programa ejecuta la funciòn **Preguntas** y muestra aleatoriamente una de 10 preguntas. Si el jugador responde correctamente aumenta en uno su puntaje, si pasan 10 segundos sin ingresar respuesta o responde incorrectamente, pierde una vida.  **Ver pantalla 4.3** |
| 4 | Usuario | Corre el juego **Gato** y selecciona ingresando por teclado el número de la casilla del tablero donde desea colocar su ficha. | El programa ejecuta la funciòn **Gato** y actualiza en cada turno la posición de las fichas en el tablero. Si el jugador forma una recta o diagonal con el símbolo X, aumenta en uno su puntaje, si el programa forma una recta o diagonal con el símbolo O, el jugador pierde una vida. Si ninguno de los dos forma una recta o diagonal de un mismo símbolo, se considera empate y la vida y vidas del jugador no se modifican,  **Ver pantalla 4.4** |
| 5 | Máquina | Muestra los resultados. Juego ganado. | Al ganar el minijuego, la pantalla resultado muestra el puntaje aumentado en un punto y las vidas restantes, posteriormente la función **Selec** escoge aleatoriamente otro juego para ejecutar.  **Ver pantalla 5.1** |
| 5 | Máquina | Muestra los resultados. Juego fallado. | Al perder el minijuego, la pantalla resultado muestra la vida disminuido en uno y los puntos obtenidos hasta el momento, en caso de aún tener vidas es enviado a la funciòn **selec** para ejecutar otro juego.  **Ver pantalla 5.2 y 5.3** |
| 6 | Máquina | Pantalla de Game Over | En caso de no poseer más vida, la función gameover muestra el puntaje final y se despliega la pantalla inicial.  **Ver pantalla 6** |
| 7 | Usuario | Selecciona la opción **Salir** | Se termina el programa.  **Ver pantalla 7.1 y 7.2** |

**Pantallas del módulo**



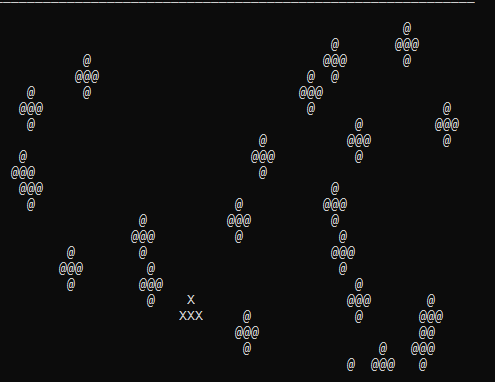
Pantalla 1



Pantalla 2



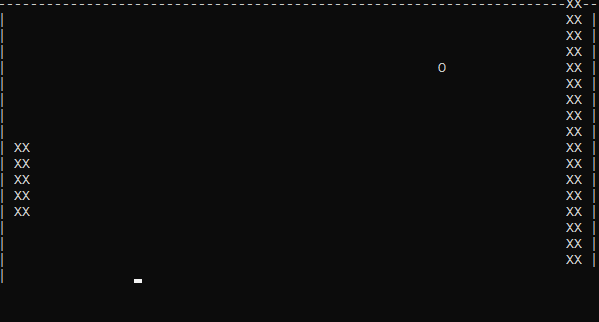
Pantalla 4.1.1



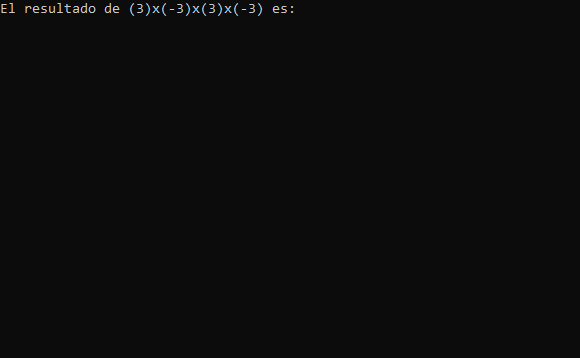
Pantalla 4.1.2



Pantalla 4.1.3



Pantalla 4.2



Pantalla 4.3



Pantalla 4.4



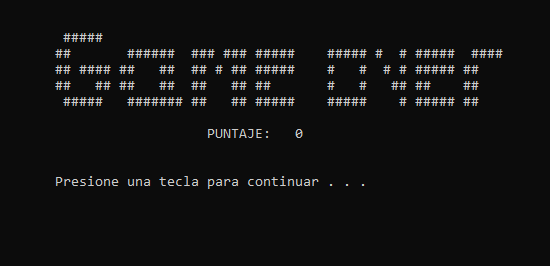
Pantalla 5.1



Pantalla 5.2



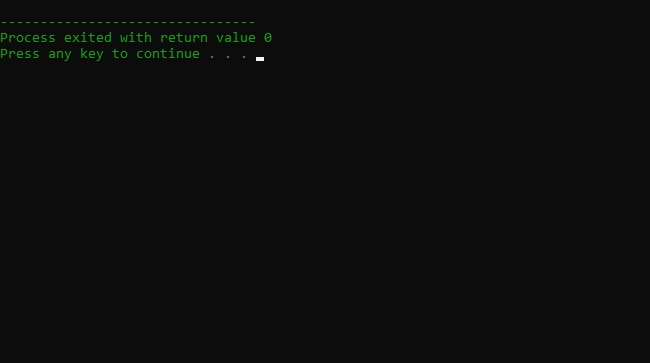
Pantalla 5.3



Pantalla 6



Pantalla 7.1



Pantalla 7.2

**Video de prueba:**

<https://youtu.be/jUGq_ElaQfk>

# Plan de trabajo

|  |  |  |
| --- | --- | --- |
| Actividades | Productos/Artefactos resultantes | Responsable |
| Junta para definir roles y variables  17 de febrero |  | Todo el equipo. |
| Desarrollo de pantalla de inicio y estructura general (Función main y selec).  18 abril | Función main.  Función selec | Gener Echeverría |
| Junta para revisión de Función main y división de los minijuegos.  19 abril | Documentación Función main. | Todo el equipo |
| Programación funciones anexas  21 abril. | Función Pantalla inicio  Función Puntaje.  Función GameOver | Gener Echeverría |
| Programación de cada minijuego.  21 a 30 de abril. | Función Pong.  Función Asteroids.  Función Preguntas.  Función Gato. | Pong: Jesús Estrella  Asteroids: William Cetina  Preguntas: Gener Echeverría  Gato: Gener Echeverría |
| Junta para revisión de minijuegos.  1 de mayo de marzo. | Documentación Pong.  Documentación Asteroids.  Documentación Preguntas  Documentación Gato | Junta: Todo el equipo |
| Actualización función asteroid  5 de mayo | Segunda versión de Asteroid (movimiento vertical) | William Cetina |
| Integración de funciones anexas a función main.  11 de mayo | Versión sin minijuegos. | Gener Echeverría |
| Actualización función pong  12 de mayo. | Segunda versión de Pong (exportar resultado de la partida) | Jesús Estrella |
| Integración de minijuegos a función main.  12 de mayo | Primera versión jugable | Gener Echeverría |
| Revision de primera versión jugable  13 de mayo | Actualizacion de primera versión jugable | Jesús Estrella |
| Revision de documentacion  14 de mayo | Actualización del documento | Todo el equipo |
| Junta de testeo.  Por definir | Feedback externo. | Todo el equipo |

\*Fechas sujetas a cambio

Repositorio

Se decidió el uso de Github como repositorio ya que es una herramienta profesional que permite trabajar en colaboración con otras personas, planificar proyectos y realizar un seguimiento del trabajo. En el apartado del proyecto se agregaran todos aquellos documentos que se requerirán para elaborar el trabajo, así como links de vídeos, productos y toda aquella información que nos ayude en el proceso.

<https://github.com/GenerEcheverria/Estructurada>

Comunicación

Para expresar opiniones, compartir ideas o discutir puntos clave del proyecto, se busca una interacción eficaz entre los diferentes miembros del grupo mediante el uso de las plataformas descritas en el apartado de herramientas y por medio de las reuniones indicadas en el calendario. Se utilizó un grupo de WhatsApp, uno de Discord y reuniones presenciales para manifestar comentarios, resolver dudas o hacer aclaraciones que puedan surgir entre reuniones con el objetivo de reducir errores y retrasos.

Monitonitoreo

Las diferentes tareas y actividades requeridas para cumplir con nuestros objetivos se asignan a cada integrante con base en sus habilidades y áreas de oportunidad. Esto se podrá adecuar y modificar de acuerdo con la opinión grupal y coevaluación que se realiza en cada reunión.

De igual forma, el calendario de actividades servirá como guía principal en el orden de las actividades que se lleven acaba. Sin embargo, si el equipo decide que es necesario realizar modificaciones al mismo, se harán las modificaciones necesarias.

Herramientas

* Google forms: (Para trabajar de manera simultánea documentación del proyecto).
* WhatsApp: (Aplicación utilizada para mantener contacto grupal las 24 horas y resolver dudas menores).
* Discord <https://discord.gg/4W83yP> (El objetivo de usar esta plataforma es mantener comunicación eficaz durante la elaboración de documentos de manera no presencial).
* Trello h[ttps://trello.com/invite/b/cGjtn6SA/7185b1827e89084c7b39e801114292bc/estructurada](https://trello.com/invite/b/cGjtn6SA/7185b1827e89084c7b39e801114292bc/estructurada) (El uso de esta herramienta surge por la necesidad de llevar una mejor organización de las actividades a realizar, asignar roles y estar al tanto del estatus de cada tarea).