

Tarea 1. Mobile Maze Bootable

1st Acevedo Rodríguez Kevin
Instituto Tecnológico de Costa Rica
Principios de Sistemas Operativos
Cartago, Costa Rica
kevinar51@estudiantec.cr

2nd Camacho Hernández José Julián
Instituto Tecnológico de Costa Rica
Principios de Sistemas Operativos
Cartago, Costa Rica
jcamacho341@estudiantec.cr

3rd Venegas Vega Jose Agustín
Instituto Tecnológico de Costa Rica
Principios de Sistemas Operativos
Cartago, Costa Rica
joseagustinvenevag@estudiantec.cr

4th Solís Arguello Juan Antonio
Instituto Tecnológico de Costa Rica
Principios de Sistemas Operativos
Cartago, Costa Rica
JuanSa@estudiantec.cr

Resumen—This document contains the whole development of an assembly-bootable game. This is also going to include the necessary theoretic concepts needed to develop the game and boot loader, specifying development environments, project attributes, and design details. Also contains the distribution of the tasks per student, specifying hours per task and the final status of every task. In the end, the team added some suggestions, advices for this specific process, followed by some conclusions also added by the team.

I. INTRODUCCIÓN

En el presente documento explicaremos la creación de un Mobile Maze bootable, donde buscaremos tratar los procesos en el booteo de una computadora mediante la implementación de un programa realizado en el ensamblador.

El booteo de una computadora es un es el proceso que ocurre cuando encendemos el equipo y se carga el sistema operativo. Durante este proceso, la computadora realiza varias operaciones en secuencia, que incluyen la inicialización de hardware, por lo tanto es de suma importancia el manejo de estos conceptos como ingenieros en computadores.

Otro aspecto importante que trabajaremos en el presente paper son las interrupciones las cuales permiten una integración de usuario con el sistema operativo de manera más amigable.

Por lo tanto manejar conceptos y flujo de información de los procesos en el booteo de una computadora mediante la implementación de un programa en ensamblador y la utilización de las interrupciones más comunes en una arquitectura nos va permitir entender de manera más detallada cómo funciona el sistema operativo y cómo interactuar con él.

II. AMBIENTE DE DESARROLLO

A continuación se detalla la configuración básica necesaria para la ejecución del proyecto.

El juego fue desarrollado en un ambiente Linux, por tanto fue posible la utilización del conjunto de herramientas base que ofrece dicha plataforma como GNU, *gcc*, entre otras bibliotecas básicas.

Adicionalmente, entre las herramientas, *frameworks*, bibliotecas y demás aplicaciones de desarrollo que fueron utilizadas

para la implementación del proyecto se encuentran las siguientes:

- **NASM:** Ensamblador para la arquitectura de CPU x86 portátil para la mayoría de las plataformas modernas. Fue utilizado para ensamblar el código de x86 desarrollado y así obtener un binario.
- **QEMU:** Emulador de máquinas de código abierto, que puede ejecutar sistemas operativos y programas para una máquina en otra diferente. En este proyecto, fue ampliamente utilizado para realizar las pruebas del juego y *bootloader* sin tener que reiniciar la computadora cada vez.
- **GitHub:** Plataforma que contiene el repositorio de la tarea. Fue de gran utilidad para el manejo de versiones, la sincronización y el acceso del código actualizado para los miembros del equipo.
- **GitKraken:** Software que facilitó el manejo del repositorio en *GitHub*, al brindar una representación visual de aspectos importantes como las *branches*, *commits*, entre otros.
- **Visual Studio Code:** Editor de texto que fue útil en el manejo y programación de los diferentes archivos de código ensamblador, *makefile*, entre otros.

III. ATRIBUTOS

III-A. Aprendizaje Continuo

Como parte del aprendizaje continuo, en la elaboración de la presente tarea fue necesario aplicar parte del conocimiento adquirido en anteriores cursos de la carrera.

En primer lugar, fue fundamental y de gran valor aplicar técnicas aprendidas en el curso Especificación y Diseño de Software. Entre ellas, se encuentra la identificación y distribución de tareas a partir de una especificación del trabajo. Así como el reconocimiento de los requerimientos tanto funcionales como no funcionales del sistema, y la capacidad de identificar requerimientos faltantes para así recolectarlos oportunamente.

En el proceso de diseño del programa, conceptos estudiados en cursos previos, como Taller de Diseño Digital, fueron de de

gran valor para comprender y modelar la solución al problema antes de empezar su implementación en código.

Adicionalmente, fue de gran importancia aplicar conocimientos de lógica en el flujo de desarrollo de software. Esto debido a que, las capacidades para comprender y codificar el funcionamiento de nuevos programas como *bootloaders* y juegos en lenguaje ensamblador, fue fundamental.

Por otro lado, las estrategias de manejo de código utilizando *git* aprendidas con anterioridad en la carrera, fueron de gran relevancia para la administración del código en diferentes *branches* y para la prevención de conflictos entre las mismas.

Finalmente, conceptos básicos de sistemas operativos aprendidos con anterioridad, y los vistos en el presente curso de Principios de Sistemas Operativos, como el tema de interrupciones a nivel de *software* o llamadas al sistema, también fueron de gran valor para el desarrollo del proyecto.

III-B. Trabajo individual y en equipo

A continuación se describen los atributos que fueron reforzados durante el desarrollo del proyecto.

III-B1. Estrategias para el trabajo individual:

- **Establecer metas de trabajo:** De manera individual se deberá de mantener una idea de lo que se quiere lograr en un día de trabajo, se deben de establecer metas con el objetivo de cumplir de manera individual el trabajo final de los compañeros.
- **Trabajar de manera organizada:** El código debe de estar en las mejores condiciones posible cuando se haga un commit, un merge o cualquier interacción con git el código debe de estar en las mejores condiciones posibles, comentando, sin bugs, y desarrollando las tareas necesarias.

III-B2. Estrategias para trabajo en equipo:

- **Establecer objetivos claros:** como equipo es importante que todos los miembros tengan los objetivos claros, como también las metas las cuales deseamos alcanzar y de esta manera poder trabajar todos en equipo con el cumplimiento de estas.
- **Identificar roles y responsabilidades:** cada miembro del equipo debe tener una comprensión clara de su papel en el proyecto, incluyendo las tareas específicas que deben realizar y las responsabilidades que tienen en el proceso.
- **Comunicación abierta:** la comunicación es de suma importancia para la resolución de problemas y el desarrollo del proyecto, puesto que una buena comunicación permite que todos puedan compartir sus ideas y opiniones.
- **Establecer plazos:** establece plazos para asegurarte de que el proyecto avance según lo planificado. Esto también puede ayudar a mantener a los miembros del equipo responsables de sus tareas y garantizar que se cumplan los plazos.
- **Resolver conflictos:** En cualquier forma de trabajo en equipo, pueden surgir conflictos. Es importante abordar los conflictos de manera abierta y honesta para encontrar soluciones y evitar que afecten el progreso del proyecto.

- **Problemas:** En caso de tener algún tipo de dudas es de suma importancia que los compañeros se comuniquen para avanzar de la manera más precisa y rápido con el desarrollo del código, puesto que tal vez algún otro compañero ya haya resuelto ese problema o tenga la posibilidad de ayudar a buscar la solución
- **Celebrar el éxito:** Cuando el proyecto esté completo, es conveniente de celebrar el éxito del equipo.

III-B3. Planificación del trabajo mediante la identificación de roles, metas y reglas:

- Antes de empezar con el desarrollo se decidieron estructurar roles, reglas y metas en base a una serie también estructurada de tareas, ordenadas por dependencias e importancia. Se decidió que uno de los integrantes debería tomar el rol de líder, el cual debería cumplir la supervisión del cumplimiento de tareas y de los aportes aporte tanto grupales como individuales, así también como velar por la armonía en el ambiente de desarrollo constante implementado.

Se definieron fechas límite para cada tarea, de modo que se definió junto con esto la regla de no exceder dicha fecha límite ya sea sin avances o un bloqueante de peso. De igual modo, el equipo consideró necesaria adición de 3 reuniones semanales para evaluar avances y unificar tareas completadas.

III-B4. Acciones que promueven la colaboración entre los miembros del equipo: La implementación de reuniones periódicas para la evaluación de avances fue un claro incentivo para la constancia del equipo, al mismo tiempo, los parámetros definidos de tiempo ayudaron al cumplimiento de las tareas en los intervalos definidos ayudó de igual forma al compromiso.

Por otro lado, se considera que haber dado la posibilidad de que en caso de no llegar a finalizar una tarea en el tiempo acordado, poder indicar la problemática encontrada y buscar una solución en equipo fue una de las reglas implementadas que permitió a los integrantes invertir tiempo en cada una de sus tareas sin sentir emociones negativas que puedan repercutir al rendimiento como el estrés o la presión.

Adicionalmente, la constante comunicación mediante los medios sociales destinados al desarrollo de esta tarea (Grupos de WhatsApp y Discord), contribuyó al compromiso y la satisfacción.

III-B5. Manera de ejecutar las estrategias planificadas:

El compañero Kevin tomó un rol de líder para el desarrollo del proyecto lo cual nos permitió tener un control de las tareas y una organización adecuado con el proyecto.

No solo esto, sino también todos los integrantes del grupo estuvieron muy acertados con el cumplimiento de sus labores y cada uno de sus entregables.

La primera estrategia que se utilizó para el desarrollo del proyecto fue un desarrollo de cascada en el cual primero analizamos los requisitos, diseñamos, implementamos y realizamos las pruebas necesarias para la ejecución del proyecto.

Aparte de esto utilizamos una especie de scrum en un único sprint por la duración del proyecto donde la idea era

implementar el proyecto en conjunto y con reuniones de alta frecuencia para buscar el desarrollo más eficiente y con una alta calidad del software.

III-B6. Evaluación para el desempeño del trabajo: Se define la calificación de manera individual según los siguientes factores:

- La resolución de sus tareas en el tiempo establecido.
- La constancia y el interés en el desarrollo del trabajo.
- El nivel de apoyo y solidaridad para ayudar a otros compañeros con sus tareas.
- La asertividad para pedir ayuda con sus tareas en caso de no poder lograrlas.

Seguidamente, se evalúa el desempeño individual de los integrantes del equipo con base en lo planteado anteriormente.

Evaluación	Kevin	Agustín	Juan	Julián
Resolución de sus tareas en el tiempo establecido	5	5	5	5
Constancia y el interés en el trabajo	5	5	5	5
Solidaridad para ayudar a otros	5	5	5	5
Asertividad para pedir ayuda	5	5	5	5

III-B7. Evaluación para las estrategias utilizadas de equidad e inclusión: La implementación de reuniones periódicas, la adición de grupos en medios sociales para la comunicación constante, es un incentivo para la participación y el involucramiento de los integrantes en todas las actividades referentes al desarrollo de la tarea.

Esto permitió que todos los integrantes se sintieran parte del equipo, y comprometidos con el objetivo, fomentando así la participación, el buen flujo de trabajo y el bienestar individual.

III-B8. Evaluación para las acciones de colaboración: Para determinar la valoración del trabajo en equipo, se toma en consideración lo siguiente:

- Interés en el avance grupal.
- El mantenimiento sano del ambiente de trabajo con los como compañeros de grupo.
- La atención a las reuniones y aportes significativos en las mismas.
- Ayuda cuando se le solicita de manera acertada y sin faltas de respeto
- Cumplimiento de tareas para lograr el objetivo común.

A continuación, se evalúa el desempeño general del equipo basado en los anteriores factores. Estas fueron dadas de manera tal que los compañeros calificaron cada uno de los factores de manera grupal y dieron como resultado el valor promedio de este.

Evaluación	Kevin	Agustín	Juan	Julián
Interés en el trabajo grupal	5	5	5	5
Ambiente de trabajo sano	5	5	5	5
La atención a las reuniones y aportes significativos	5	5	5	5
Ayuda cuando se le solicita	5	5	5	5
Cumplimiento de tareas para lograr el objetivo común	5	5	5	5

IV. DETALLES DEL DISEÑO DEL PROGRAMA DESARROLLADO

Abarcando inicialmente el *Bootloader*, se implementó este apartado por 'capas' es decir, al empezar, se desarrolló una versión inicial del mismo, el cual se encargaba de imprimir un mensaje simple, esto fue probado mediante emuladores, una vez obtenido esto, se hicieron las modificaciones necesarias para que este pudiese ser cargado en hardware real. Logrado esto último, se tenía una versión estable del *Bootloader*, la cual posteriormente se uniría a la aplicación desarrollada.

Debido a necesidades de la aplicación, se requirió una implementación temprana entre el apartado *Bootloader* y aplicación, debido a las restricciones de memoria. Para poder añadir features nuevos a la aplicación sin la preocupación de exceder una cantidad máxima de memoria, se implementó lo más pronto posible el *Bootloader* a la aplicación, con el objetivo de desentenderse parcialmente de las limitaciones de memoria.

La implementación se realizó mediante la concatenación de los binarios generados a partir de los archivos en ensamblador, formando así un binario resultante que contendrá el "Mobile Maze Booteable", esto se hará automáticamente mediante el Makefile.

En cuanto la jugabilidad la parte principal fue la creación de los muros los cuales fueron dibujados en los diferentes píxeles correspondientes, en la jugabilidad se esperaba que si el jugador colisionaba con alguno de estos muros el juego volvería a su estado inicial.

Luego de esto las teclas cumplen con un funcionamiento muy sencillo si se presiona alguna de esta el movimiento del cuadro cambia según la tecla, esto se puede ser de manera más sencilla en el diagrama de flujo correspondiente con la fig 1

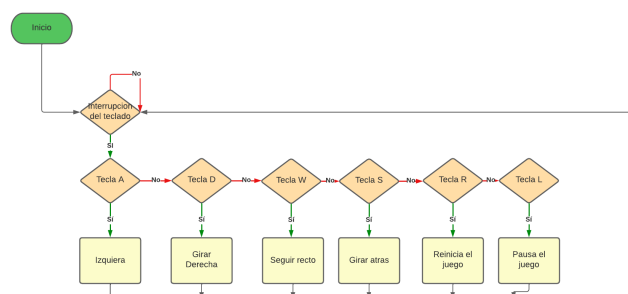


Figura 1. Diagrama de flujo moviento del jugador.

V. INSTRUCCIONES DE UTILIZACIÓN DEL PROYECTO

Seguidamente se detallarán una serie de pasos necesarios para poder jugar Mobile Maze Booteable.

Inicialmente, debe tener en sus posesión una llave maya que contenga el juego, insertarla en una computadora apagada, y encender, el juego deberá iniciar automáticamente.

Una vez iniciado el juego recibirá un mensaje de bienvenida como el mostrado a continuación en la figura 2.



Figura 2. Pantalla de Bienvenida.

Como indica dicha pantalla, se deberá presionar la tecla 'Enter' para ingresar al juego en el nivel 1. Obteniendo así lo visualizado en la figura 3, lo cual muestra la pantalla a completar en primera instancia, con sus debidos obstáculos.

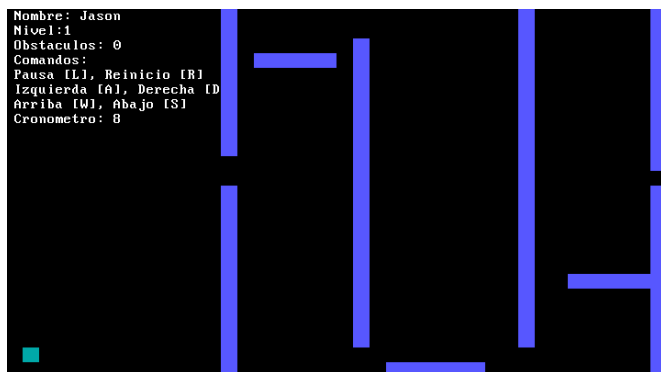


Figura 3. Primer nivel.

Para empezar a jugar, el usuario tiene en la esquina superior izquierda una serie de teclas que debe presionar para realizar diversas funcionalidades según se requiera, entre ellas se encuentra la lista de teclas para movimiento, arriba, abajo, izquierda, derecha (W, A, S, D, respectivamente), pausa y reinicio (L y R, respectivamente), así como también información general sobre el juego. Esto mencionado anteriormente puede ser observado en la figura 4.

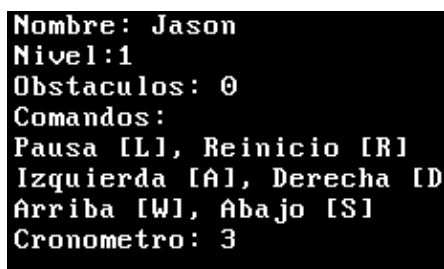


Figura 4. Instrucciones de juego.

Para completar el nivel 1, se deberá encajar la nave por el espacio correspondiente en el extremo derecho de la pantalla

de juego, cada vez que el usuario colisione, será devuelto al inicio del juego, en caso de que el contador llegue a cero, de igual manera, se reiniciará al nivel uno. Una vez logrado esto, el usuario se encontrará con el reto de completar el siguiente y último nivel, el cual puede ser observado en la figura 5, este nivel al igual que el anterior, contará con instrucciones y detalles de juego, idéntico a lo mostrado en la figura 4.

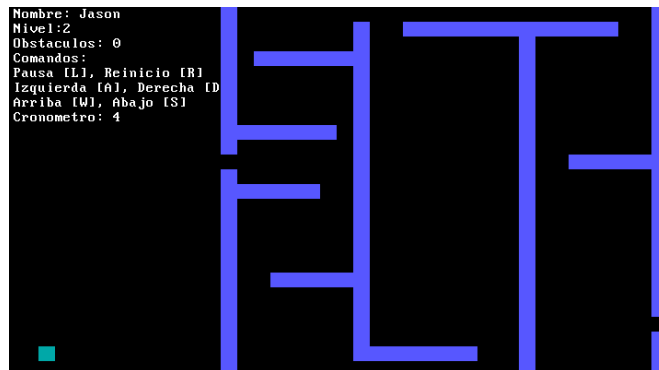


Figura 5. Segundo nivel.

Una vez logres completar este nivel, felicidades, has completado el contenido de Mobile Maze Bootable! (Ver figura 6)



Figura 6. Pantalla de victoria.

VI. TABLA DE ACTIVIDADES POR ESTUDIANTE

La tabla que contiene los detalles de las actividades por estudiante se presente en la sección IX como anexo.

VII. CONCLUSIONES

En el proceso de realización del trabajo fue posible comprobar la importancia de las diversas herramientas de trabajo y tecnologías que están a disposición para el implementación de este tipo de proyectos, ya que apoyan y facilitan las labores de los desarrolladores.

Adicionalmente, se reforzaron conceptos relacionados con el análisis de requerimientos, diseño de sistemas, y desarrollo de programas en lenguaje ensamblador. Específicamente, se repasó la implementación en x86, a través de la implementación de un juego con diversas funcionalidades.

De igual manera, se reforzó el tema visto en el curso sobre el *booteo* las computadoras y los espacios de memoria donde inicia la ejecución de las mismas. Esto mediante la implementación de un *bootloader* funcional en x86, el cual se utilizó para cargar el sector de memoria donde se encuentra el juego, en lugar del sistema operativo.

Finalmente, el proyecto fue de gran provecho para comprender y aplicar el tema del curso relacionado con llamadas al sistema. Esto debido a que durante la ejecución del mismo, es necesario realizar interrupciones por medio de software para que el *kernel* tome control y pueda ejecutar ciertas funciones.

VIII. SUGERENCIAS Y RECOMENDACIONES

El equipo reflexionó sobre todo lo realizado, y se considera que en base a nuestras experiencias, podríamos aportar cierto conjunto de sugerencias y recomendaciones, las cuales serían muy útiles a futuro, para que interese y desee entender y/o desarrollar un juego de estas características:

- Se recomienda mantener la calma en todo momento, desarrollar un juego de estas características no es una labor sencilla que pueda ser realizada de manera correcta en un solo día. Será normal encontrar inconvenientes en el camino, de modo que se aconseja no entrar en estrés.
- Como se mencionó anteriormente, el desarrollo de un juego de este estilo, requiere concentración, compromiso, y como uno de los puntos principales, entendimiento, es por esto que se recomienda estudiar a profundidad todos aquellos aspectos a abarcar en el desarrollo, conceptos teóricos y de implementación de Bootloader, así como también conocimiento en general del ensamblador a utilizar, esto librará al usuario de muchos inconvenientes, aunque no todos, por supuesto.
- Si se trabajará en equipo, se recomienda mantener actividades, reglas y medidas que contribuyan a la cohesión del equipo, al desarrollo constante y al interés por el trabajo individual de cada uno, esto contribuirá al entendimiento general de la aplicación y su desarrollo.
- El manejo de memoria es esencial, se debe entender en todo momento de qué forma está siendo utilizada la memoria. Esto es vital, ya que el Bootloader requiere un entendimiento de memoria significativo.
- Si se cuenta con un tiempo límite, se recomienda suministrar el tiempo de forma inteligente, es decir, no delegar tareas a últimas instancias, pues este trabajo a como es de demandante, es de estresante, lo cual requiere que el desarrollador pueda permitirse descansar cuando lo considere necesario.

IX. ANEXO: TABLA DE ACTIVIDADES POR ESTUDIANTE

A continuación se presenta la distribución de las tareas identificadas en la tarea por estudiante, así como datos como su fecha límite establecida, horas dedicadas y estado de completitud.

ID	Descripción	Dependencia	Fecha límite (estimada)	Responsable(s)	Horas de trabajo	Estado
0	Crear un demo del programa boot loader escrito en asm x86.	N/A	28/02	Agustín	5	Completado
1	Programar el bootloader oficial que se usará para la tarea.	0	01/03	Agustín	6	Completado
2	Ejecutar el bootloader en un USB drive durante el booteo de la máquina.	1	02/03	Agustín Juan	6	Completado
3	Diseñar una interfaz para desarrollar los aspectos de jugabilidad.	N/A	27/02	Juan	6	Completado
4	La primera vez que se inicia el juego se debe dar un mensaje de bienvenida.	3	01/03	Julián	3	Completado
5	Programar la funcionalidad de poder mover un elemento en pantalla con el teclado.	3	02/03	Kevin	5	Completado
6	Programar la aparición de obstáculos en pantalla de manera estática.	3	02/03	Kevin	6	Completado
7	Programar la funcionalidad de que algunos obstáculos se mueven hacia arriba o abajo de manera aleatoria.	6	04/03	Kevin	-	Descartado
8	Crear un “túnel” en la parte derecha de la pantalla del juego, donde la nave tenga que pasar para poder avanzar al siguiente nivel.	3	04/03	Kevin Juan	1	Completado
9	Programar la lógica que detecta cuando la nave logra pasar el túnel y redirige al usuario al siguiente nivel.	8	04/03	Juan Kevin	2	Completado
10	Programar la selección de dificultad antes de inicializar el juego	4		Julián *	-	Descartado
11	Recolección y muestra de estadísticas: Cuántos obstáculos se han superado, nivel de dificultad actual y comandos. Implementar una sección en la parte superior de la pantalla del juego donde se muestren estas estadísticas.	3 6 10	04/03	Julián Kevin Juan	1.5	Completado
12	Programar la lógica para reiniciar los niveles.	5 6	05/03	Agustín	2	Completado
13	Programar la animación de victoria.	N/A	06/03	Julián	5	Completado
14	Pausar el juego en cualquier instante que el usuario se desee	7	05/03	Juan	4	Completado
15	Programar la funcionalidad de que se pueda reiniciar el juego cuando se finalizan los niveles, o cuando el usuario presione una tecla.	5 6 12	06/03	Agustín	3	Completado
16	Salir del juego al final o en cualquier instante.	5 6	05/03	Juan	3.5	Completado
17	Aumentar la propiedad de velocidad cuando el usuario haya llegado al final del primer nivel	9	05/03	Juan	-	Descartado
18	Desplegar mensaje de finalización del juego. La pantalla de finalización debe tener también las opciones para salir o reiniciar	9 13 17	07/03	Julián Agustín	1	Completado
19	Reiniciar el juego en el momento que haya una colisión entre el usuario y un obstáculo	5 6 12	07/03	Kevin Agustín	2	Completado

Cuadro I

TABLA DE ACTIVIDADES POR ESTUDIANTE