



Universidad Nacional Autónoma de México
Facultad de Ingeniería
División de Ingeniería Eléctrica



Proyecto: Aterrizaje Lunar

Nombre: Flores Saavedra José Gerardo

Semestre: 2021-1

Maestro: M. I. Marco Antonio Martínez Quintana

Materia: Fundamentos de Programación

Fecha de elaboración: octubre 2020 – enero 2021

Resumen

En este documento se encuentra toda la información sobre el proyecto final que se llevó a cabo durante el semestre 2021-1 de la materia de Fundamentos de programación.

De inicio vamos a dar una introducción del tema que aborda el proyecto

Después, veremos sobre el desarrollo del proyecto, lugar donde se encuentran el algoritmo, el diagrama de flujo, pseudocódigo y código fuente del proyecto que se fue realizando según la planeación y del diagrama de Gantt.

Luego, los resultados del proyecto donde se verán algunas capturas de pantallas del funcionamiento del proyecto, además de las comparaciones sobre lo que se tenía en previsto de lo que se iba a necesitar a inicios de la realización del proyecto y ahora que se debe de entregar un avance, una comparación en los aspectos económicos, de conocimientos y de recursos tecnológicos para poder trabajar.

Incluso, en este apartado podemos encontrar un enlace a un canal de YouTube en el cual se encuentra un video explicando parte de este documento y mostrando el programa en ejecución, y otro enlace a un repositorio de GitHub en el que se encuentra este archivo, el algoritmo, diagrama de flujo, pseudocódigo y código fuente por separados, haciendo mención de que cualquier persona pueda descargar el código fuente y pueda usarlo libremente.

Posteriormente, se encuentran las conclusiones en las cuales se habla un poco de la relación con GNU, FSF y IEEE, junto con su impacto en la industria 4.0 y un comentario personal del desarrollador.

Finalmente, se encontrarán las referencias de donde se baso este proyecto, incluyendo algunas fuentes que fueron consultadas y recuperación de temas dados por otros profesores tenidos antes de llegar a esta etapa para poder tener un amplio conocimiento para la construcción de todo este proyecto final.

Introducción

El tema que aborda este proyecto son los videojuegos ya que es algo que siempre me ha gustado desde pequeño y siempre he querido trabajar programando esto desde el momento en el que me introdujeron al mundo de la programación.

También me mantengo informado sobre todo lo más reciente de este mundo, aunque tengo que aprender las nuevas tecnologías que usan tanto de software como de hardware debido a que no solo basta con conocer un solo tipo de lenguaje de programación o no es suficiente con quedarse con las tecnologías de ahora porque las tecnologías y los componentes de los videojuegos siempre andan en constante cambio para brindar una mejor experiencia para el jugador.

Siempre que hablo de los videojuegos menciono que estoy a favor de que los videojuegos son muy buenas herramientas para aprender, desarrollar habilidades o actualmente acaban de decir que los van a empezar a usar como terapia en niños con cáncer ya que reduce el dolor y ayuda en su recuperación. Además, es bueno que se empiece a quitar los estereotipos sobre que los videojuegos hacen violentos a las personas, es una obligación respetar la clasificación de cada uno, pero está comprobado científicamente que los videojuegos no provocan algún cambio en el comportamiento de las personas.

La violencia en nuestra sociedad es un tema que más bien comprende la educación y los valores aprendidos en casa, y que entraría dentro del ámbito de la psicología, y que sería digno de un gran estudio para entender nuestra sociedad en estos días. En la actualidad hay estudios sobre los beneficios que tiene el jugar con videojuegos, entre ellos: mejora la atención, fortalece la memoria, previene el envejecimiento cerebral, entre otros.

Además, podemos ver la trascendencia de los videojuegos que ya son considerados un deporte y hasta van a ser parte de los Juegos Olímpicos, o como dije anteriormente, van a ser parte de tratamientos médicos.

También en este momento que vivimos de encierro por la pandemia, al auge de los videojuegos es innegable, pues para los adolescentes confinados en sus casas este es un medio de escape y del lugar de socializar, al conectarse con otras personas que son afines en temas de interés.

Así que por el lado que veamos y analicemos, a la tendencia de los videojuegos sigue en crecimiento y con un gran futuro, además de que puede ser una herramienta, que bien encaminada puede traer beneficios sociales, y dejar de lado un poco el tema de la violencia en esta industria, para abordar más temas como el baile, el deporte, la historia que también pueden resultar interesantes para diferentes sectores de la sociedad,

Desarrollo del proyecto

- Descripción general del proyecto

Mi proyecto es un videojuego retro llamado aterrizaje lunar programado en lenguaje c ++ junto con el programa de Allegro para poder mostrar gráficamente las instrucciones que se le están dando a la computadora. Al inicio, iba a ser programado en Dev-cpp, pero por complicaciones con la instalación de la librería Allegro, se decidió usar CodeBlock para ello.

- Algoritmo

Sin devolver:

- Pintar nave
- Mover nave

Instrucciones principales:

- Iniciar Allegro
- Hacer que el teclado se el dispositivo de entrada
- Definir 32 pixeles
- Definir el tamaño de la pantalla a mostrar
- Crear un lugar donde se vayan a proyectar las imágenes
- Crear cuatro variables flotantes, dos para la nave y dos para la caída libre
- Darle el valor de la nave de 100
- Hacer una repetición mientras no se presione la tecla escape
 - Darle color la pantalla
 - Darle color a la nave
 - Mostrar la pantalla junto con la nave
 - Que la nave se mueva conforme la aceleración de la gravedad
 - Esperar 20 milisegundos
- Fin de la función principal
- Terminar Allegro

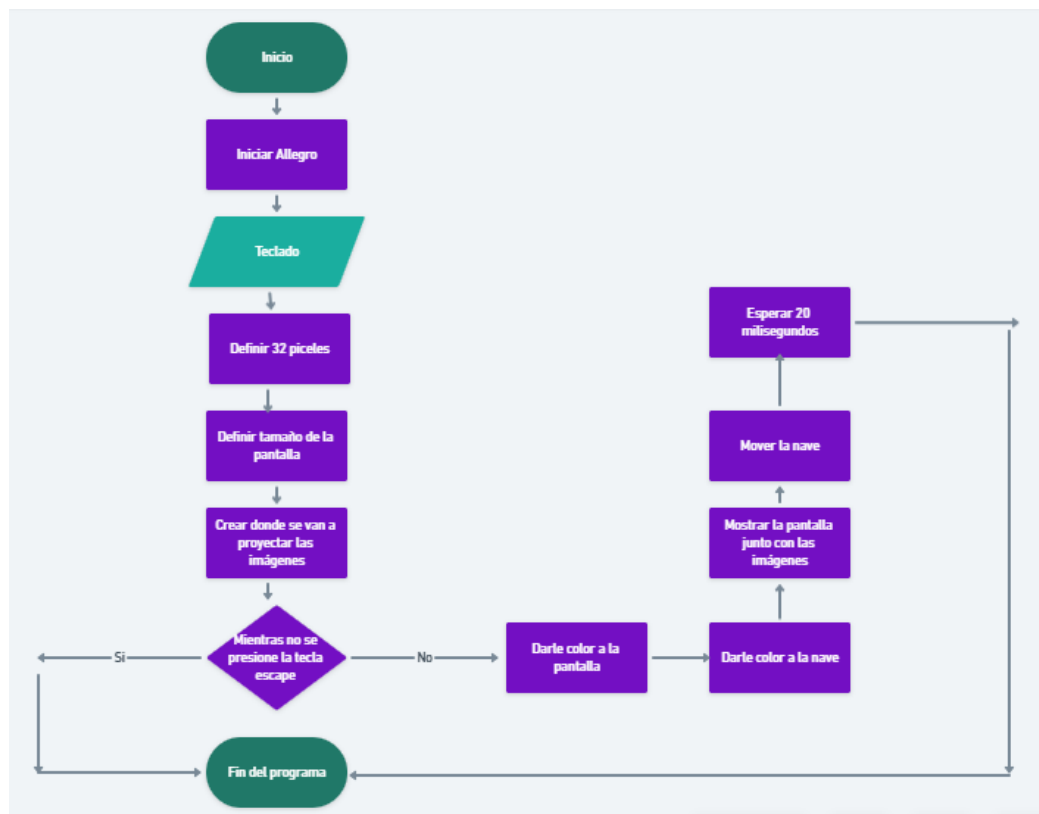
Sin devolver (1):

- Pintar la nave
 - Definir un arreglo flotante con 22 espacios
- Estructura de repetición for (Declarar un entero que sea menor o igual a 14 e irle sumando 2)
 - Definir las líneas que armaran a la nave
 - Condicional if (La variable anteriormente declarada sea igual a 14)
 - Definir el resto de las líneas de la nave

Sin devolver (2):

- Mover nave
- Declarar dos variables flotantes para la constante de la gravedad
- Darle el valor de una 0 y la otra de 0.05
- Sumarles a las variables de la velocidad de caída las variables de la constante de la gravedad
- Sumarles los centros de la nave las variables de caída libre

- Diagrama de flujo



- Pseudocódigo

FUNCIÓN VOID pintar nave
FUNCIÓN VOID mover nave

INICIO

INICIAR Allegro
LLAMAR Teclado
DEFINE formato de 32 pìxeles
DEFINE tamaño de pantalla

CREAR BITMAP y buffer como puntero

FLOTANTE CentroX, CentroY

FLOTANTE VelocidadX, VelocidadY

VALOR CentroX = 100

VALOR CentroY = 100

Repetir MIENTRAS NO se presiona tecla escape:

 PINTAR buffer

 PINTAR nave

 ABRIR pantalla

 MOVER nave

 ESPERAR 20 milisegundos

FIN

TERMINAR Allegro

FUNCIÓN VOID pintar nave

 FLOTANTE ARREGLO nav de 22 espacios

 REPETIR ENTERO i igual 0; i menor o igual 14; sumar 2 a i

 LINEA

 SI i igual 14

 LINEA

FUNCIÓN VOID mover nave

 FLOTANTE aceleración en x, aceleración en y

 VALOR aceleración en x igual 0, aceleración en y igual 0.05

 Velocidad x más aceleración en x

 Velocidad y más aceleración en y

 Centro en x más aceleración en x

 Centro en y más aceleración en y

- Código fuente

```

1  #include <iostream>
2  #include <allegro.h>
3
4  using namespace std;
5
6  void pintar_nave(float cx,float cy, BITMAP *buffer); //Prototipo de la función
7  void mover_nave(float &cx, float &cy, float &vx, float &vy); //Simulación de la caída libre
8
9  int main(){
10     allegro_init(); //Inicia el entorno de Allegro
11     install_keyboard(); //Hace el llamado a la función del teclado, dispositivo de entrada
12     set_color_depth(32); //Formato de 32 pixeles o 32 bits
13     set_gfx_mode(GFX_AUTODETECT_WINDOWED,740,500,0,0); //Define el tamaño de la pantalla donde se va a abrir el programa
14     BITMAP *buffer = create_bitmap(740,500); //Los BITMAPS son imágenes que se mostrarán en la pantalla
15     //El buffer es donde se van a estar agregando todas las imágenes del juego
16     float cx,cy; //Variables para dibujar la nave, cx y cy son el centro de la nave
17     float vx,vy; //Variables para la velocidad de caída
18
19     cx=100; cy=100;
20
21     while(!key[KEY_ESC]){ //Condicionamos que mientras no se presiones la tecla ESC, el juego seguira corriendo, como teclas de seguridad
22         clear_to_color(buffer,0x000000); //Hacemos que el buffer tenga un determinado color
23         pintar_nave(cx,cy,buffer);
24         blit(buffer,screen,0,0,0,740,500); //Imprime el buffer en coordenadas determinadas junto con las coordenadas en la pantalla
25         mover_nave(cx,cy,vx,vy);
26
27         rest(20); //Hace que el programa no ejecute todo rápidamente y espere, al llegar a este punto, 20 ms
28     }
29 }
30
31 return 0;
32 }
33 END_OF_MAIN(); //Especifica el fin de la función main
34
35 void pintar_nave(float cx,float cy, BITMAP *buffer){ //Función para poder imprimir el diseño de la nave en el buffer
36     float nav[22] =
37     {cx-20,cy+20, cx-20,cy+10, cx-20,cy, cx-10,cy-10, cx,cy-15, cx+10,cy-10,
38     cx+10,cy, cx+20,cy+10, cx+20,cy+20, cx-10,cy, cx+10,cy};
39
40     for(int i=0;i<=14;i+=2){
41         line(buffer,nav[i],nav[i+1],nav[i+2],nav[i+3],0x1854E5);
42         if(i==14) line(buffer,nav[i+4],nav[i+5],nav[i+6],nav[i+7],0x1854E5);
43     }
44 }
45 void mover_nave(float &cx, float &cy, float &vx, float &vy){ //Simulación de la caída libre
46     float ax, ay;
47     ax = 0;
48     ay = 0.05; //Aceleración constante de la gravedad
49
50     vx += ax;
51     vy += ay; //Aplicación de la gravedad sobre la nave, cambios de la velocidad
52
53     cx += vx;
54     cy += vy; //Cambio de posición de la nave con la gravedad
55 }
56

```

Resultados del proyecto

- Capturas de pantalla del funcionamiento del proyecto

Finalmente, no pude agregar ninguna captura de pantalla debido a que estaba trabajando normal, había hecho pruebas con Allegro antes de empezar el proyecto y todo funcionaba bien, se me habría dos pantallas, una de Allegro y otro la normal de siempre del compilador del sistema, había logrado instalar bien la librería de Allegro, pero hace dos días antes de la entrega, sigo agregando líneas de código y cuando quiero compilar me aparece que la librería de Allegro tenía que ser definida. Pero si puedo mostrar el erro que me da la librería:

Logs & others

```

Code::Blocks X Search results X Cccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X
File Line Message
obj\Debug\main.o
C:\Users\JGFS... 10 undefined reference to `_install_allegro_version_check'
C:\Users\JGFS... 11 undefined reference to `install_keyboard'
C:\Users\JGFS... 12 undefined reference to `set_color_depth'
C:\Users\JGFS... 13 undefined reference to `set_gfx_mode'
C:\Users\JGFS... 14 undefined reference to `create_bitmap'
C:\Users\JGFS... 21 undefined reference to `__imp_key'
C:\Users\JGFS... 24 undefined reference to `__imp_screen'
C:\Users\JGFS... 24 undefined reference to `blit'
C:\Users\JGFS... 27 undefined reference to `rest'
obj\Debug\main.o
C:\Users\JGFS... 33 In function `WinMain':
undefined reference to `_WinMain'
error: ld returned 1 exit status
=== Build failed: 11 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===

```

- Tablas comparativas de recursos informáticos (octubre 2020 – enero 2021)

Hardware	
Octubre 2020	Enero 2021
Laptop	Laptop
Mouse	Mouse

Software	
Octubre 2020	Enero 2021
Dev++	Dev++
Compilador MinGW	Compilador MinGW
	Allegro

- Tablas comparativas de costos asociados al proyecto (octubre 2020 – enero 2021)

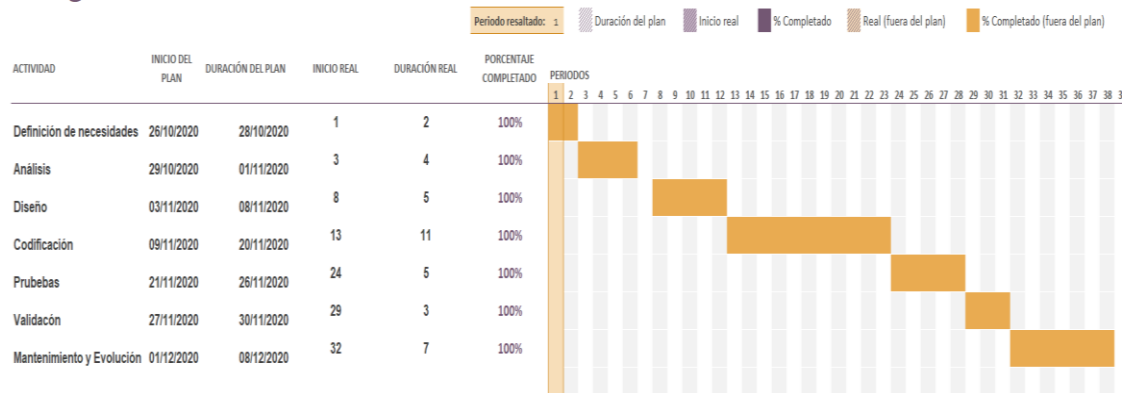
Costos

Octubre 2020	Enero 2021
20,000\$	35,000\$

- Comparación de diagramas de Gantt (octubre 2020 – enero 2021)

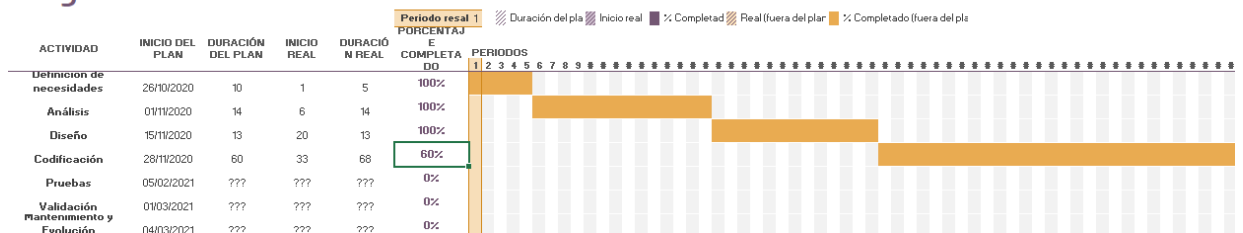
Octubre 2020

Diagrama de Gantt



Enero 2021

Diagrama de Gantt



- Enlace al canal de YouTube
<https://www.youtube.com/channel/UCayqsL9RkPfnWyifJ2XFOcw>
- Enlace al repositorio de GitHub
<https://github.com/gerard-flores-99/Fundamentos-de-Programacion-Evidencias/tree/main/Proyecto%20Final>

Conclusiones

Este proyecto se relaciona con GNU y FSF porque cualquier persona va a poder acceder al código fuente sin la necesidad de pagar alguna cantidad de dinero. Se habló de dinero en las comparaciones, pero era para el caso de prestar o solicitar el servicio para la creación del proyecto. Cualquier persona va a poder modificar el código fuente a su gusto pese a que no está completo, pero se puede ver como un libro con un final abierto, cada persona ideará su final para este proyecto.

Con la IEEE y la Industria 4.0 no lo veo tan relacionado ya que en estos son la innovación tecnológica con los últimos conocimientos adquiridos donde se requiere mayor equipo para poder desarrollar un programa, una IA o lo que sea para realmente sea considerado una innovación. Donde si lo puedo considerar dentro de la Industria 4.0, es en el uso de repositorios en línea y el impulso de que los demás sean creativos al tener el código fuente libre para su uso y lo terminen como ellos quieran.

La verdad me gustó empezar a desarrollar un programa sobre lo que siempre me ha interesado. Al inicio, no tenía definido qué hacer de proyecto final, pero buscando un poco pude encontrar algunos videos en los cuales puedo desarrollar videojuegos retros para practicar para tener experiencia y puede que me atreva a desarrollar mi propio videojuego para sacarlo al mercado. Lo que no me gustó, fue que la librería de Allegro se me haya arruinado al final cuando ya tenía gran avance del código fuente y por falta de tiempo no me va a dar para poder estar revisando atentamente cuál o dónde está el error para que vuelva a funcionar como era anteriormente.

Referencias

- Aterrizaje Lunar – Desarrollo de Juego Retro en C++. Deividcoptero Programación. 3 de diciembre de 2019. Consultado el día 16 de noviembre de 2020. https://www.youtube.com/watch?v=-cY9AubE6_k&list=PL6hPvfzEEMDZOHeVaomaRPhWK8KJonAaY&index=1
- Cruz, G. y Martínez, M. (s/f). Soluciones MyL. Desarrollo de Software mediante Lenguaje C. http://solucionesmyl.com/cursos/lenguaje_c/home.php
- Solano, J., García, E., Sandoval, L., Nakayama, A., Ricci, T. y Castañeda, M. (2018). Manual de prácticas del laboratorio de Fundamentos de programación. Recuperado el día 28 de septiembre de 2020, en: <http://lcp02.fi-b.unam.mx/>
- Solórzano, J., Millán, A., Solórzano, C. y Sánchez, A. (2016). Fundamentos de Computación Panorama Histórico y Programación. Facultad de Ingeniería, UNAM.
- Villegas, Rebeca. “Curso de Informática Aplicada a la Ciencia y a la Industria”. Informática Aplicada a la Ciencia y a la Industria. Escuela Nacional Preparatoria No. 7, UNAM. 12 de agosto de 2019 – 12 de marzo de 2020.