



**Universidad Nacional Autónoma de México**

Facultad de Ingeniería

Ingeniería en Computación



Computación Gráfica e Interacción Humano Computadora

Proyecto Final

**“Manual Técnico”**

Alumnos

González Castro Juan Fernando

López Cruz Luis Enrique

Osorio Alvarado Jorge Adalberto

Grupo 3

6 de enero del 2023

Semestre 2023-1

## INDICE

## Pág.

1. Información de animaciones (documentación técnica)	2
2. Información de cronograma de actividades	16
3. Estudio técnico para determinar el costo de la aplicación	11
4. Descripciones de actividades en Ingles	12
5. Manual general de controles del proyecto	13

## 1-. Información de Animación (Información Técnica)

### Documentación técnica de la animación del Automóvil.

Esta animación se hizo mediante el método de animación por estados o también conocida como en tiempo real.

```
llanta -= 3.5f;
//Vehículo
if (animacion) {
    if (movimiento == 0)
        movAuto_x += 1.5f;
    if (movAuto_x > 190.0f) {
        rotCarro = -90.0f;
        movimiento = 1;
    }

    if (movimiento == 1) {
        movAuto_z += 1.5f;
        if (movAuto_z > 260.0f) {
            rotCarro = -180.0f;
            movimiento = 2;
        }
    }

    if (movimiento == 2) {
        movAuto_x -= 1.5f;
        if (movAuto_x < -200.5f) {
            rotCarro = -270.0f;
            movimiento = 3;
        }
    }

    if (movimiento == 3) {
        movAuto_z -= 1.5f;
        if (movAuto_z < -140.0f) {
            rotCarro = -360.0f;
            movimiento = 4;
        }
    }

    if (movimiento == 4) {
        movimiento = 0;
    }
}
```

Para realizar esto se tuvo primero que definir las variables a utilizar que darían movimiento a nuestros modelos en este caso que serían dos.

Las llantas son un modelo independiente que conforman al carro y tienen ellas mismas un movimiento en particular, simulando que ruedan sobre su mismo eje, dando así la sensación de que el automóvil está andando.

Por otro lado, tenemos nuestro modelo base que sería la Carrocería (cuerpo del automóvil), este solo tendrá movimientos de translación los cuales se condicionan como se ven en la imagen del código.



El movimiento que realiza nuestro automóvil por lo tanto sigue la ruta de la carretera que se modelo para dicho objeto, por lo que al ejecutarse se podrá ver cómo es que recorre ese circuito.

```
model = glm::rotate(glm::mat4(1.0f), glm::radians(rotacion), glm::vec3(0.0f, 1.0f, 0.0f)); //direccion del carro como se haran la ruta
model = glm::translate(model, glm::vec3(movAuto_x, movAuto_y, movAuto_z)); //Pone al carro en su posicion inicial
tmp = model = glm::rotate(model, glm::radians(rotCarro), glm::vec3(0.0f, 5.0f, 0.0f)); //la rotacion de como el carro va andar
model = glm::scale(model, glm::vec3(1.0f, 1.0f, 1.0f));
staticShader.setMat4("model", model);
Carro.Draw(staticShader);

model = glm::translate(tmp, glm::vec3(18.0f, 3.0f, -11.5f));
model = glm::rotate(model, glm::radians(llanta), glm::vec3(0.0f, 0.0f, 1.0f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::scale(model, glm::vec3(6.7f, 6.7f, 6.7f));
staticShader.setMat4("model", model);
Llanta.Draw(staticShader); //Izq delantera
```

Como nota adicional, sabemos que en este tipo de animaciones debemos tener definida una jerarquización la cual se define como se muestra en la imagen anterior, definiendo el modelo base que es la carrocería a las demás llantas para que así, cuando la carrocería se mueva las llantas se muevan junto con ella.

### ¿COMO SE EJECUTA LA ANIMACION?

Para que la animación se ejecute en el espacio **SE DEBE PRESIONAR LA TECLA "1"** del teclado, al igual que si se quiere pausar debe presionar la misma tecla del teclado.

Ahora para REINICIAR **SE DEBE PRECIONAR LA TECLA "R"**

### Documentación técnica de la animación del pingüino saludando

Esta animación fue hecha por keyframes y para su ejecución deberemos tener todo bien configurado, empezando porque nuestra animación recibe los valores de cada frame mediante un archivo txt, en este caso el nombre del archivo viene hardcodedo y se llama "Valores.txt" este archivo deberá de estar al mismo nivel de donde está el cpp para que funcione.

```
-65
0
170
0
50
0
-60
-45
-65
0
130
0
0
0
-60
60
```

#### De primeras esto no es necesario

Como nota de funcionamiento o un error que se podría dar es que al leerse cada fila del documento y guardarse en un String llamado "línea" hasta que encuentre un EOF puede darse el caso que se guarde el archivo con enter debajo de los valores y estos sean reconocidos y truee, por lo que se tendría que ver que el cursor de edición al lado del último número sin espacios.

La animación consiste en un recorrido saludando a los hábitats aledaños, mueve sus bracitos y sus pies por lo que todos estos movimientos se definieron previamente como ya mencionamos en nuestro archivo de texto que fue llamado “Valores.txt”.

Como vemos en la imagen de código, podemos mencionar también adicionalmente que la función que se utilizó para trabajar con este tipo de animaciones fue un poco complicada de entender, pero se logró utilizando conceptos básicos de extracción y redacción de archivos.

```
float my_points[10000];
int count = 0;

string nombreArchivo = "Chida.txt";
ifstream archivo(nombreArchivo.c_str());
string linea;
// Obtener línea de archivo, y almacenar contenido en "linea"
while (getline(archivo, linea)) {
    my_points[count] = stof(linea);
    cout << my_points[count] << " ";
    count++;
}

int j = 0;

for (int i = 0; i < FrameIndex; i++)
{
    KeyFrame[i].posXP = my_points[j++];
    KeyFrame[i].posYP = my_points[j++];
    KeyFrame[i].posZP = my_points[j++];
    KeyFrame[i].giroPingu = my_points[j++];
    KeyFrame[i].rotAletaIzq = my_points[j++];
    KeyFrame[i].rotAletaDer = my_points[j++];
    KeyFrame[i].MovBracitoIzq = my_points[j++];
    KeyFrame[i].MovBracitoDer = my_points[j++];
}
```



### ¿COMO SE EJECUTA LA ANIMACION?

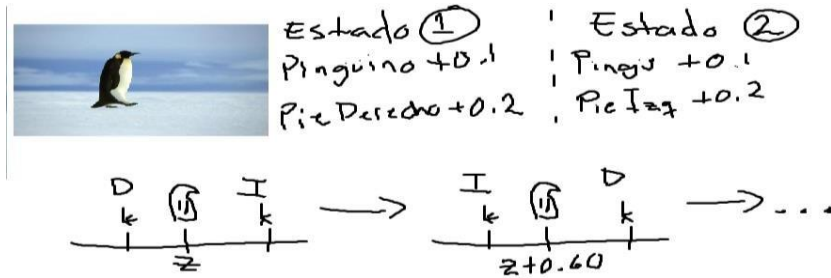
Para que la animación se ejecute en el espacio **SE DEBE PRESIONAR LA TECLA “9”** del teclado, al igual que si se quiere pausar debe presionar la misma tecla del teclado.

Ahora para REINICIAR **SE DEBE PRECIONAR LA TECLA “R”**

### Documentación técnica de la animación Pingüino.

Para la animación del pingüino se pensó en que diera un recorrido a lo largo de su escenario designado en el proyecto, se tuvo que pensar bien la lógica detrás de la caminata del pingüino debido a que con la poca experiencia que se tiene al separar los modelos por partes se deben de colocar de manera que sea fácil utilizarlos en el algoritmo, por eso se eligió que como posición inicial un pie estuviera delante de otro por 0.60 unidades de diferencia respecto al eje z.

Este algoritmo fue un poco más difícil que los otros por la naturaleza del movimiento y porque para ser eficientes nuestro algoritmo debe reducirse a 2 estados, uno en donde el pingüino da un paso derecho y avanza y otro en donde da un paso izquierdo y avanza, todo esto debe de estar ligado a una restricción en la coordenada z para que no salga de su espacio designado.



El problema al que se enfrentó al realizar este algoritmo fue una operación matemática que nos permitiera reutilizar estos dos estados, como no se puede usar la operación de residuo “%” debido a que trabajamos con números flotantes y dicho operador solo trabaja con números enteros se decidió usar un contador iniciado en 1 que dividiera el movimiento del pingüino entre 0.60 por el contador. De tal forma que con cada transición de estado la condición de esta operación siempre debe de ser mayor a 1 para cambiar de estado.

Es decir, en cada estado un paso es de 1.20 unidades y el desplazamiento del cuerpo aumenta 0.60 unidades por cada transición de estado de tal forma que la división de (movimiento pingüino / contador\*0.60) siempre deba de dar valores mayores a 1.0 para poder seguir caminando, si esto sucede el contador aumentara en uno para seguir satisfaciendo la operación.

Con esta misma lógica se realizaron los demás estados que es cuando sube la rampa, cuando la baja y cuando regresa a su posición original, solo que para el estado en donde sube la rampa y la baja se utilizó otro contador de tal modo que se evitaran comportamientos inapropiados al reiniciar la animación.



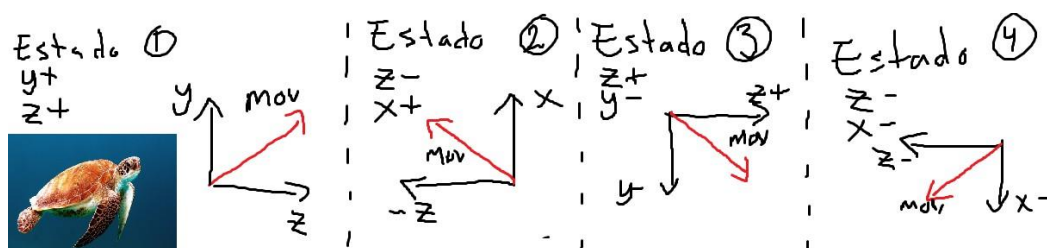
Para cada estado “principal” por así llamarlo, del pingüino, se necesitan dos estados “secundarios” en donde cada uno de estos dos estados corresponde a una pata del pingüino, haciendo un total de 8 estados que se repiten de manera cíclica, además de eso, el pingüino también mueve ligeramente su cuello, añadiéndole otros 2 estados extras a esta sencilla animación.

### ¿COMO SE EJECUTA LA ANIMACION?

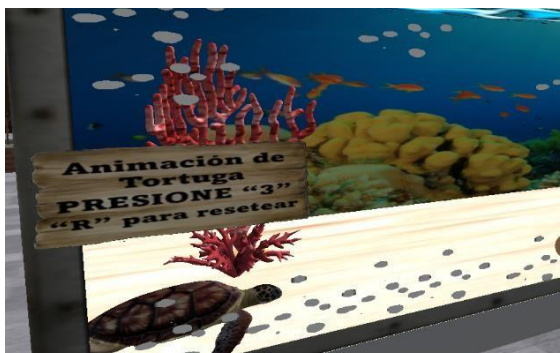
La tecla utilizada para activar dicha animación **SE DEBE PRESIONAR EL NÚMERO “2”**, si se quiere reiniciar la animación **SE DEBE APRETAR LA TECLA “R”**, inmediatamente volverá a su lugar original y podrá apretarse nuevamente el numero “2” para observar la animación nuevamente.

### Documentación técnica de la animación Tortuga

La lógica detrás de la animación de la tortuga es relativamente sencilla, se pensó en que el modelo de la tortuga realizara un recorrido a través del estanque aplicando traslaciones en los ejes x, y & z para que simulara que está nadando dentro de su estanque, el método más sencillo para poder implementar esta animación es el método de animación en tiempo real o por estados y por eso fue por lo que se eligió realizarlo así.



Para conocer las dimensiones del estanque en donde iba a trasladarse la tortuga se introdujo el modelo del estanque designado para ese modelo a 3ds Max para ver las coordenadas aproximadas del interior del estanque, una vez obtenidas dichas coordenadas se calcularon las pendientes y las rotaciones correspondientes a cada estado para que el movimiento de la tortuga fuera el adecuado. Además, para hacer aún más realista esta animación se recortó el modelo en 3ds Max separando el cuerpo y ambas aletas de la tortuga, a estas aletas también se les programó una animación la cual aplica rotación a estos modelos para simular que verdaderamente la tortuga está desplazándose con sus propias aletas. Estas animaciones se repiten en un ciclo y cuenta con una luz de tipo posicional que sigue la ruta de la tortuga, esta luz está programada para cambiar de color, dándole un toque especial a esta animación.





### ¿COMO SE EJECUTA LA ANIMACION?

La tecla utilizada para activar dicha animación **SE DEBE PRESIONAR EL NÚMERO “3”**, si se quiere reiniciar la animación **SE DEBE APRETAR LA TECLA “R”**, inmediatamente volverá a su lugar original y podrá apretarse nuevamente el numero “3” para observar la animación nuevamente.

#### Documentación técnica de la animación Huevo de tortuga

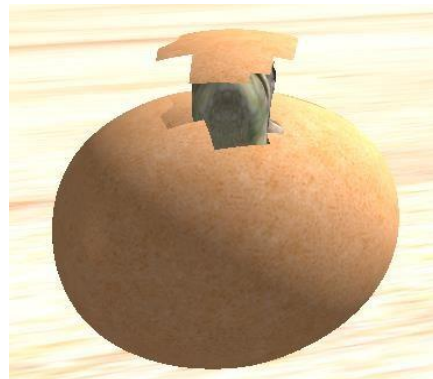
Esta animación simula a una tortuga rompiendo su huevo, mira a su alrededor y se vuelve a esconder, se realizó por animación en tiempo real.

El procedimiento fue partir el modelo de un huevo hueco para poder introducir dentro de él otro modelo de tortuga, este modelo de tortuga se recortó y únicamente se dejó la parte visible en la animación para así optimizar el objeto en cuestión, se trasladó y se colocó justo a la parte del huevo que se va a romper para que dé la sensación de que la tortuga realmente está rompiéndolo.

El diagrama de estados pensado para la creación del algoritmo es el siguiente:



La variable movimiento huevo esta asignada tanto al movimiento en el eje y de la tortuga y el huevo roto, de tal forma que se muevan al mismo ritmo sin tener que usar dos variables por separado, lo mismo aplica para la variable rotación huevo.



### ¿COMO SE EJECUTA LA ANIMACION?

La tecla utilizada para activar dicha animación **SE DEBE PRESIONAR EL NÚMERO “4”**, si se quiere reiniciar la animación **SE DEBE APRETAR LA TECLA “R”**, inmediatamente volverá a su lugar original y podrá apretarse nuevamente el numero “4” para observar la animación nuevamente.

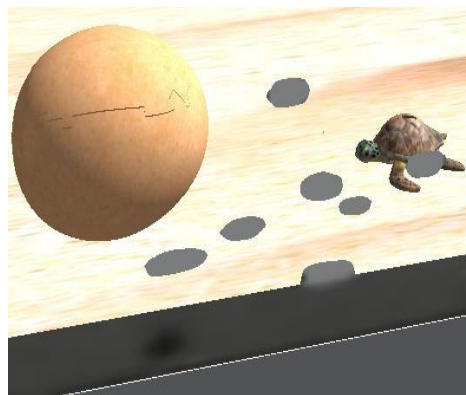
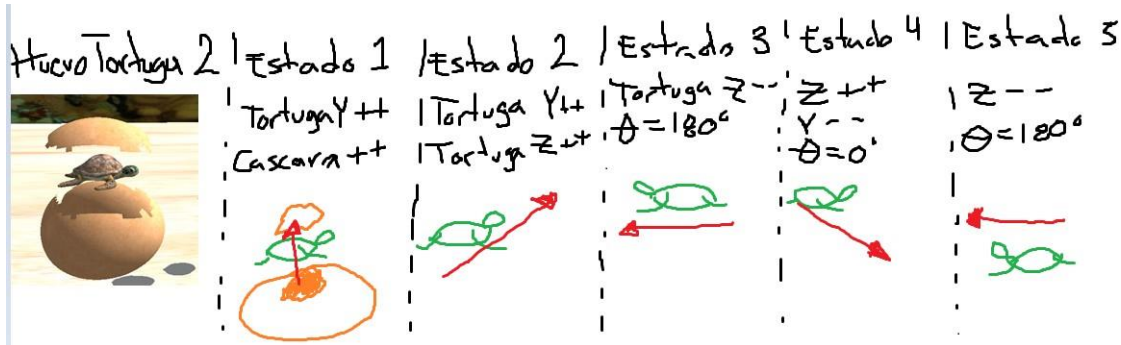
## Documentación técnica de la animación Huevo Tortuga 2

Esta animación simula a una tortuga que rompe su huevo y realiza su primer nado.

Para esta animación se pensó en combinar las dos animaciones de tortugas anteriores, por esto mismo se programó por estados.

El estado uno es cuando la tortuga y la cascara del huevo flotan en el estanque. Cuando se alcanza una altura definida se activa el estado dos, el cual inicia el recorrido de la tortuga bebé mientras la cascara del huevo cae nuevamente a su posición original. Para el estado 3 la tortuga nada en diagonal. El estado 4 es la tortuga desplazándose de manera horizontal. El estado 5 y último es la tortuga descendiendo en diagonal hasta llegar cerca de su huevo para al fin detener su camino programado.

En todo momento se está ejecutando otra pequeña animación que corresponde a las aletas de la tortuga bebé, así como en la animación *Tortuga* de este modo se le añade un poco mas de dinamismo al modelo.



## ¿COMO SE EJECUTA LA ANIMACION?

La tecla utilizada para activar dicha animación **SE DEBE PRESIONAR EL NÚMERO "5"**, si se quiere reiniciar la animación **SE DEBE APRETAR LA TECLA "R"**, inmediatamente volverá a su lugar original y podrá apretarse nuevamente el numero "5" para observar la animación nuevamente.



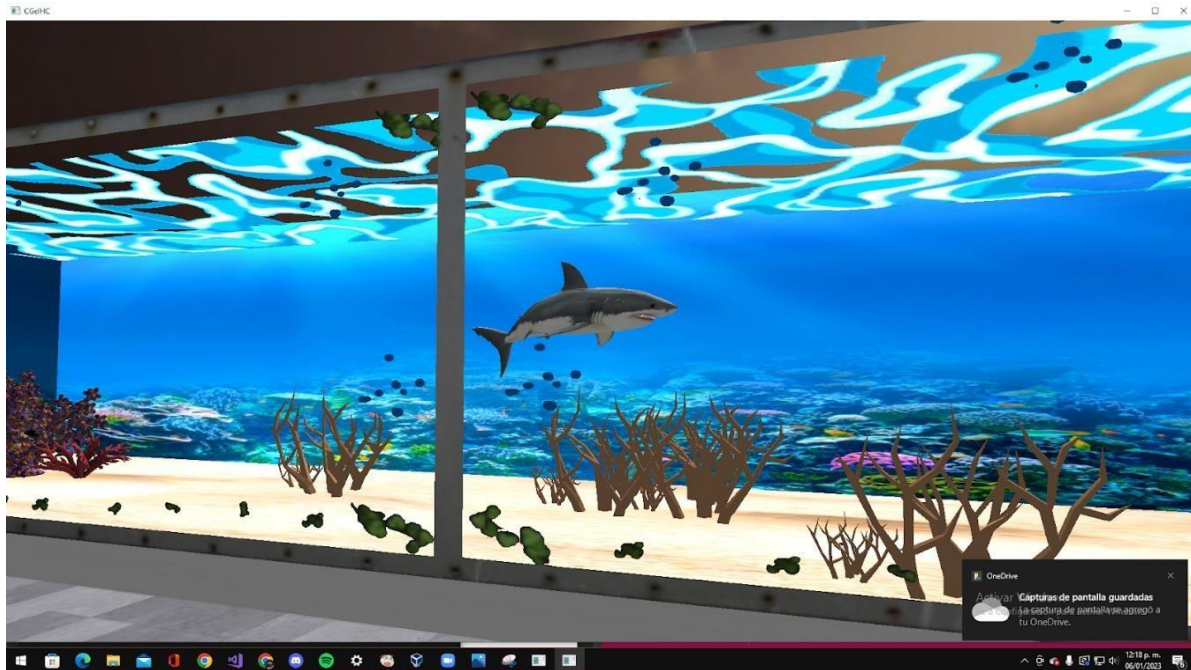
## Documentación técnica de la animación Tiburón

En esta animación se realiza un recorrido de lado a lado dentro de un estanque, el seguimiento de esto es mediante unas variables  $x, y, z$  para manipular al tiburón mediante traslaciones simulando que está nadando. Además, el modelo hace un aleteo en su movimiento en la aleta superior, laterales, cola, cuerpo trasero, esto para dar una sensación más realista de cómo se desplaza un tiburón.

La forma técnica en cómo se manejó esta animación fue el uso de una máquina de estados controlada por una variable `int` que representa el estado actual en el que está el tiburón y una vez terminado un estado, este pasará al siguiente hasta terminar. Se realizó de este modo debido a que es fácil visualizar el movimiento como estados.

El tratamiento del modelo fue partirlo en partes para que así cada una tuviera ese movimiento que se deseaba, un problema que tuvimos al momento de recortar y unir el modelo fue el usar un sistema de referencia para que el modelado jerárquico funcionara, una vez solucionado esto nos dimos cuenta que en algunas partes quedaba hueco al mover, esto se solucionó poniendo una masa que simulará ser la articulación en el caso del cuerpo trasero.





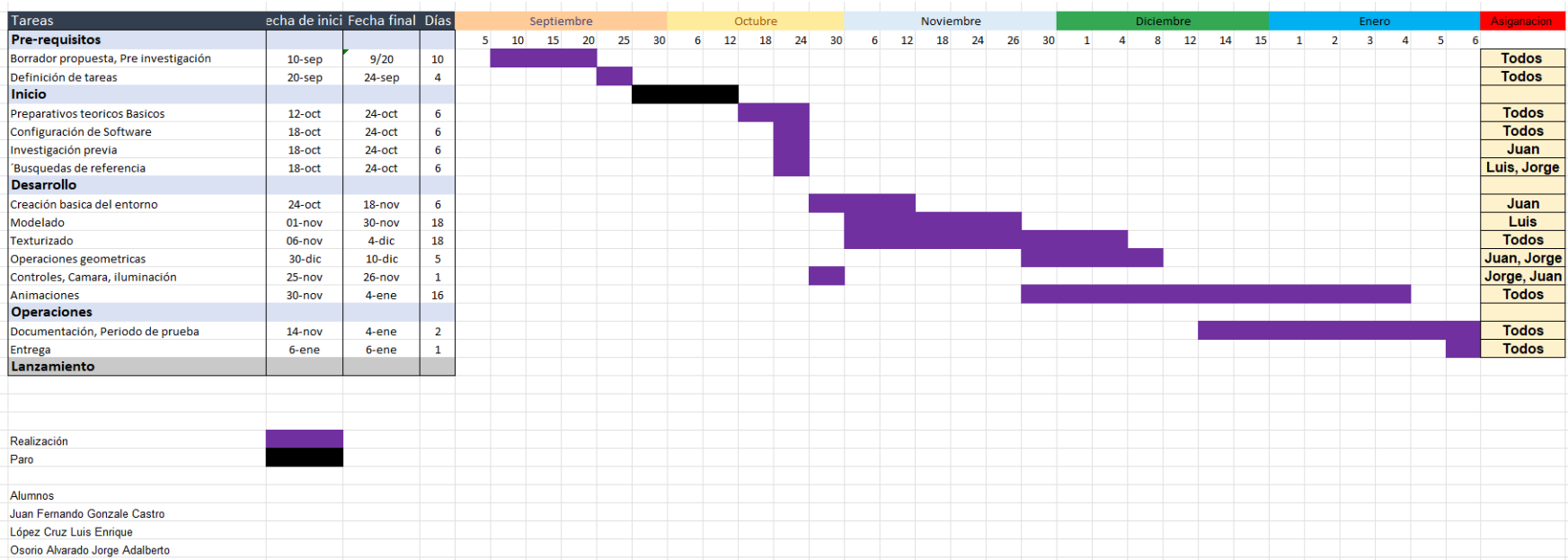
### ¿CÓMO SE EJECUTA LA ANIMACIÓN?

La tecla utilizada para activar dicha animación SE DEBE PRESIONAR EL NÚMERO "8", si se quiere reiniciar la animación SE DEBE APRETAR LA TECLA "R", inmediatamente volverá a su lugar original y podrá apretar nuevamente el número "8" para observar la animación si se desea.

## 2-. Información de cronograma de actividades

Para esta sección de cronograma de actividades se realizaron muchos cambios ya que nuestro cronograma de actividades inicial no fue planeado de esta manera sin embargo como entramos en paro de actividades en nuestra facultad, tuvimos que acoplarnos a un nuevo plan de desarrollo del proyecto el cual está definido como se muestra a continuación.

Cabe mencionar que las actividades que se muestran en el cronograma fueron en su mayoría hechas por los tres integrantes de equipo por lo que el proyecto se pudo realizar en un ambiente bueno y sin problemas entre los integrantes.



Desafortunadamente tuvimos problemas con el primer repositorio que creamos por lo que para la entrega de este proyecto decidimos crear uno nuevo donde solo estuviera el proyecto ya completo ( enlace que se encuentra en el manual) el cual se puede encontrar en la siguiente liga

<https://github.com/LuisG8/ProyectoFinalCompu.git>

**Sin embargo, nosotros ya habíamos trabajado en el repositorio anterior y es ahí donde se puede observar todo nuestro avance a lo largo del proyecto. Desde que se inicio hasta el ultimo cambio, por lo que anexamos la liga para que se puedan ver las actualizaciones y el trabajo que se realizo en el repositorio.** <https://github.com/JuanGonzalez1212/ProyectoFinal.git>

### 3-. Estudio técnico para determinar el costo de la aplicación.

Proyección de Costos y Ganancias.

#### Recursos.

Se cuenta con un total de 3 ingenieros para la realización del proyecto cada uno de nosotros cobramos a rededor de 30mil pesos mexicanos por proyecto con una duración de realización de 15 semanas. Las computadoras que se necesitan para el proyecto son de nuestra propiedad y cada una de ellas tiene un software listo con la licencia que se necesita.

<b>Gastos Adicionales.</b>	<b>Costo Neto</b>
Licencia completa para repositorio en GitHub	\$4,620 MXN
Licencia para AutoDesk (3dsMax, Maya)	\$9,900 MXN
Office	\$13,654 MXN
Visual Studio	\$0,0 MXN

Sin embargo, se planean tener reuniones en donde se acordarán los términos y adelantos del trabajo, así como la administración de del proyecto, para esto se tiene planeado un total de 8 reuniones en donde se contemplará:

- Transporte (gasolina) = \$1,500 MXN
- Comida = \$2,000 MXN

Por último siempre pueden ocurrir gastos inesperados en donde el presupuesto del proyecto no se toma en cuenta por lo que gastos no tomados en cuenta podrían ser hasta \$3,000 MXN  
-Precio total del proyecto **\$124,665 MXN**

#### Precio de venta al por menor sugerido.

El precio de sugerido y calculado es de \$170 MXN

#### Proyección de Ganancias

Realizando un análisis de mercado e invirtiendo se llegaron a las siguientes cifras:

- Primer perfil: Pesimista  
Se espera que después de tres meses del lanzamiento del proyecto al publico o empresa se recupere el precio total del proyecto \$124,665 MXN (por copia descargada)
- Segundo perfil: Esperado  
Se espera que después de tres meses del lanzamiento del proyecto al público o empresa se recupere \$249,330 MXN invertidos inicialmente (por copia descargada)
- Tercer perfil: Optimista  
Se espera que después de tres meses del lanzamiento del proyecto al público o empresa se recupere \$311,662 MXN invertidos inicialmente (por copia descargada)

## **4. Descripciones de actividades en Ingles**

### **Osorio Alvarado Jorge Adalberto**

The activities I performed for this project were: 3D pond models using boxes, cylinders, and textures: this task I was able to perform satisfactorily with the knowledge acquired in the laboratory practices. The textures were taken from the internet and the frames were created with the GIMP tool adding transparency giving the sensation of glass. As the project progressed, these same bases were used for the creation of the ponds observed in the final result.

Models for the ponds: all the models I used in the project were taken from free3d.com, some did not have textures, however, I was able to create them on my own using again the GIMP tool.

Animation of a turtle, an egg, and a penguin: these animations were made entirely with algorithms using the logic seen in the practice of complex animations.

Scenery scaling: one of the problems we faced in this project was the scaling of the scenery, the scale was very large and the objects were too heavy, so I decided to scale it in its entirety using 3dMax, with the help of my partner Juan we could export all the models as they were in the beginning because in my computer did not load the materials of some objects in its entirety, but he did not have that problem.

Optimization of objects: as the project was still taking a long time to run because I don't have a good computer, I decided to optimize all the objects by scaling all the images because there were some that were too heavy, when I finished scaling all the images, the project ran a minute faster and was much more fluid. After that, the project could run better for everyone.

Signs for the stage: I made signs that made the user experience much more intuitive, being able to easily interact with the animations being inside the stage.

### **Gonzalez Castro Juan Fernando**

The activities I worked with for this project were mainly focused on the creation of the workspace, I mean the creation and arrangement of the static objects that make up our project. What I did first was understand the problems and then order the ideas that all my colleagues and I had and be able to make a good proposal.

I worked a lot with 3dsMax to create objects like fish tanks and some other objects that we can find in the workspace, however for this it was very difficult for me to understand the program, but with the help of friends and a teacher I was able to understand it better.

I also worked with the creation and modulation of animations; in my case it was the animation of a car that travels a path that is around our aquarium. However, I ran into some problems at the time of animation because I didn't understand how pivot points were handled, luckily, I understood how it works and was able to create such animation.

I think this project gave us a basic idea of the world that is infographics, it gives us an idea of what could be done professionally using these basic concepts, sometimes it's a bit complicated to understand but with these concepts one can do very impressive things like an aquarium.



## Lopez Cruz Luis Enrique

Before starting the project we had to do a review of all the studies involved in the project. This activity will include model import, light management, texturing, tools like 3D Max, and model correction.

The main problem I had was the dimensions and units of the land, I think that a unit system must have been necessary or at least contemplated to speed up all the processes, but once the floor was built and textured, the next step to give shape was the skybox. and look for a sky that would fit well, this was found on a video game mods forum.

The next step, once the models of my colleagues and their respective modifications have been imported, look for a shark, Golden fish, souvenir shop, stand, trees, a penguin, crab and other models. The main problem with this was that many models were badly exported and the textures did not match, for this I corrected some with 3dmax and others with mtl.

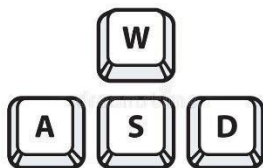
The difficult part that I did was the implementation of the animation by keyFrames, this was difficult because first I had to think about how I would do the animation if it would be hardcoding the values of the frames or taking it directly from a file to make it look cleaner and more convenient. In the end I decided that because of a text document, the tricky part of the animation was the route that it would take since as I mentioned before the route is difficult due to the lack of a coordinate system, in the end I had to more or less feel for the coordinates and As for the model, I had to split it into joints, this part was theoretically speaking and in 3D Max manipulation and the calculation of the pivot, I corrected this error by putting the entire axis of rotation at the origin and using code to place it where there was the penguin with a translation.

In terms of activities, I feel that this project was an incorporation of everything, so it seemed like a joint challenge to solve and I had a lot of fun, but the activities make it clear to me that they must be resolved with time if not everything comes together.

## 5-. Guía general de controles del proyecto

- **Teclas "W", "A", "S", "D"**

Con estas teclas como hemos mencionado podemos movernos dentro de nuestro espacio.



Tecla W: Movimiento hacia el frente

Tecla A: Movimiento hacia la izquierda

Tecla S: Movimiento hacia abajo

Tecla D: Movimiento hacia la derecha.

- **Tecla "1"** Se ejecuta la animación del Automóvil
- **Tecla "2"** Se ejecuta la animación del Pingüino de la Pecera
- **Tecla "3"** Se ejecuta la animación de la Tortuga
- **Tecla "4"** Se ejecuta la animación del huevo de tortuga
- **Tecla "5"** Se ejecuta la animación del huevo de tortuga que nada
- **Tecla "8"** Se ejecuta la animación del Tiburón
- **Tecla "9"** Se ejecuta la animación del Pingüino que recorre el acuario

Ligas de acceso.

GitHub VIEJO.

<https://github.com/JuanGonzalez1212/ProyectoFinal.git>

GitHub Nuevo

<https://github.com/LuisG8/ProyectoFinalCompu.git>

Google Drive.

[https://drive.google.com/drive/folders/1i1gx5KQerNxi93OGFGVMhmmdoJnP\\_AweD5?usp=sharing](https://drive.google.com/drive/folders/1i1gx5KQerNxi93OGFGVMhmmdoJnP_AweD5?usp=sharing)