

**INFORME ABET**

CURSO DE PROGRAMACIÓN II – CC67

Carreras de

Ing. de Software

Ing. de Sistemas de la Información

Ciencias de la Computación

Sección: SS2A

Alumnos:

U20181E301 - Carlo Andre Bolivar Orosco

U201918658 - Freddy Manuel Osorio Frías

U202012835 - Jean Sebastian Anduaga Beramendi

U20171F413 - Sofía Beatriz Allca Urbano

Noviembre 2020

**ÍNDICE**

1. **Introducción**
2. **Objetivo del estudiante**
3. **Diseño del diagrama de clases**
4. **Plan de actividades**
5. **Diseño del producto y funcionalidad adicional**
6. **Aporte**
7. **Conclusiones**
8. **Anexos**
9. **Bibliografía**
10. **INTRODUCCIÓN**

En la historia se han escrito grandes obras de arte ficticias y reales que en la actualidad son consideradas maravillas y también existen tantas construcciones como laberintos, torres y castillos. Estas ideas fueron creadas gracias a las mentes brillantes e imaginativas mentes de los escritores de aquellas épocas, de quienes se desarrollaron diferentes y nuevas formas de arte. El ingeniero de software puede desarrollar todo tipo de historias y contarlas desde una perspectiva a modo de juego el cual le da un toque personal a cada historia y algo nuevo cada vez que se cuenta. En el siguiente trabajo se va a contar una historia sobre un líder del reino Zion X el cual debe llegar al castillo a toda costa…

1. **OBJETIVO DEL ESTUDIANTE**

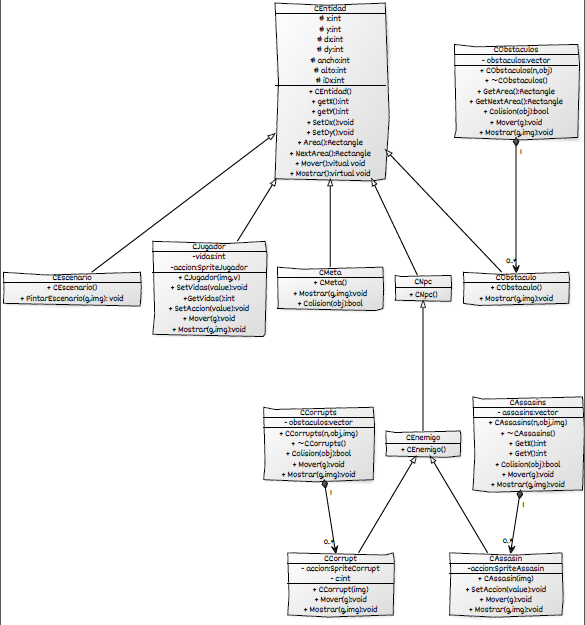
Jean Sebastian Anduaga Beramendi (EAC): El desarrollo de este trabajo me ha enseñado a cumplir con necesidades específicas que me son dadas.

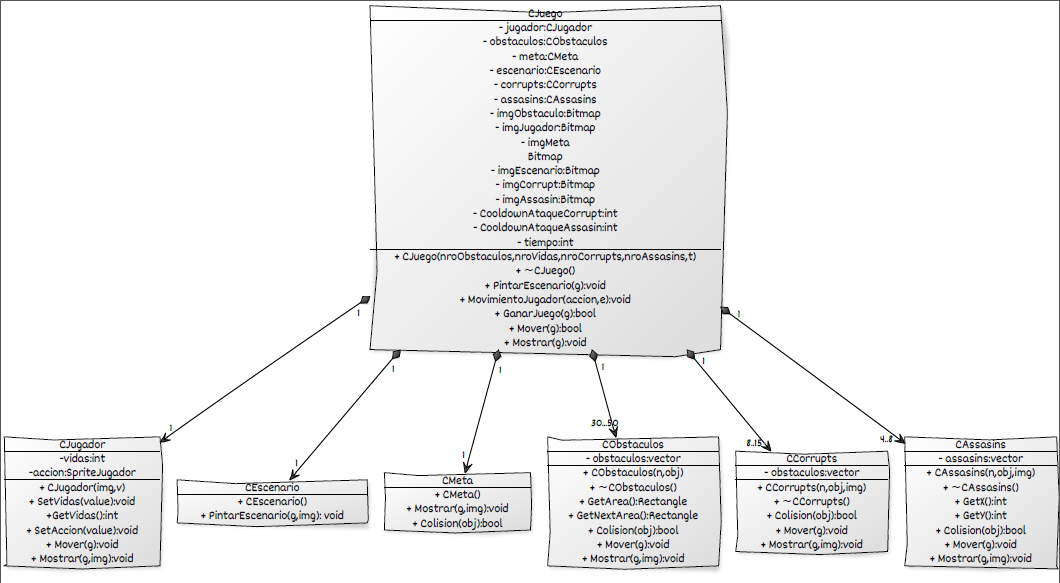
Sofia Beatriz Allca Urbano (CAC): Mi objetivo durante la creación de este proyecto es poder brindar soluciones a las dificultades que se presenten.

Carlo Andre Bolivar Orosco (EAC):El desarrollo de este trabajo me ha permitido emplear todo el conocimiento obtenido del curso ayudándome a mejorar en los aspectos faltantes y a incrementar mi conocimiento en los aspectos más fuertes de este.

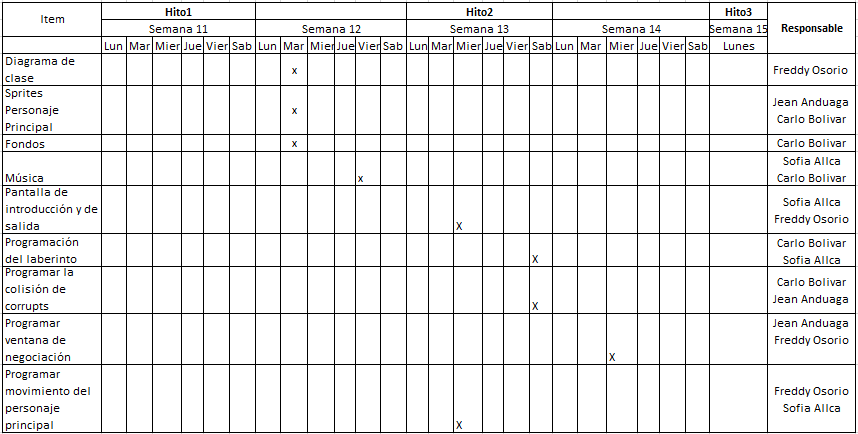
Freddy Manuel Osorio Frías (EAC): Adquirir las habilidades necesarias para conseguir información que me permitan resolver problemas que aparezcan en el desarrollo de un proyecto.

1. **DISEÑO DEL DIAGRAMA DE CLASES**





1. **PLAN DE ACTIVIDADES**



1. **DISEÑO DEL PRODUCTO Y FUNCIONALIDAD ADICIONAL**

**// Funcionalidad Adicional los enemigos no tienen colisión con los obstáculos incrementando la dificultad al jugador el que carece de esta adición.**

Mover y Colisión:

bool Mover(Graphics^ g)

{

//Quitar vidas con colision de Corrupts

if (corrupts->Colision(jugador->Area()) && clock()-CooldownAtaqueCorrupt >= 2000)

{

jugador->setVidas(-1);

CooldownAtaqueCorrupt = clock();

if (jugador->getVidas() == 0)

{

return false;

}

}

if(assasins->Colision(jugador->Area()) && clock() - CooldownAtaqueAssasin >= 2000)

{

jugador->setVidas(-1);

CooldownAtaqueAssasin = clock();

if (jugador->getVidas() == 0)

{

return false;

}

}

if (tiempo <= clock())

return false;

//Mover

if (obstaculos->Colision(jugador->NextArea()) == false)

jugador->Mover(g);

obstaculos->Mover(g);

meta->Mover(g);

corrupts->Mover(g);

assasins->Mover(g);

return true;

}

Colisión:

bool Colision(Rectangle obj)

{

for each (CObstaculo \* obs in obstaculos)

{

if(obs->NextArea().IntersectsWith(obj))

{

return true;

}

}

return false;

}

Mostrar:

void Mostrar(Graphics^ g)

{

g->DrawString("Tiempo : " + ((tiempo - clock())/1000), gcnew Font("Arial", 12), Brushes::Blue, 75, 0);

jugador->Mostrar(g, imgJugador);

meta->Mostrar(g, imgMeta);

obstaculos->Mostrar(g, imgObstaculo);

corrupts->Mostrar(g, imgCorrupt);

assasins->Mostrar(g, imgAssasin);

}

1. **APORTE**

En el presente trabajo hemos implementado todo lo aprendido durante el curso, desde punteros hasta el uso de formularios. Además, hemos investigado para encontrar soluciones a los problemas que se nos presentaban y también aprender nuevas técnicas y formas a la hora de programar.

1. **CONCLUSIONES**

El programa nos ayuda a plantear una salida, un escape para poder triunfar ante fuerzas enemigas a pesar del tiempo y las circunstancias.

-El programa funciona correctamente a la hora de mostrar obstáculos.

-El programa grafica correctamente una salida.

-El programa será de utilidad en procesos recreativos

1. **ANEXOS**

Inspiración para el laberinto en una etapa temprana del juego(antes de reformular la idea):

<https://www.lawebdelprogramador.com/codigo/Dev-C/5012-Crear-y-resolver-un-laberinto.html>

Modelo Final del líder el cual es una edición de un modelo existente: 

Modelo Final del Assassin el cual es una edición de un modelo existente:

Modelo Final del Corrupt el cual es una edición de un modelo existente:

Modelo Final del Aliado el cual es una edición de un modelo existente:

**Código YUML:**

**Parte 1:**

[CEntidad|# x:int;# y:int;# dx:int;# dy:int;# ancho:int;# alto:int;# iDx:int|+ CEntidad();+ getX():int;+ getY():int;+ SetDx():void;+ SetDy():void;+ Area():Rectangle;+ NextArea():Rectangle;+ Mover():vitual void;+ Mostrar():virtual void]

[CEscenario|+ CEscenario();+ PintarEscenario(g,img): void]

[CJugador|-vidas:int;-accion:SpriteJugador|+ CJugador(img,v);+ SetVidas(value):void; +GetVidas():int;+ SetAccion(value):void;+ Mover(g):void;+ Mostrar(g,img):void]

[CMeta|+ CMeta();+ Mostrar(g,img):void;+ Colision(obj):bool]

[CObstaculo|+ CObstaculo();+ Mostrar(g,img):void]

[CObstaculos|- obstaculos:vector|+ CObstaculos(n,obj);+ ~CObstaculos();+ GetArea():Rectangle;+ GetNextArea():Rectangle;+ Colision(obj):bool;+ Mover(g):void;+ Mostrar(g,img):void]

[CNpc|+ CNpc()]

[CEnemigo|+ CEnemigo()]

[CCorrupt|- accion:SpriteCorrupt; - c:int|+ CCorrupt(img);+ Mover(g):void;+ Mostrar(g,img):void]

[CCorrupts|- obstaculos:vector|+ CCorrupts(n,obj,img);+ ~CCorrupts();+ Colision(obj):bool;+ Mover(g):void;+ Mostrar(g,img):void]

[CAssasin|-accion:SpriteAssasin|+ CAssasin(img);+ SetAccion(value):void;+ Mover(g):void;+ Mostrar(g,img):void ]

[CAssasins|- assasins:vector|+ CAssasins(n,obj,img);+ ~CAssasins();+ GetX():int;+ GetY():int;+ Colision(obj):bool;+ Mover(g):void; + Mostrar(g,img):void]

[CEntidad]^-[CJugador]

[CEntidad]^-[CNpc]

[CEntidad]^-[CMeta]

[CEntidad]^-[CObstaculo]

[CEntidad]^-[CEscenario]

[CNpc]^-[CEnemigo]

[CEnemigo]^-[CCorrupt]

[CEnemigo]^-[CAssasin]

[CObstaculos]++1-0..\*>[CObstaculo]

[CCorrupts]++1-0..\*>[CCorrupt]

[CAssasins]++1-0..\*>[CAssasin]

**Parte 2:**

[CJuego|- jugador:CJugador;- obstaculos:CObstaculos;- meta:CMeta;- escenario:CEscenario;- corrupts:CCorrupts;- assasins:CAssasins;- imgObstaculo:Bitmap;- imgJugador:Bitmap;- imgMeta;Bitmap;- imgEscenario:Bitmap;- imgCorrupt:Bitmap;- imgAssasin:Bitmap;- CooldownAtaqueCorrupt:int;- CooldownAtaqueAssasin:int;- tiempo:int|+ CJuego(nroObstaculos,nroVidas,nroCorrupts,nroAssasins,t);+ ~CJuego();+ PintarEscenario(g):void;+ MovimientoJugador(accion,e):void;+ GanarJuego(g):bool;+ Mover(g):bool;+ Mostrar(g):void]

[CJugador|-vidas:int;-accion:SpriteJugador|+ CJugador(img,v);+ SetVidas(value):void; +GetVidas():int;+ SetAccion(value):void;+ Mover(g):void;+ Mostrar(g,img):void]

[CEscenario|+ CEscenario();+ PintarEscenario(g,img): void]

[CMeta|+ CMeta();+ Mostrar(g,img):void;+ Colision(obj):bool]

[CObstaculos|- obstaculos:vector|+ CObstaculos(n,obj);+ ~CObstaculos();+ GetArea():Rectangle;+ GetNextArea():Rectangle;+ Colision(obj):bool;+ Mover(g):void;+ Mostrar(g,img):void]

[CCorrupts|- obstaculos:vector|+ CCorrupts(n,obj,img);+ ~CCorrupts();+ Colision(obj):bool;+ Mover(g):void;+ Mostrar(g,img):void]

[CAssasins|- assasins:vector|+ CAssasins(n,obj,img);+ ~CAssasins();+ GetX():int;+ GetY():int;+ Colision(obj):bool;+ Mover(g):void; + Mostrar(g,img):void]

[CJuego]++1-1>[CJugador]

[CJuego]++1-1>[CEscenario]

[CJuego]++1-1>[CMeta]

[CJuego]++1-30...50>[CObstaculos]

[CJuego]++1-8..15>[CCorrupts]

[CJuego]++1-4..8>[CAssasins]

1. **BIBLIOGRAFÍA**

<https://www.spriters-resource.com/ds_dsi/legobattles/sheet/69157/>

[**https://www.spriters-resource.com/ds\_dsi/legobattles/sheet/77023/**](https://www.spriters-resource.com/ds_dsi/legobattles/sheet/77023/)

[**https://www.spriters-resource.com/ds\_dsi/legobattles/sheet/69156/**](https://www.spriters-resource.com/ds_dsi/legobattles/sheet/69156/)

[**https://www.spriters-resource.com/ds\_dsi/legobattles/sheet/69251/**](https://www.spriters-resource.com/ds_dsi/legobattles/sheet/69251/)