

**Asignatura:**

**ISC-210-T-001**

**Trabajo de Curso:**

**Recoge los cheles**

**Autores:**

**20170861 Joseph De Los Santos**

**20170502 Ruben O. Díaz**

[Matrícula y Nombre del Estudiante 3]

**Tutor:**

**Prof. Miguel T. Moronta**

**Entregado en Fecha:**

8 de marzo del 2019

**PONTIFICIA UNIVERSIDAD CATÓLICA MADRE Y MAESTRA – PUCMM  
FACULTAD DE CIENCIAS DE LAS INGENIERÍAS  
Departamento de Ingeniería en Sistemas y Computación – ISC**

# Resumen / Abstracto del Problema

Se desea programar un videojuego sencillo, con objetos y animaciones sencillas que incorpore por lo menos cuatro funciones físicas, detección de colisiones y el efecto parallax. El mismo debe ser en 2D.

You want to program a simple video game, with simple objects and animations that incorporate at least four physical functions, collision detection and the parallax effect. The same must be in 2D.

Tabla de contenido

[Resumen / Abstracto del Problema 1](#_Toc409432871)

[Análisis 1](#_Toc409432872)

[Cuerpo del Informe 1](#_Toc409432873)

[Código Fuente 1](#_Toc409432874)

[Condiciones de Error 1](#_Toc409432875)

[Notas al Profesor 1](#_Toc409432876)

[Conclusiones 1](#_Toc409432877)

[Recomendaciones 1](#_Toc409432878)

[Referencias Bibliográficas 1](#_Toc409432879)

# Análisis

El juego por imitar es “Dash De Coins”, un juego tipo plataforma en el cual se debe recolectar monedas para obtener mayor puntaje. El jugador solo puede “invertir la gravedad” para evitar los obstáculos del mapa, recoger más monedas y evitar caerse o salirse del mapa.

El jugador solo puede invertir la gravedad, si y solo si, está en contacto con alguna superficie. Y si queda atascado en una plataforma, como esta está en movimiento lo sacará de la pantalla y perderá automáticamente.

El personaje y la cámara permanecerán estáticos, a diferencia de las plataformas que se mantendrán en movimiento aplicándole una fuerza constante hasta que se salga de la pantalla y sea destruida. De fondo se moverán algunos elementos a diferentes velocidades para aplicar el efecto parallax, el cual dará la sensación de profundidad.

The game to imitate is "Dash De Coins", a platform type game in which coins must be collected to obtain a higher score. The player can only "reverse gravity" to avoid obstacles on the map, collect more coins and avoid falling or leaving the map.  
The player can only reverse gravity, if and only if it is in contact with any surface. And if it gets stuck on a platform, since it is in motion it will take it out of the screen and it will automatically lose.  
The character and the camera will remain static, unlike the platforms that will keep moving, applying a constant force until it leaves the screen and is destroyed. In the background, some elements will move at different speeds to apply the parallax effect, which will give the sensation of Depth.

# Cuerpo del Informe

Capítulo 1 – Descripción del Negocio

Se nos pidió hacer un juego implementando al menos 4 fórmulas. El juego deberá ser jugable y poder jugarse varias veces sin interrupciones y sin fallos mayores.

Capítulo 2 – Solución Propuesta

El video juego tiene 3 estados generales el primer estado se define por la instanciación del primer mapa y la jugabilidad. En este estado el programa genera aleatoriamente mapas, monedas, y puyas(spikes).

void CargarMapa()

{

TextAsset []contenido= {Resources.Load<TextAsset>("Mapa 1"), Resources.Load<TextAsset>("Mapa 2"), Resources.Load<TextAsset>("Mapa 3") };

Quaternion rotacion;

int mapa = 0;

float i = 0, j = 0;

GameObject nuevaCelda = null;

tiempoInicio = Time.time;

if (!primerMapa)

{

mapa = Random.Range(0, 3);

j = 15.6f;

}

foreach (string lineaActual in contenido[mapa].text.Split('\n'))

{

foreach(char celdaActual in lineaActual)

{

switch (celdaActual)

{

case '\_':

nuevaCelda = bloque;

rotacion = bloque.transform.rotation;

break;

case '|':

nuevaCelda = bloqueVertical;

rotacion = bloqueVertical.transform.rotation;

break;

case '/':

nuevaCelda = bloque;

rotacion = bloque.transform.rotation;

rotar = true;

break;

case '-':

nuevaCelda = bloque;

rotacion = bloque.transform.rotation;

spike = true;

rotar = true;

break;

case '\*':

nuevaCelda = bloque;

rotacion = bloque.transform.rotation;

spike = true;

break;

default:

j += 4.25f;

continue;

}

nuevaCelda = Instantiate(nuevaCelda, new Vector3(j, -i), rotacion);

if (rotar)

{

nuevaCelda.GetComponent<SpriteRenderer>().flipY = true;

rotar = false;

}

if (spike)

{

nuevaCelda.GetComponent<MovimientoPlataforma>().Spikes = true;

spike = false;

}

j += 4.25f;

}

j = 0;

if (!primerMapa)

{

j = 15.6f;

}

i += 3;

}

Aquí se puede apreciar el algoritmo de la selección del mapa pre-creado y su instanciación. Este se llama cuando un mapa ha sido totalmente des instanciado. Para la primera corrida se instancia un mapa en el start.

Las monedas se generan con una probabilidad, si es muy alta la plataforma tendrá una moneda encima si es muy baja puede tener una espina o no tener nada.

if(gameObject.tag != "Plataforma 90")

{

probMoneda = Random.Range(1, 5);

if (probMoneda < 3 && !Spikes)

{

if (gameObject.GetComponent<SpriteRenderer>().flipY)

{

moneda = Instantiate(coin, new Vector3(gameObject.transform.position.x, gameObject.transform.position.y - 1), Quaternion.identity);

}

else

{

moneda = Instantiate(coin, new Vector3(gameObject.transform.position.x, gameObject.transform.position.y + 1), Quaternion.identity);

}

moneda.transform.parent = gameObject.transform;

}

if (Spikes)

{

if (gameObject.GetComponent<SpriteRenderer>().flipY)

{

spike = Instantiate(puya, new Vector3(gameObject.transform.position.x, gameObject.transform.position.y - 0.75f),Quaternion.identity);

spike.GetComponent<SpriteRenderer>().flipY = true;

}

else

{

spike = Instantiate(puya, new Vector3(gameObject.transform.position.x, gameObject.transform.position.y + 0.75f), Quaternion.identity);

}

spike.transform.parent = gameObject.transform;

}

}

En este fragmento de código se puede apreciar mejor que cada plataforma es un objeto que se auto define, y se mueve independientemente pero su instanciación depende de la clase controladora, en este caso CentroJuego.

En el segundo estado se define si el jugador perdió una vida o si perdió el juego. Si el jugador pierde una vida se le reinicia el puntaje, pero no las monedas ya que estas están para que él pueda obtener más vidas.

case EstadoJuego.Fase2:

vida--;

if (vida == 0)

{

estado = EstadoJuego.Fase3;

}

else

{

SceneManager.LoadScene("Dash de Coins");

}

break;

El puntaje es una variable que depende de la distancia recorrida por el jugador la cual es calculada por la formula física:



Donde la velocidad inicial siempre será 0 y el tiempo de juego está siempre dado por la clase Time. Teniendo esto en cuenta lo mejor que se puede hacer es cargar la escena para que el tiempo se reinicie solo.

El jugador tiene sus propias propiedades físicas, en las cuales se usaron unas físicas personalizadas para la caída libre y la caída parabólica.

void FixedUpdate()

{

cambioPosicion.x = velocidadFinal.x \* Time.deltaTime + Physics.gravity.x \* (Mathf.Pow(Time.deltaTime, 2) / 2);

cambioPosicion.y = velocidadFinal.y \* Time.deltaTime + Physics.gravity.y \* (Mathf.Pow(Time.deltaTime, 2) / 2);

gameObject.transform.Translate(cambioPosicion);

velocidadFinal += direccion \* Physics.gravity \* Time.deltaTime;

}

En este fragmento de código perteneciente la clase Jugador, la cual es responsable de todo lo que el jugador puede hacer y casi todas las interacciones del mundo con el personaje. Aquí se le está aplicando la gravedad al personaje

if ((Input.GetKeyDown(KeyCode.Space)|| Input.GetMouseButton(0)) && isOnPlat)

{

gameObject.GetComponent<SpriteRenderer>().flipY = !gameObject.GetComponent<SpriteRenderer>().flipY;

direccion \*= -1;

isOnPlat = false;

}

Cuando se presiona espacio o el botón izquierdo del ratón la dirección se invierte cambiando así el vector velocidad final el cual es afectado por la gravedad.

# Código Fuente

Bajo este título se incluye el código fuente del programa a entregar. Deberá siempre anexar al documento el o los archivo(s) fuente para poderlos compilar de manera adecuada. Puede adjuntar en documento Zip toda la solución, en caso de que sea necesario. En caso de que su proyecto tenga múltiples archivos de código fuente, o resulte impráctico colocarlo dentro de esta sección, bastará con enviar el código junto con este reporte sin incluirlo en esta sección.

# Condiciones de Error

Por lo general, hay motivos que pueden causar que el usuario de un programa obtenga una anomalía o que el algoritmo usado tenga ciertas restricciones de uso. Ejemplo de esto es cuando a un programa que opera con números positivos se le digitan negativos. Ocurre también con funciones que no son capaces de operar con un número muy grande (por ejemplo, el factorial).

Estas anomalías deben enumerarse en esta sección del trabajo. Deberá incluir ejemplos de corrida con errores en esta sección donde se contemple el mismo claramente y se pueda visualizar cualquier mensaje de error desplegado. Tenga cuidado al decir que su programa no tiene condiciones de error. Quedará luego a consideración del profesor valorar las condiciones de error.

# Notas al Profesor

En caso de que desee hacer un señalamiento al profesor acerca de su trabajo, incluirlo bajo este título al final. Puede incluir cualquier otro aspecto que el equipo considere necesario exponer o explicar.

Por ejemplo: aspectos tenidos en cuenta para el diseño de la interfaz, filosofía de trabajo en la aplicación, tratamiento de errores, ayudas, aspectos novedosos tenidos en cuenta en la implementación, organización del trabajo en equipo, cronogramas de trabajo (plan y real), entrevistas (planificadas, reales, plantillas utilizadas, tipos de usuarios/clientes entrevistados) , estándar de código utilizado, ente otros..

# Conclusiones

Aquí se abordan las conclusiones del trabajo realizado, no de la asignatura, ni sobre los beneficios que le reporta a un estudiante la realización de este proyecto de curso. Deben redactarse teniendo en cuenta:

* Estado de cumplimiento de los objetivos planteados en la introducción.
* Observaciones, mediciones, estilos, etc., puestos en práctica durante la ejecución del proyecto. Por ejemplo: conclusiones, que consideren de interés a partir de sus propias experiencias, respecto a la organización del trabajo en equipo.

# Recomendaciones

Se redactan teniendo en cuenta aspectos que quedaron pendientes, mejoras que en el futuro se pueden realizar a la aplicación, posibilidades de implantación, entre otros.

# Referencias Bibliográficas

Especifique las referencias bibliográficas utilizadas durante la elaboración del proyecto.