**PROYECTO FINAL INFORMATICA 2**

**MARIA ISABEL GIRALDO GUZMAN**

**ANDRES FELIPE SAN MARTIN**

**Docente**

**Augusto Enrique Salazar Jimenez**

**FACULTAD DE INGENIERÍA**

**UNIVERSIDAD DE ANTIOQUIA**

**MEDELLÍN**

**2023-1**

**Desarrollo de la idea del juego**

**Introducción a la temática del juego:**

-El juego Abuela´s Run, consiste en la incesante misión de una abuela por alcanzar a su nieto para reprenderlo, ya que este se comió sus galletas. Como la mayoría de las abuelas esta va tras su nieto con una chancla, al nieto darse cuenta de la situación sale corriendo ya que no quiere ser alcanzado por la chancla. ¿Sera capaz la abuela de darle a su nieto con la chancla para que este aprenda la lección? O ¿El nieto será tan escurridizo para no permitir ser alcanzado por la chancla de su abuela?

El objetivo del juego es que después de cierta cantidad de chanclazos que la abuela logre darle al nieto el decida darse por vencido, ya que gracias a estos aprendió su lección de no robar las galletas, para esto nuestra querida abuela contara con cierta cantidad predeterminada de chanclas pues si lo quiere alcanzar deberá contar con una muy buena puntería. A Lo largo del mapa la abuela contara con ciertos comodines llamados super chanclas, las cuales harán el intento reprensión del nieto un poco más fácil. Si la abuela agota todas sus chanclas y aun no logra que su nieto aprenda la lección se dará por perdido el nivel; si el porcentaje de rebeldía del nieto llega a cero antes de que la abuela agote todas sus chanclas pasara de nivel.

**Vista del juego:**

-La vista que se escogió para el juego será la side-scroller view. Esta consiste en que el jugador se desplaza por un escenario en dos dimensiones, usualmente de izquierda a derecha o de derecha a izquierda, superando obstáculos, derrotando enemigos y recolectando objetos en el camino. El término "side-scroller" se refiere como tal a la acción de la pantalla, donde la vista se desplaza lateralmente a medida que el jugador avanza. Algunos juegos emblemáticos o de mayor renombre que usan este tipo de vista son: Super Mario Bros, Sonic the Hedgehog, Crash Bandicoot XS. En la figura 1 podemos observar un ejemplo de cómo es la side-scroller view:



Figura 1.

**Físicas del juego:**

En el desarrollo del juego se contarán con diferentes físicas aplicadas a objetos, algunas podrán ser compartidas entre ellos y otras serán totalmente únicas.

**-Físicas personaje abuela:**

-Se tendrá un movimiento parabólico el cual será implementado al momento de saltar algún objeto que se presente interviniendo el camino, o al momento de caer de un nivel superior a uno inferior. Este movimiento que ayudará a caracterizar el salto y la caída a un nivel inferior de la abuela se puede apreciar de una mejor forma en la figura 2.

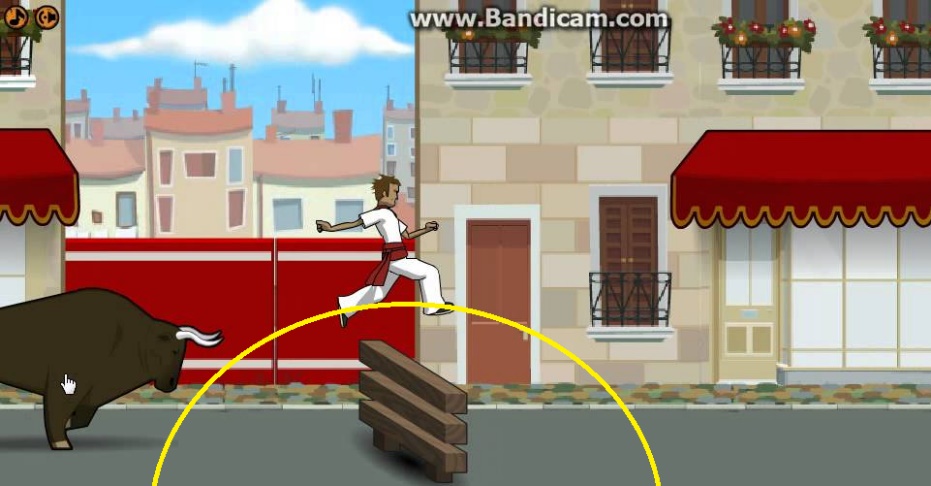


Figura 2.

**-Físicas personaje nieto:**

- Se tendrá un movimiento parabólico el cual será implementado al momento de saltar algún objeto que se presente interviniendo el camino, o al momento de caer de un nivel superior a uno inferior. Este movimiento que ayudará a caracterizar el salto y la caída a un nivel inferior del nieto se puede apreciar de una mejor forma en la figura 2.

**-Físicas chancla estándar:**

-La chancla estándar tendrá un movimiento parabólico al ser lanzada, con este movimiento se busca que se dificulte un poco más calcular donde debe ir la chancla para alcanzar al nieto. En la figura 3 podemos observar como el movimiento de la chancla seria semejante al que se realiza cuando lanzamos un pájaro en forma parabólica en el videojuego AngryBirds.

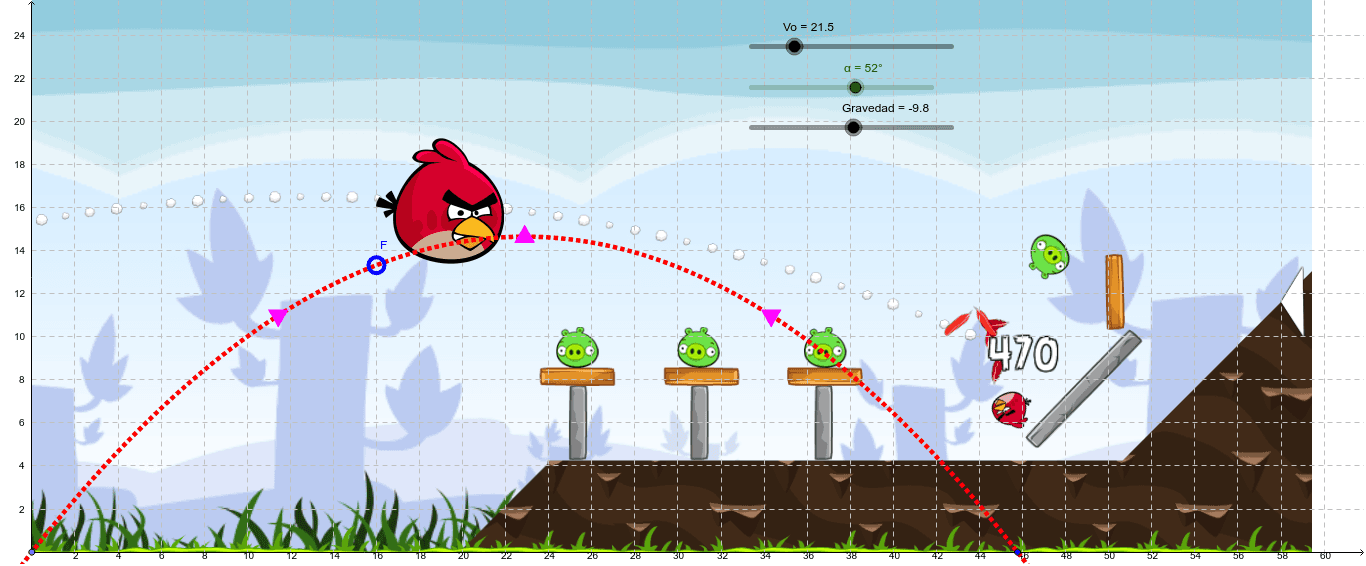


Figura 3.

-Esta chancla contara con un sistema de colisión, para así facilitar la detección del momento en el que la chancla tenga algún contacto con un obstáculo o en su defecto con el nieto, para posteriormente desaparecer y si es debido, es decir, si esta se encuentra colisionando con el nieto, se ejecuten las acciones que esto detonan, como lo es bajar el porcentaje de rebeldía del nieto.

**-Físicas super chancla:**

-La super chancla tendrá un movimiento rectilíneo contando con un poco más de velocidad de la que posee el movimiento parabólico de la chancla estándar, por lo tanto, en el momento que el usuario lo decida contará con un tiro con mayor efectividad para así asegurarse de que la chancla alcance al nieto. Este tipo de movimiento que tendría el lanzamiento de la chancla se puede observar en la figura 4.



Figura 4.

-Esta super chancla contara con un sistema de colisión, para así facilitar la detección del momento en el que la chancla tenga algún contacto con un obstáculo o en su defecto con el nieto, para posteriormente desaparecer y si es debido, es decir, si esta se encuentra colisionando con el nieto, se ejecuten las acciones que esto detonan, como lo es bajar el porcentaje de rebeldía del nieto de mayor forma de la que lo hace la chancla normal.

**Definición de clases:**

**Clase menú:**

Atributos:

char empezar\_juego: si se presiona espacio empezar a jugar.

Métodos:

void ejecutar\_clase\_juego(): si se presiona espacio se empiezan a ejecutar los métodos de la clase juego.

**Clase juego:**

Atributos:

double gravedad: se instancia una macro para la gravedad que se usara en todo el juego.

clase juego: public abuela,public nieto,public chancla, public super\_chancla, public mapa, public obstáculo.

Métodos:

void unión\_mapa\_escena(): se realiza la unión de la escena con el mapa, los dos con la misma longitud, se ira cambiando el foco de la escena para recorrer el mapa.

void mapa\_abuela: se sitúa la abuela en un punto inicial en el mapa.

void\_mapa\_nieto: se sitúa al nieto en un punto inicial del mapa.

void\_interaccion\_abuela\_nieto: definición de la distancia máxima y mínima entre los dos personajes.

void\_abuela\_chanclas\_n: se define la cantidad de chanclas que va a poseer la abuela y se verifica si aún posee una chancla normal para ser lanzada.

void\_chanclas\_mapa: detectar si la abuela obtuvo del mapa una super chancla para así poder realizar el cambio en el momento del siguiente lanzamiento.

void\_colisiones\_chanclas: se crea la colisión de las chanclas con el mapa o el nieto.

**Clase mapa:**

Atributos:

char tablero\_mxn: un arreglo de char de donde se obtendrá el mapa el cual se hará manualmente.

int m: es el alto del mapa.

int n: es el largo del mapa.

Métodos:

Void crear\_mapa: se lee la matriz para posteriormente crear el mapa.

Void inserta\_mapa\_escena: se inserta el mapa en la escena (tanto el mapa como la escena tendrán el mismo tamaño, nos iremos moviendo a través de ellos con setSceneRec).

Void introducir\_video: se introduce el video a la escena el cual generara el efecto de movimiento en el mapa.

**Clase obstáculo:**

Atributos:

double tamaño\_obstáculoA(): se define el alto de los obstáculos.

double tamaño\_obstáculol(): se define el largo de los obstáculos.

double pos\_x; se define la posición en x del obstáculo en el mapa.

double pos\_y: se define la posición en y del obstáculo en el mapa.

Métodos:

void generar\_en\_el\_mapa: se posiciona el obstáculo en el mapa.

**Clase abuela:**

Atributos:

double gravedad\_a: indica la gravedad de la abuela en el juego.

double x: posicion inicial en x de la abuela.

double y: posicion inicial de y de la abuela.

double velocidad\_x: velocidad de movimiento en x de la abuela.

double velocidad\_y: velocidad de salto en y de la abuela.

Métodos:

void mover\_personaje\_a(): este solo mueve hacia la derecha el personaje abuela mediante la interacción con el teclado.

void saltar\_a(): se define el salto parabólico de la abuela.

void lanzar\_chancla\_n(): se verifica la existencia de la chancla y se define el movimiento parabólico de ella.

void lanzar\_chancla\_s(): se verifica la existencia de la super chancla y se define el movimiento rectilíneo de ella.

**Clase nieto:**

Atributos:

double gravedad\_a: gravedad inicial en el juego del nieto.

double x: posición inicial en x del nieto.

double y: posición inicial de y del nieto.

double velocidad\_x: velocidad del movimiento en x del nieto.

double velocidad\_y: velocidad del salto en y del nieto.

Métodos:

void mover\_personaje\_nieto(): se define el movimiento automático del nieto este se mueve hacia la derecha con una velocidad constante.

void saltar\_nieto(): se define el salto parabólico para el nieto.

**Clase chancla:**

Atributos:

double x: posición inicial en x de la chancla (depende de donde este situada la abuela en ese instante de tiempo).

double y: posición inicial en y de la chancla (depende de donde este situada la abuela en ese instante de tiempo).

double velocidad\_x: velocidad del movimiento en x de la chancla.

double velocidad\_y: velocidad del movimiento en y de la chancla.

Métodos:

void lanzamiento­­\_parabólico­\_chancla(): se define el lanzamiento parabólico de la chancla.

**Clase superchancla:**

Atributos:

double x: posición inicial en x de la chancla (depende de donde este situada la abuela en ese instante de tiempo).

double y: posición inicial en y de la chancla (depende de donde este situada la abuela en ese instante de tiempo).

double velocidad\_x: velocidad del movimiento en x de la chancla.

Métodos:

void lanzamiento­­\_rectilíneo\_chancla(): se define el lanzamiento rectilíneo acelerado de la superchancla.

**Clase fin del juego:**

Atributos:

int rebeldia\_niño: en este entero se tiene el porcentaje de rebeldía del niño en el momento.

char salir\_juego: si se presiona enter se sale del juego.

char volver\_a\_jugar: si se presiona r se vuelve a ejecutar el juego desde cero.

Métodos:

void cerrar\_juego(): se verifica si el nivel de rebeldía es cero y se pregunta si desea salir del juego o volver a jugar.