Participantes

Juan Camilo García Castro

Juan Esteban Gonzales Alzate

Juego planeado para Proyecto final.

El juego que tenemos planeado es un personaje que se duerme y cae en un sueño, el sueño tiene algo en particular y es que, para poder despertarse, el personaje tiene que pasar un mundo buscando unas llaves y recogiéndolas, una vez las tiene todas pasa a otro mundo y así, la idea es crear 3 niveles diferentes, cada uno es un mundo.

La modalidad planeada es un juego de plataformas, con una vista general del mundo, obstáculos, caídas, y trampas, de esta manera, tenemos planeado agregar las propiedades físicas solicitadas…

Por ejemplo

Gravedad, cuando el personaje salta o caen trampas del techo o de la parte superior.

Movimiento rectilíneo uniforme, el movimiento del personaje

Movimiento rectilíneo acelerado, unos cañones.

Movimiento circular, trampas de corte

El fondo está pensado en ser cómo el del juego Mario Bros, una serie de fondos que van rotando, incluso se puede usar el tema de los sueños para hacer fondos hipnóticos.

ENTREGA 2 – CLASES DE DEPURADAS.

class Inicio;

}

class Inicio :

{

public:

explicit Inicio();

~Inicio();

void startgame();

void abrir();

void closegame();

void loadgame();

void loadclose();

void newgame();

private slots:

void botonnew ();

void botonload ();

Clase de inicio para el menú, que incluirá botones de iniciar partida y cerrar partida, también funciones para cargar, cerrar o abrir el juego.

#define COLECCIONABLES\_H

class Coleccionables

{

private:

int tipo;

float px,py;

public:

Coleccionables();

void paint();

Coleccionables (float \_px, float \_py, int \_tipo);

};

Coleccionables que servirán para puntaje en el juego, el usuario puede escoger si ir por ellos. Estarán agregados en posiciones cada uno px y py.

class objetos :

{

private:

int tipo;

float px,py,vx=10,vy;

proyectiles \*bala;

int cont = 0;

float theta = 90; // angulo

float w = sqrt(G/1); //gravedad por logingitud

float i = 1;

float pyini,pxini;

public:

objetos();

void paint()

objetos(float \_px, float \_py, int \_tipo);

proyectiles \* generarbala();

float getPx() const;

void setPx(float value);

float getPy() const;

void setPy(float value);

float getVx() const;

void setVx(float value);

float getVy() const;

void setVy(float value);

int getCont() const;

};

Clase pensada para los objetos que va a tener el juego, entre ellos, plataformas para subida y bajada, obstáculos, trampas y demás cosas que se deben esquivar.

Proyectiles serán pensados desde cañones que dispararán hacia una dirección, se incluirán formulas de gravedad y velocidad en ellas.

Los objetos tendrán posiciones, que cambiarán según se trasladen.

class Personaje :

{

private:

int vidas;

int puntaje;

int puntaje\_maximo;

float px,py,vx,vy,ax;

float ay = 10;

int nivel;

int llaves = 0;

int medallones;

std::string nombre;

public:

Personaje();

void paint();

Personaje(float \_px, float \_py, float \_vx, float \_vy, float \_ax,int \_vidas, int \_nivel, int \_puntaje);

void advance(int phase);

int getVidas() const;

void setVidas(int value);

int getPuntaje() const;

void setPuntaje(int value);

int getNivel() const;

void setNivel(int value);

float getVx() const;

void setVx(float value);

float getVy() const;

void setVy(float value);

float getAy() const;

void setAy(float value);

float getPx() const;

void setPx(float value);

float getPy() const;

void setPy(float value);

int getLlaves() const;

void setLlaves(int value);

std::string getNombre() const;

void setNombre(const std::string &value);

int getPuntaje\_maximo() const;

void setPuntaje\_maximo(int value);

};

Clase de personaje, el personaje tiene como condiciones privadas, vida, puntaje, coleccionables y llaves.

El personaje contiene parámetros públicos de movimiento, regidos por física, puede saltar entonces maneja posiciones y velocidad en el eje Y

Visualmente contiene parámetros públicos como vida y puntaje.

class proyectiles :

{

private:

float px,py,pxini,pyini,vx,vy,vyini,dt;

float ay = 10;

int tipo;

public:

proyectiles();

void paint();

void advance(int phase);

proyectiles(float \_px, float \_py, float vx, float vy,int tipo);

int getTipo() const;

void setTipo(int value);

};

Los proyectiles que tendrá que esquivar el usuario, serán lanzados desde cañones, tendrán parámetros privados de posición y velocidad, y parámetros públicos que serán el tipo de proyectil, y visualización de posición.