



**Midterm Project :
Don't let it cross**

**13016235 C Programming
Software Engineering Program
Faculty of Engineering, KMITL**

By

62011286 Unn Jertjamjarat

13016235: C Programming

First Semester, 2019

Project Proposal

1. Project developer

Student ID	Name
62011286	Unn Jertjamjarat

2. Project title

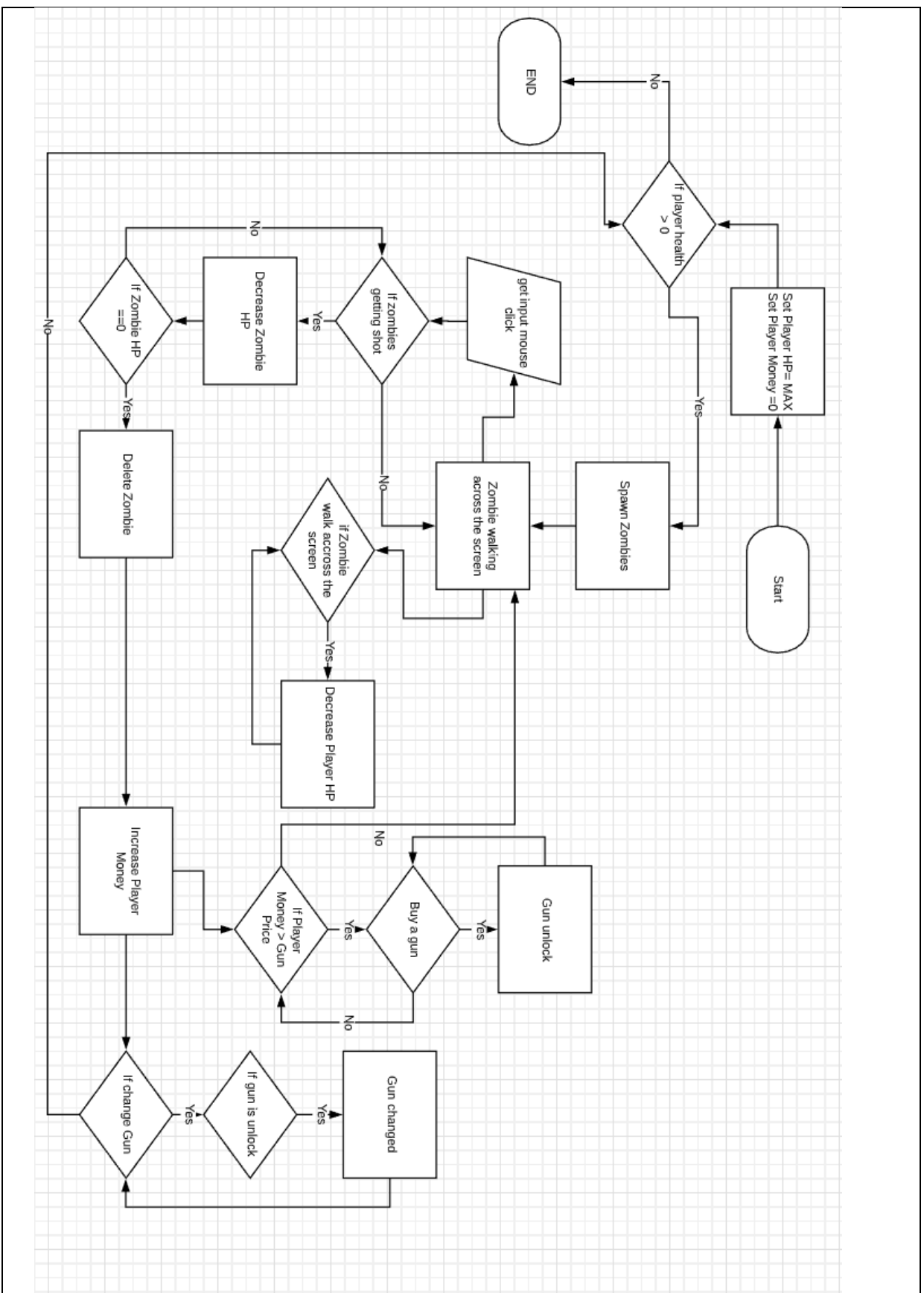
Don't let it cross

3. Project description and requirements

Project description:

Point and click 2D platformer game. Use a mouse to point and shoot the zombies.
Earn the money buy killing the zombies and purchase various weapons. The game over
When the zombies cross the screen and the health is zero.

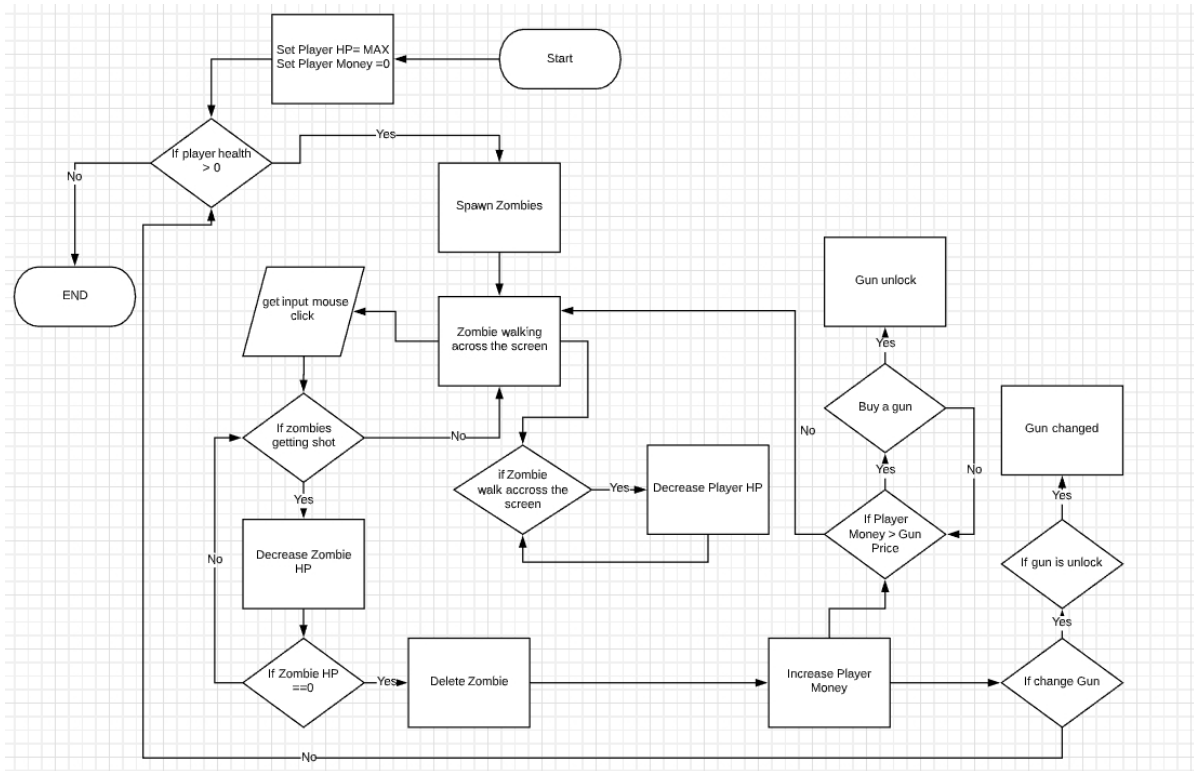
(Note: figures, flowcharts, diagrams may be used to describe the project)



Project requirements:

- C language IDE (MS Visual Studio 2017)
- Raylib library.
- Tile (for editing sprites)

(Describe the problem's statements or a specific request of the program that will be achieved.)



SOURCE CODE

```
#define _CRT_SECURE_NO_WARNINGS

#include "raylib.h"

#include <stdio.h>

#define MAX_FRAME_SPEED 15

#define MIN_FRAME_SPEED 1

#define MAX_ZOMBS 2000

#define SAND CLITERAL{255,208,122,255}

#define MAX_MACHINEGUN 300

#define MAX_SHOTGUN 20

#define MAX_NYAN_AMMO 10

#define MAX_NYAN 30
```

```
typedef struct Animation {
```

```
    Rectangle frameRec;
```

```
    int currentFrame;
```

```
    int framesCounter;
```

```
    int framesSpeed;
```

```
    Rectangle nyanRec;
```

```
    int nyanCurrentFrame;
```

```
    int nyanFramesCounter;
```

```
    int nyanFrameSpeed;
```

```
}Animation;
```

```
typedef struct collisionBox{
```

bool active;

Rectangle colliderZombs;

}collisionBox;

typedef struct Zombie {

bool active;

Vector2 position;

int health;

int speed;

int zombieCount;

}Zombie;

typedef struct Zombie2 {

bool active;

Vector2 position;

int health;

int speed;

int zombieCount;

}Zombie2;

typedef struct Player {

int money;

int gun;

int damage;

int health;

int wave;

bool startWave;


```
}Player;
```

```
typedef struct Guns {
```

```
    bool pistol;
```

```
    bool machineGun;
```

```
    bool shotGun;
```

```
    bool nyanGun;
```

```
    int machineGunAmmo;
```

```
    int shotGunAmmo;
```

```
    int nyanAmmo;
```

```
}Guns;
```

```
typedef struct Render {
```

```
    Texture2D pist;
```

```
    Texture2D ar;
```

```
    Texture2D shotGun;
```

```
    Texture2D bg_s;
```

```
    Texture2D crossHair;
```

```
    Texture2D shotCross;
```

```
    Texture2D startScreen;
```

```
    Texture2D bg_n;
```

```
    Texture2D bg;
```

```
    Texture2D ar_b;
```

```
    Texture2D shotGun_b;
```

Texture2D nyanGun;

Texture2D go;

}Render;

typedef struct CrossHair {

Vector2 mousePos;

Rectangle shotGunCross;

}CrossHair;

typedef struct NyanNyan {

Vector2 mousePos;

Rectangle nyanCross;

}NyanNyan;

typedef struct NyanShot {

Rectangle rec;

Vector2 speed;

Vector2 position;

bool active;

}NyanShot;

typedef struct HitSound {

Sound gunfire;

Sound marker;

bool active;

```
bool gActive;  
  
Sound oof;  
}HitSound;
```

```
//Global variable declaration.
```

```
static HitSound hitSound[100] = { 0 };  
  
static collisionBox collider[MAX_ZOMBS] = { 0 };  
  
static Zombie zombie[MAX_ZOMBS] = { 0 };  
  
static Player player[1] = { 0 };  
  
static Guns gun[] = { 0 };  
  
static Render render;  
  
static Animation animation;  
  
static CrossHair crosshair;  
  
static NyanNyan nyannyan[MAX_NYAN] = { 0 };  
  
static NyanShot nyanShot[MAX_NYAN] = { 0 };
```

```
static float zombieSpeed = 1.5;  
  
static int zombieSpawn = 20;  
  
const int screenWidth = 1600;  
  
const int screenHeight = 900;  
  
static int framesCounter = 0;
```

```
void tryRender(void) {  
  
    render.go = LoadTexture("sprite/gameOver.jpg");  
  
    render.bg_s = LoadTexture("sprite/bg_s.png");  
  
    render.pist = LoadTexture("sprite/pistol_s.png");  
  
    render.ar = LoadTexture("sprite/ar_s.png");
```

```

render.crossHair = LoadTexture("sprite/crossHair.png");
render.shotCross = LoadTexture("sprite/shotC.png");
render.shotGun = LoadTexture("sprite/shotgun.png");
render.startScreen = LoadTexture("sprite/Splash.png");
render.bg_n = LoadTexture("sprite/bg_n.png");
render.bg = render.bg_s;
render.ar_b = LoadTexture("sprite/ar_b.png");
render.shotGun_b = LoadTexture("sprite/shotgun_b.png");
render.nyanGun = LoadTexture("sprite/nyan.png");
}

void draw(void) {
    int ammo;

    ammo = NULL;

    DrawText(FormatText("MONEY : %i", player->money), 20, 30, 40, RED);
    DrawText(FormatText("Health : %i", player->health), 20, 80, 40, RED);
    DrawText(FormatText("Zombie Count : %i / %i", zombie->zombieCount,zombieSpawn), 20, 130, 40, RED);

    switch (player->gun) {
        case 0: DrawText(FormatText("Bullets : INFINITE"), 20, 180, 40, RED); break;
        case 1: ammo = gun->machineGunAmmo; break;
        case 2: ammo = gun->shotGunAmmo; break;
        case 3: ammo = gun->nyanAmmo; break;

    }

    if(player->gun !=0)
        DrawText(FormatText("Bullets : %i", ammo), 20, 180, 40, RED);

}

```

```
void initZombie(void) {  
    Texture2D zombie_default = LoadTexture("sprite/zombie_default.png");  
  
    for (int i = 0; i < MAX_ZOMBS; i++)  
    {  
        if (zombie[i].position.x == zombie[i - 1].position.x)  
            zombie[i].position.x = GetRandomValue(screenWidth - 3000, screenWidth - 1500);  
  
        if (zombie[i].position.y == zombie[i - 1].position.y)  
            zombie[i].position.y = GetRandomValue(0, screenHeight - zombie_default.height);  
  
        zombie[i].speed = zombieSpeed;  
        zombie[i].active = true;  
        zombie[i].health = 100;  
        zombie[i].position.x = GetRandomValue(screenWidth - 3000, screenWidth - 1500);  
        zombie[i].position.y = GetRandomValue(0, screenHeight - zombie_default.height);  
    }  
}
```

```
    UnloadTexture(zombie_default);  
  
}
```

```
void colliderZombies(void) {  
    Texture2D zombie_default = LoadTexture("sprite/zombie_default.png");
```

```
for (int i = 0; i < zombieSpawn; i++)
{
    if (zombie[i].active) {
        collider[i].active = true;
        collider[i].colliderZombs.x = zombie[i].position.x;
        collider[i].colliderZombs.y = zombie[i].position.y;
        collider[i].colliderZombs.width = zombie_default.width / 5;
        collider[i].colliderZombs.height = (float)zombie_default.height;
    }
}
UnloadTexture(zombie_default);
}
```

```
void init(void)
{

    zombieSpeed = 1.5;
    zombieSpawn = 20;
    player->startWave = true;
    player->health = 10;
    player->money = 8000;
    player->gun = 0;
    player->wave = 1;

    zombie->zombieCount = 0;
```

```
//Render sprite
```

```
tryRender();
```

```
//Init zombies
```

```
initZombie();
```

```
//Init Collider for zombies
```

```
colliderZombies();
```

```
//Init Player
```

```
//Init guns
```

```
gun->machineGun = false;
```

```
gun->shotGun = false;
```

```
gun->nyanGun = false;
```

```
gun->machineGunAmmo = 0;
```

```
gun->nyanAmmo = 0;
```

```
gun->shotGunAmmo = 0;
```

```
//Init nyanShot
```

```
for (int i = 0; i < MAX_NYAN; i++)
```

```
{
```

```
    nyanShot[i].rec.x = GetMousePosition().x - ((render.nyanGun.width / 5)/2);
```

```
    nyanShot[i].rec.y = GetMousePosition().y - (render.nyanGun.height)/2;
```

```
    nyanShot[i].rec.width = render.nyanGun.width / 5;
```

```
    nyanShot[i].rec.height = render.nyanGun.height;
```

```
    nyanShot[i].speed.x = 5;
```

```
    nyanShot[i].speed.y = 0;
```

```
    nyanShot[i].active = false;
```

```
}
```

```
//loading sound
```

```
for (int i = 0; i < 10; i++) {
```

```
    hitSound[i].marker = LoadSound("sprite/Hit.ogg");
```

```
    hitSound[i].gunfire = LoadSound("sprite/pistol.ogg");
```

```
    SetSoundVolume(hitSound[i].marker, 1);
```

```
    SetSoundVolume(hitSound[i].gunfire, 0.1);
```

```
    hitSound[i].active = false;
```

```
    hitSound[i].gActive = false;
```

```
}
```

```
hitSound->oof = LoadSound("sprite/OOF.ogg");
```

```
}
```

```
void buyWeapons(void) {
```

```
    //MachineGun
```

```
    if (IsKeyPressed(KEY_F1) && player->money>=2000 && gun->machineGun == false )
```

```
    {
```

```
        player->money -= 2000;
```

```
        gun->machineGun = true;
```

```
        gun->machineGunAmmo = MAX_MACHINEGUN;
```

```
    }
```



```

//shotGun
if (IsKeyPressed(KEY_F2) && player->money >= 4000 && gun->shotGun == false)
{
    player->money -= 4000;
    gun->shotGun = true;
    gun->shotGunAmmo = MAX_SHOTGUN;
}

if (IsKeyPressed(KEY_F3) && player->money >= 8000 && gun->nyanGun == false)
{
    player->money -= 8000;
    gun->nyanGun = true;
    gun->nyanAmmo = MAX_NYAN_AMMO;
}

//Reload
if (IsKeyPressed(KEY_A) && player->money >= 1000 && gun->machineGunAmmo < MAX_MACHINEGUN
&&gun->machineGun)
{
    player->money -= 1000;
    gun->machineGunAmmo = MAX_MACHINEGUN;
}

if (IsKeyPressed(KEY_S) && player->money >= 1000 && gun->shotGunAmmo < MAX_SHOTGUN &&gun-
>shotGun)
{
    player->money -= 1000;
    gun->shotGunAmmo = MAX_SHOTGUN;
}

```

```
if (IsKeyPressed(KEY_D) && player->money >= 1000 && gun->nyanAmmo < MAX_NYAN_AMMO && gun->nyanGun)
```

```
{
```

```
    player->money -= 1000;
```

```
    gun->nyanAmmo = MAX_NYAN_AMMO;
```

```
}
```

```
}
```

```
void hitMarkerSound(void) {
```

```
    Vector2 mousePoint = GetMousePosition();
```

```
    //Hit marker Sound behaviour
```

```
    for (int i = 0; i < 10; i++) {
```

```
        if (IsMouseButtonPressed(MOUSE_LEFT_BUTTON) && (player->gun == 0 || (player->gun == 2 && gun->shotGunAmmo > 0))) {
```

```
            hitSound[i].gActive = true;
```

```
        }
```

```
    else if (IsMouseButtonPressed(MOUSE_LEFT_BUTTON) && player->gun == 1 && gun->machineGunAmmo > 0) {
```

```
        hitSound[i].gActive = true;
```

```
    }
```

```
    for (int j = 0; j < zombieSpawn; j++)
```

```
    {
```

```
        if (!zombie[j].active)
```

```

if (zombie[j].active && collider[j].active) {
    if (IsMouseButtonPressed(MOUSE_LEFT_BUTTON) && player->gun == 0) {
        if (CheckCollisionPointRec(mousePoint, collider[j].colliderZombs))
        {
            if (!hitSound->active)
                hitSound->active = true;
        }
    }
}

if (CheckCollisionPointRec(mousePoint, collider[j].colliderZombs) &&
IsMouseButtonDown(MOUSE_LEFT_BUTTON) && gun->machineGunAmmo > 0 && player->gun == 1)

{
    if (!hitSound->active)
        hitSound->active = true;
}

if (IsMouseButtonPressed(MOUSE_LEFT_BUTTON) && player->gun == 2 && gun->shotGunAmmo > 0)
{
    if (CheckCollisionRecs(crosshair.shotGunCross, collider[j].colliderZombs)) {
        if (!hitSound->active)
            hitSound->active = true;
    }
}

if (CheckCollisionRecs(nyanShot[j].rec, collider[j].colliderZombs))
{
    if (!hitSound->active)
        hitSound->active = true;
}

```

```
}
```

```
}
```

```
}
```

```
}
```

```
for (int i = 0; i < 100; i++) {
```

```
    if (hitSound->active) {
```

```
        PlaySound(hitSound->marker);
```

```
        hitSound->active = false;
```

```
    }
```

```
    if (hitSound[i].gActive) {
```

```
        PlaySound(hitSound[i].gunfire);
```

```
        hitSound[i].gActive = false;
```

```
    }
```

```
}
```

```
}
```

```
void Damage(void) {
```

```
    SetSoundVolume(hitSound->oof, 1.0);
```

```
    Vector2 mousePoint = GetMousePosition();
```

```
//setting damage for each gun
```

```
int damage;
```

```
switch (player->gun) {  
case 0: damage = 20; break;  
case 1: damage = 20; break;  
case 2: damage = 50; break;  
case 3: damage = 100; break;  
}
```

```
//doing damage
```

```
for (int i = 0; i < zombieSpawn; i++) {  
  
    if (zombie[i].active) {  
  
        if (CheckCollisionPointRec(mousePoint, collider[i].colliderZombs) &&  
IsMouseButtonPressed(MOUSE_LEFT_BUTTON) && player->gun == 0)  
        {  
            zombie[i].health -= damage;  
            printf("%d", zombie[i].health);  
            if (zombie[i].health <= 0 && collider[i].active) {  
                PlaySound(hitSound->oof);  
                zombie->zombieCount += 1;  
                player->money += 50;  
                collider[i].active = false;  
                zombie[i].active = false;  
            }  
        }  
    }  
}
```

```
        printf("\n%d", player->money);
    }
}
else if (player->gun == 1)
{
```

```
    if (CheckCollisionPointRec(mousePoint, collider[i].colliderZombs) &&
        IsMouseButtonDown(MOUSE_LEFT_BUTTON) && gun->machineGunAmmo > 0) {
        zombie[i].health -= damage;
        printf("%d", zombie[i].health);
```

```
    if (zombie[i].health <= 0 && collider[i].active) {
        PlaySound(hitSound->oof);
        zombie->zombieCount += 1;
        player->money += 50;
        collider[i].active = false;
        zombie[i].active = false;
        printf("\n%d", player->money);
```

```
    }
```

```
}
```

```
}
```

```

0) else if (IsMouseButtonPressed(MOUSE_LEFT_BUTTON) && player->gun == 2 && gun->shotGunAmmo >

{

    if (CheckCollisionRecs(crosshair.shotGunCross, collider[i].colliderZombs)) {

        zombie[i].health -= damage;

        printf("%d", zombie[i].health);

        if (zombie[i].health <= 0 && collider[i].active) {

            PlaySound(hitSound->oof);

            zombie->zombieCount += 1;

            player->money += 50;

            collider[i].active = false;

            zombie[i].active = false;

            printf("\n%d", player->money);

        }

    }

}

}

}

}

```

```
//NYAN GUN SPECIAL BEHAVIOUR
```

```
if (IsMouseButtonPressed(MOUSE_LEFT_BUTTON) && player->gun == 3 && gun->nyanAmmo > 0) {
```

```
    for (int i = 0; i < MAX_NYAN; i++)
```

```
    {
```

```
        if (!nyanShot[i].active)
```

```
        {
```

```
            nyanShot[i].rec.x = GetMousePosition().x - (render.nyanGun.width / 5) / 2;
```

```
            nyanShot[i].rec.y = GetMousePosition().y - (render.nyanGun.height / 2);
```

```
            nyanShot[i].active = true;
```

```
            break;
```

```
        }
```

```
    }
```

```
}
```

```
for (int i = 0; i < MAX_NYAN; i++)
```

```
{
```

```
    if (nyanShot[i].active)
```

```
    {
```

```
        nyanShot[i].rec.x -= nyanShot[i].speed.x + 3;
```

```
        for (int j = 0; j < zombieSpawn; j++) {
```

```
            if (CheckCollisionRecs(nyanShot[i].rec, collider[j].colliderZombs))
```

```
                zombie[j].health -= damage; {
```

```
                if (zombie[j].health <= 0 && collider[j].active) {
```

```
                    PlaySound(hitSound->oof);
```



```

        zombie->zombieCount += 1;

        player->money += 50;
        collider[j].active = false;
        zombie[j].active = false;

    }

}

}

if (nyanShot[i].rec.x <= 0)
{
    nyanShot[i].active = false;

}

}

}

}

void switchWeapon(void) {

    if (IsKeyPressed(KEY_Z))
        player->gun = 0;

    if (IsKeyPressed(KEY_X) && gun->machineGun) {

```

```

        player->gun = 1;
    }
    if (IsKeyPressed(KEY_C) && gun->shotGun)
        player->gun = 2;

    if (IsKeyPressed(KEY_V) && gun->nyanGun)
        player->gun = 3;
}

void playerCrossHair(void) {
    crosshair.mousePos = GetMousePosition();

    crosshair.shotGunCross.x = crosshair.mousePos.x - 100;
    crosshair.shotGunCross.y = crosshair.mousePos.y - 100;

    crosshair.shotGunCross.height = 200;
    crosshair.shotGunCross.width = 200;
}

void doNyan(void) {
    Vector2 temp = GetMousePosition();

    nyannyan->mousePos.x = temp.x - (750/5);
    nyannyan->mousePos.y = temp.y - 69;

    nyannyan->nyanCross.x = temp.x - (750/5);
    nyannyan->nyanCross.y = temp.y - 69;

    nyannyan->nyanCross.height = 138;

```

```
nyannyan->nyanCross.width = 750/5;
```

```
}
```

```
void GunSprites(void) {
```

```
    Vector2 nyanPose;
```

```
    switch (player->gun) {
```

```
    case 0: DrawTexture(render.pist, 900, 0, WHITE);
```

```
        DrawTexture(render.crossHair, crosshair.mousePos.x - 25, crosshair.mousePos.y - 25, RED);
```

```
        break;
```

```
    case 1: DrawTexture(render.ar, 900, -90, WHITE);
```

```
        DrawTexture(render.crossHair, crosshair.mousePos.x - 25, crosshair.mousePos.y - 25, RED);
```

```
        break;
```

```
    case 2: DrawTexture(render.shotGun, 900, 0, WHITE);
```

```
        DrawTexture(render.shotCross, crosshair.mousePos.x - 100, crosshair.mousePos.y - 100, WHITE);
```

```
        break;
```

```
    case 3:
```

```
        animation.nyanFramesCounter += 1;
```

```
    if (animation.nyanFramesCounter >= (60 / animation.nyanFrameSpeed))
```

```
    {
```

```
        animation.nyanFramesCounter = 0;
```

```
        animation.nyanCurrentFrame++;
```

```
        if (animation.nyanCurrentFrame > 5) animation.nyanCurrentFrame = 0;
```

```
        animation.nyanRec.x = (float)animation.nyanCurrentFrame*(float)render.nyanGun.width / 5;
```

```
    }
```

```

    DrawTextureRec(render.nyanGun, animation.nyanRec, nyannyan->mousePos, WHITE);

}

nyanPose.x = 1000;
nyanPose.y = 465;

if (player->startWave == false) {
    if (gun->machineGun) {
        DrawTexture(render.ar_b, 25, 400, WHITE);
    }
    else {
        DrawTexture(render.ar, 25, 400, WHITE);
    }
    DrawText(FormatText("F1 : Price 2000"), 25, 450, 30, RED);

    if (gun->shotGun) {
        DrawTexture(render.shotGun_b, 500, 455, WHITE);
    }
    else {
        DrawTexture(render.shotGun, 500, 455, WHITE);
    }
    if (gun->nyanGun) {
        DrawTextureRec(render.nyanGun, animation.nyanRec, nyanPose, BLACK);
    }
    else {
        DrawTextureRec(render.nyanGun, animation.nyanRec, nyanPose, WHITE);
    }

    DrawText(FormatText("A : %i/%i", gun->machineGunAmmo, MAX_MACHINEGUN), 25, 600, 30, BLUE);
    DrawText(FormatText("S : %i/%i", gun->shotGunAmmo, MAX_SHOTGUN), 500, 600, 30, BLUE);
}

```

```
DrawText(FormatText("D : %i/%i", gun->nyanAmmo, MAX_NYAN_AMMO), 1000, 600, 30, BLUE);
```

```
DrawText(FormatText("F2 : Price 4000"), 500, 450, 30, RED);
```

```
DrawText(FormatText("F3 : Price 8000"), 1000, 450, 30, RED);
```

```
DrawText(FormatText("WAVE END"), 600, 300, 60, BLUE);
```

```
DrawText(FormatText("PRESS SPACE TO START NEW WAVE"), 200, 350, 60, BLUE);
```

```
}
```

```
}
```

```
void doPlayerDamage(void) {
```

```
    for (int i = 0; i < zombieSpawn; i++)
```

```
    {
```

```
        if (zombie[i].position.x > screenWidth && zombie[i].active)
```

```
        {
```

```
            player->health -= 1;
```

```
            zombie[i].active = false;
```

```
            zombie->zombieCount += 1;
```

```
        }
```

```
    }
```

```
}
```

```
void doAnimation(void) {
```

```
    Texture2D zombie_default = LoadTexture("sprite/zombie_default_filt.png");
```

```
animation.frameRec.x = 0.0f;  
animation.frameRec.y = 0.0f;  
animation.frameRec.width = (float)zombie_default.width / 5;  
animation.frameRec.height = (float)zombie_default.height;
```

```
animation.currentFrame = 0;  
animation.framesCounter = 0;  
animation.framesSpeed = 10;
```

```
animation.nyanRec.x = 0.0f;  
animation.nyanRec.y = 0.0f;  
animation.nyanRec.width = (float)render.nyanGun.width / 5;  
animation.nyanRec.height = (float)render.nyanGun.height;
```

```
animation.nyanCurrentFrame = 0;  
animation.nyanFramesCounter = 0;  
animation.nyanFrameSpeed = 15;
```

```
UnloadTexture(zombie_default);
```

```
}
```

```
void zombieMovement(void) {  
    for (int i = 0; i < zombieSpawn; i++)  
    {  
        zombie[i].position.x += zombie[i].speed;  
    }  
}
```

```
void DecreaseAmmo(void) {
```

```

if (player->startWave) {
    if (IsMouseButtonDown(MOUSE_LEFT_BUTTON) && player->gun == 1 && gun->machineGunAmmo > 0)
    {
        gun->machineGunAmmo -= 1;
    }

    if (IsMouseButtonPressed(MOUSE_LEFT_BUTTON) && player->gun == 2 && gun->shotGunAmmo > 0)
    {
        gun->shotGunAmmo -= 1;
    }

    if (IsMouseButtonPressed(MOUSE_LEFT_BUTTON) && player->gun == 3 && gun->nyanAmmo > 0)
    {
        gun->nyanAmmo -= 1;
    }
}
}

```

```

int main(void)
{
    // Initialization
    //-----

    bool start;

    start = false;

    InitWindow(screenWidth, screenHeight, "Don't let it cross");

```

```
init();
```

```
InitAudioDevice();
```

```
//Load sound.
```

```
Sound pistolShot = LoadSound("sprite/Pistol.wav");
```

```
Sound first_screen = LoadSound("sprite/Start_music.ogg");
```

```
Sound main_music = LoadSound("sprite/main_music.ogg");
```

```
Sound machineGun_sound = LoadSound("sprite/MachineGun.ogg");
```

```
Sound shotGun = LoadSound("sprite/shotGun.ogg");
```

```
Sound nyanCat = LoadSound("sprite/Nyan.ogg");
```

```
Texture2D zombie_default = LoadTexture("sprite/zombie_default_filt.png"); //Load a ref.texture
```

```
//Animating
```

```
doAnimation();
```

```
SetTargetFPS(60); // Set our game to run at 60 frames-per-second
```

```
//-----
```

```
if(!start)
```

```
    PlaySound(first_screen);
```

```
// Main game loop
```

```
while (!WindowShouldClose()) // Detect window close button or ESC key
```



```
{  
    framesCounter++;  
  
    if (start) {  
        StopSound(first_screen);  
  
        for (int i = 0; i < MAX_NYAN; i++)  
        {  
            if (nyanShot[i].active)  
            {  
                nyanShot[i].rec.x -= nyanShot[i].speed.x+3;  
  
                nyanShot[i].position.x = nyanShot[i].rec.x;  
                nyanShot[i].position.y = nyanShot[i].rec.y;  
  
                if (nyanShot[i].rec.x + nyanShot[i].rec.width <= -screenWidth)  
                {  
                    nyanShot[i].active = false;  
                }  
            }  
        }  
    }  
}
```

```
//Playing BGM
```

```
if (!IsSoundPlaying(main_music))
```

```
    PlaySound(main_music);
```

```
//Decreasing Player's Ammo
```

```
DecreaseAmmo();
```

```
//Drawing Player's Crosshair
```

```
playerCrossHair();
```

```
//Doing NyanGun special Behaviour
```

```
doNyan();
```

```
HideCursor();
```

```
if (player->health > 0) {
```

```
    //Switching Gun Sound
```

```
    if (IsKeyDown(KEY_X) && player->gun != 1 && gun->machineGun && player->startWave)
```

```
    {
```

```
        PlaySound(machineGun_sound);
```

```
    }
```

```
    if (IsKeyDown(KEY_C) && player->gun != 2 && gun->shotGun && player->startWave)
```

```
    {
```

```
        PlaySound(shotGun);
```

```
    }
```

```
    if (player->gun == 3)
```

```
    {
```

```
    if (IsSoundPlaying(main_music))
        StopSound(main_music);

    if(!IsSoundPlaying(nyanCat))
        PlaySound(nyanCat);
}
else {
    StopSound(nyanCat);
}

//NEW WAVE BEHAVIOUR
if (zombie->zombieCount == zombieSpawn)
{

    player->startWave = false;
    buyWeapons();
    initZombie();

    if (IsKeyPressed(KEY_SPACE))
    {
        player->wave += 1;
        zombieSpeed += 0.2;
        zombieSpawn += 20;
        zombie->zombieCount = 0;
        player->startWave = true;
    }

}
```

```
// Update
```

```
//-----
```

```
//Background changing
```

```
if (player->wave % 2 == 0) {
```

```
    render.bg = render.bg_n;
```

```
}
```

```
else
```

```
    render.bg = render.bg_s;
```

```
animation.framesCounter += 1;
```

```
//Drawing collision box
```

```
if (player->startWave) {
```

```
    colliderZombies();
```

```
//updating animation frame
```

```
if (animation.framesCounter >= (60 / animation.framesSpeed))
```

```
{
```

```
    animation.framesCounter = 0;
```

```
    animation.currentFrame++;
```

```
if (animation.currentFrame > 5) animation.currentFrame = 0;
```

```
animation.frameRec.x = (float)animation.currentFrame*(float)zombie_default.width / 5;
```

```
}
```

```
//player buy weapon
```

```
//player switch weapon
```

```
switchWeapon();
```

```
//doDamage
```

```
Damage();
```

```
//doing Sound
```

```
hitMarkerSound();
```

```
//do player damage
```

```
doPlayerDamage();
```

```
//Zombie movement
```

```
zombieMovement();
```

```
if (animation.framesSpeed > MAX_FRAME_SPEED) animation.framesSpeed = MAX_FRAME_SPEED;
```

```
else if (animation.framesSpeed < MIN_FRAME_SPEED) animation.framesSpeed =  
MIN_FRAME_SPEED;
```

```
//-----
```

```
}
```

```

// Draw
//-----

BeginDrawing();

ClearBackground(WHITE);


DrawTexture(render.bg, 0, 0, WHITE); // Draw Background


if (player->startWave) {

    for (int i = 0; i < zombieSpawn; i++) {

        if (IsKeyDown(KEY_F) && collider[i].active)

            DrawRectangle(zombie[i].position.x, zombie[i].position.y, zombie_default.width / 5,
zombie_default.height, RED);

        if (zombie[i].active)

            DrawTextureRec(zombie_default, animation.frameRec, zombie[i].position, WHITE); // Draw part
of the texture (Animation)

    }

}

```

```

    for (int i = 0; i <= MAX_NYAN; i++) {
        if (nyanShot[i].active) {
            DrawTextureRec(render.nyanGun, animation.nyanRec, nyanShot[i].position, WHITE);
        }
    }

    draw(); //Draw text

    //switch gun sprites
    GunSprites();

    EndDrawing();
    //-----
}

//Restarting the game
else {

    BeginDrawing();

    DrawTexture(render.go, 0, 0, WHITE);
    if (IsKeyPressed(KEY_R))
        init();
    EndDrawing();

}

}

```

```

//Starting game
else {

    if (IsKeyPressed(KEY_SPACE)) {
        start = true;
    }

    BeginDrawing();

    DrawTexture(render.startScreen, -155, 0, WHITE);
    EndDrawing();
}
}

```

// De-Initialization

//-----

```

UnloadTexture(zombie_default);
UnloadTexture(render.pist);
UnloadTexture(render.ar); // Texture unloading
UnloadTexture(render.crossHair);
UnloadTexture(render.bg_s);
UnloadTexture(render.bg_n);
UnloadTexture(render.shotCross);
UnloadTexture(render.shotGun);
UnloadTexture(render.ar_b);
UnloadTexture(render.shotGun_b);
UnloadTexture(render.nyanGun);

```



```
UnloadTexture(render.go);
```

```
UnloadSound(pistolShot);
```

```
UnloadSound(first_screen);
```

```
UnloadSound(main_music);
```

```
UnloadSound(machineGun_sound);
```

```
UnloadSound(shotGun);
```

```
UnloadSound(nyanCat);
```

```
UnloadSound(hitSound->oof);
```

```
//looping to unload all the marker sounds made
```

```
for(int i = 0 ; i < 10 ; i++)
```

```
    UnloadSound(hitSound[i].marker);
```

```
CloseAudioDevice();
```

```
CloseWindow();          // Close window and OpenGL context
```

```
//-----
```

```
return 0;
```

```
}
```

