```c
#include "raylib.h"
#include <math.h>
int main()
{
   InitWindow(800, 600, "TERRAIN");
   SetTargetFPS(60);

   int terrainheights[40];
   for (int i = 0; i < 40; i++)
   {
       terrainheights[i] = GetRandomValue(50, 100);
   }
   // Cannons
   int cannon1col = GetRandomValue(3, 10);
   int cannon1X = cannon1col * 20;
   int cannon1Y = (600 - terrainheights[cannon1col]) - 15;

   int cannon2col = GetRandomValue(30, 37);
   int cannon2X = cannon2col * 20;
   int cannon2Y = (600 - terrainheights[cannon2col]) - 15;
   // angles
   int cannon1Angle = -45.0f;
   int cannon2Angle = -135.0f;
   int currentPlayer = 1;
   int shooter = 0;
   // POWER hardcoded
   int Power1 = 500;
   int Power2 = 500;
   // projectile
   float Velocity1X = Power1 * cosf(cannon1Angle * DEG2RAD);
   float Velocity1Y = Power1 * sinf(cannon1Angle * DEG2RAD);

   int Health1 = 100;
   int Health2 = 100;
   int projX = cannon1X;
   int projY = cannon1Y;
   float time = 0;
   int projFlying = 0;

   int gameOver = 0;
```

```
int winner = 0;
while (!WindowShouldClose())
{
    if (currentPlayer == 1)
    {
        if (IsKeyDown(KEY_W))
            cannon1Angle -= 5;
        if (IsKeyDown(KEY_S))
            cannon1Angle += 5;
        if (IsKeyDown(KEY_A))
            Power1 -= 5;
        if (IsKeyDown(KEY_D))
            Power1 += 5;
        if (Power1 > 1000)
            Power1 = 1000;
        if (Power1 < 200)
            Power1 = 200;
        if (cannon1Angle < -90)
            cannon1Angle = -90;
        if (cannon1Angle > 0)
            cannon1Angle = 0;
    }
    else
    {
        if (IsKeyDown(KEY_UP))
            cannon2Angle -= 5;
        if (IsKeyDown(KEY_DOWN))
            cannon2Angle += 5;
        if (IsKeyDown(KEY_A))
            Power2 -= 5;
        if (IsKeyDown(KEY_D))
            Power2 += 5;

        if (Power2 > 1000)
            Power2 = 1000;
        if (Power2 < 200)
            Power2 = 200;
        if (cannon2Angle < -180)
            cannon2Angle = -180;
        if (cannon2Angle > -90)
```

```
            cannon2Angle = -90;
        }

        if (IsKeyPressed(KEY_F) && !projFlying)
        {
            if (currentPlayer == 1)
            {
                projX = cannon1X;
                projY = cannon1Y;
                Velocity1X = Power1 * cosf(cannon1Angle * DEG2RAD);
                Velocity1Y = Power1 * sinf(cannon1Angle * DEG2RAD);
            }
            else
            {
                projX = cannon2X;
                projY = cannon2Y;
                Velocity1X = Power2 * cosf(cannon2Angle * DEG2RAD);
                Velocity1Y = Power2 * sinf(cannon2Angle * DEG2RAD);
            }
            projFlying = 1;
            time = 0;
            shooter = currentPlayer;
        }

        // if (currentPlayer == 2) {
        //     if (IsKeyDown(KEY_UP)) cannon2Angle -= 5;
        //     if (IsKeyDown(KEY_DOWN)) cannon2Angle += 5;
        //     if(IsKeyDown(KEY_A)) Power2-=5;
        //     if(IsKeyDown(KEY_D)) Power2+=5;

        //       if(Power2>1000) Power2=1000;
        //     if(Power2<200) Power2=200;
        //     if (cannon2Angle < -180) cannon2Angle = -180;
        //     if (cannon2Angle > -90) cannon2Angle = -90;
        // }
        if (gameOver && IsKeyPressed(KEY_R))
        {
            Health1 = 100;
            Health2 = 100;
            currentPlayer = 1;
```

```
            gameOver = 0;
            projFlying = 0;

            for (int i = 0; i < 40; i++)
            {
                terrainheights[i] = GetRandomValue(50, 100);
            }

            cannon1col = GetRandomValue(3, 10);
            cannon1X = cannon1col * 20;
            cannon1Y = (600 - terrainheights[cannon1col]) - 15;

            cannon2col = GetRandomValue(30, 37);
            cannon2X = cannon2col * 20;
            cannon2Y = (600 - terrainheights[cannon2col]) - 15;
            projX = cannon1X;
            projY = cannon1Y;
        }

        if (IsKeyPressed(KEY_R) && !projFlying)
        {
            for (int i = 0; i < 40; i++)
            {
                terrainheights[i] = GetRandomValue(50, 100);
            }

            cannon1col = GetRandomValue(3, 10);
            cannon1X = cannon1col * 20;
            cannon1Y = (600 - terrainheights[cannon1col]) - 15;

            cannon2col = GetRandomValue(30, 37);
            cannon2X = cannon2col * 20;
            cannon2Y = (600 - terrainheights[cannon2col]) - 15;
            projX = cannon1X;
            projY = cannon1Y;
        }
        BeginDrawing();
        ClearBackground(SKYBLUE);
        DrawRectangle(50, 80, 80, 30, WHITE);
        DrawRectangle(520, 70, 60, 25, WHITE);
```

```
        DrawRectangle(200, 120, 100, 35, WHITE);
        DrawRectangle(230, 110, 70, 30, WHITE);
        DrawRectangle(500, 60, 90, 30, WHITE);
        DrawRectangle(520, 50, 60, 25, WHITE);
        DrawCircle(750, 50, 70, YELLOW);
        if (projFlying)
        {
            time += GetFrameTime();
            int gravity = 300;
            projX = (currentPlayer == 1 ? cannon1X : cannon2X) + Velocity1X
* time;
            projY = (currentPlayer == 1 ? cannon1Y : cannon2Y) + Velocity1Y
* time + 0.5 * gravity * time * time;

            if (projY >= 600 || projX < 0 || projX >= 800)
            {
                projFlying = 0;
                if (currentPlayer == 1)
                {
                    currentPlayer = 2;
                }
                else
                {
                    currentPlayer = 1;
                }
            }
            if (CheckCollisionCircles((Vector2){projX, projY}, 5,
(Vector2){cannon1X, cannon1Y}, 15) && shooter != 1)
            {
                Health1 -= 20;
                projFlying = 0;
                if(Health1<0) Health1=0;

                if (currentPlayer == 1)
                {
                    currentPlayer = 2;
                }
                else
                {
                    currentPlayer = 1;
```

```
            }
        }
        if (CheckCollisionCircles((Vector2){projX, projY}, 5,
(Vector2){cannon2X, cannon2Y}, 15) && shooter != 2)
        {
            Health2 -= 20;
            projFlying = 0;
            if(Health2<0) Health2=0;
            if (currentPlayer == 1)
            {
                currentPlayer = 2;
            }
            else
            {
                currentPlayer = 1;
            }
        };
    }

    for (int i = 0; i < 40; i++)
    {
        Rectangle tRect = {i * 20, 600 - terrainheights[i], 20,
terrainheights[i]};
        if (CheckCollisionPointRec((Vector2){projX, projY}, tRect))
        {
            projFlying = 0;
            if (shooter == 1)
            {
                currentPlayer = 2;
            }
            else
            {
                currentPlayer = 1;
            }
            break;
        }
    }

    if (Health1 <= 0)
    {
```

```
            gameOver = 1;
            winner = 2;
        }
        if (Health2 <= 0)
        {
            gameOver = 1;
            winner = 1;
        }

        for (int i = 0; i < 40; i++)
        {
            int x = i * 20;
            int y = 600 - terrainheights[i];
            int height = terrainheights[i];

            DrawRectangle(x, y, 20, height, DARKGREEN);
        }

        DrawCircle(cannon1X, cannon1Y, 15, RED);
        DrawCircle(cannon2X, cannon2Y, 15, BLUE);
        float barrelLength = 25;
        float barrel1EndX = cannon1X + cosf(cannon1Angle * DEG2RAD) *
barrelLength;
        float barrel1EndY = cannon1Y + sinf(cannon1Angle * DEG2RAD) *
barrelLength;
        DrawLineEx(
            (Vector2){cannon1X, cannon1Y},
            (Vector2){barrel1EndX, barrel1EndY},
            4,
            DARKGRAY);

        float barrel2EndX = cannon2X + cosf(cannon2Angle * DEG2RAD) *
barrelLength;
        float barrel2EndY = cannon2Y + sinf(cannon2Angle * DEG2RAD) *
barrelLength;
        DrawLineEx(
            (Vector2){cannon2X, cannon2Y},
            (Vector2){barrel2EndX, barrel2EndY},
            4,
            DARKGRAY);
```

```
        if (currentPlayer == 1)
        {
            DrawText("PLAYER 1'S TURN (W/S to aim)", 10, 10, 20, WHITE);
        }
        else
        {
            DrawText("PLAYER 2'S TURN (UP/DOWN to aim)", 10, 10, 20,
WHITE);
        }
        DrawCircle(projX, projY, 5, BLACK);

        DrawText(TextFormat("P1  %d", Power1), 10, 40, 20, RED);
        DrawText(TextFormat("Health 1 %d", Health1), 10, 70, 20, RED);
        DrawText(TextFormat("P2  %d", Power2), 600, 40, 20, BLUE);
        DrawText(TextFormat("Health 2 %d", Health2), 600, 70, 20, BLUE);
        DrawText("Press R to regenerate terrain", 250, 570, 20, WHITE);

        if (gameOver)
        {
            DrawRectangle(0, 0, 800, 600, (Color){0, 0, 0, 150});

            if (winner == 1)
            {
                DrawText("PLAYER 1 WINS!", 200, 250, 50, RED);
            }
            else
            {
                DrawText("PLAYER 2 WINS!", 200, 250, 50, BLUE);
            }

            DrawText("Press R to Restart", 250, 320, 30, WHITE);
        }

        EndDrawing();
    }

    CloseWindow();
    return 0;
}
```