

МИРЭА – Российский технологический университет

Институт искусственного интеллекта

Симуляция боевых действий с участием нескольких сторон

Выполнил: ст. гр. КМБО-04-18. Валяев Н. А.

Руководитель: старший преподаватель Митин А.В

Москва

2022

Цель Работы

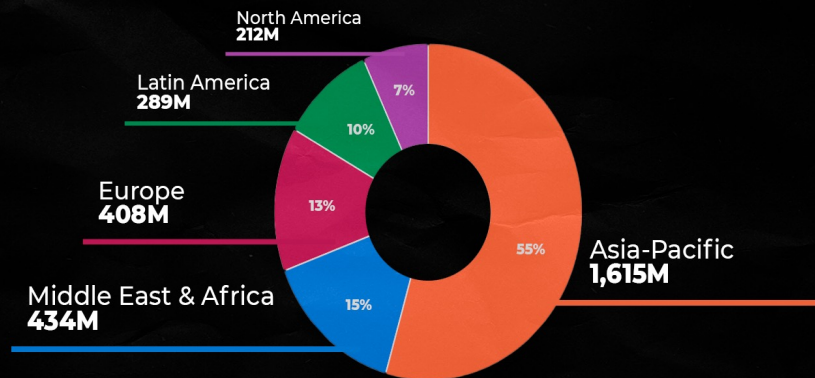
Разработать прикладные программы, реализующие гибкую физическую среду игрового взаимодействия.

- Правила игровой среды
- Подконтрольные пользователю игровые персонажи
- Генерация разнообразного ландшафта

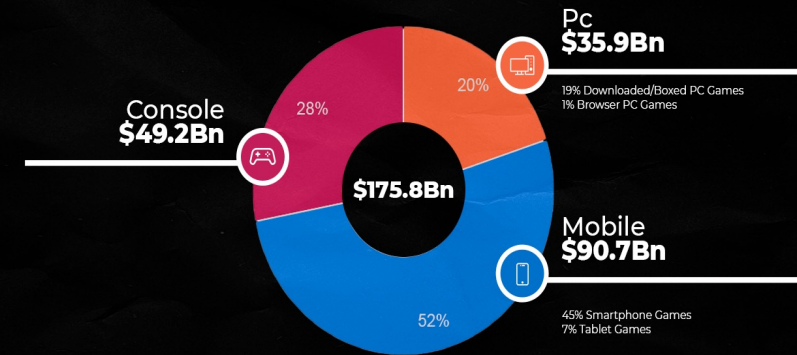
Актуальность работы

- Видеоигры в настоящее время являются весомой частью индустрии развлечений, наряду с кино, музыкой и литературой.

2021 Global Games Players



2021 Global Games Market



Скрипты Unity

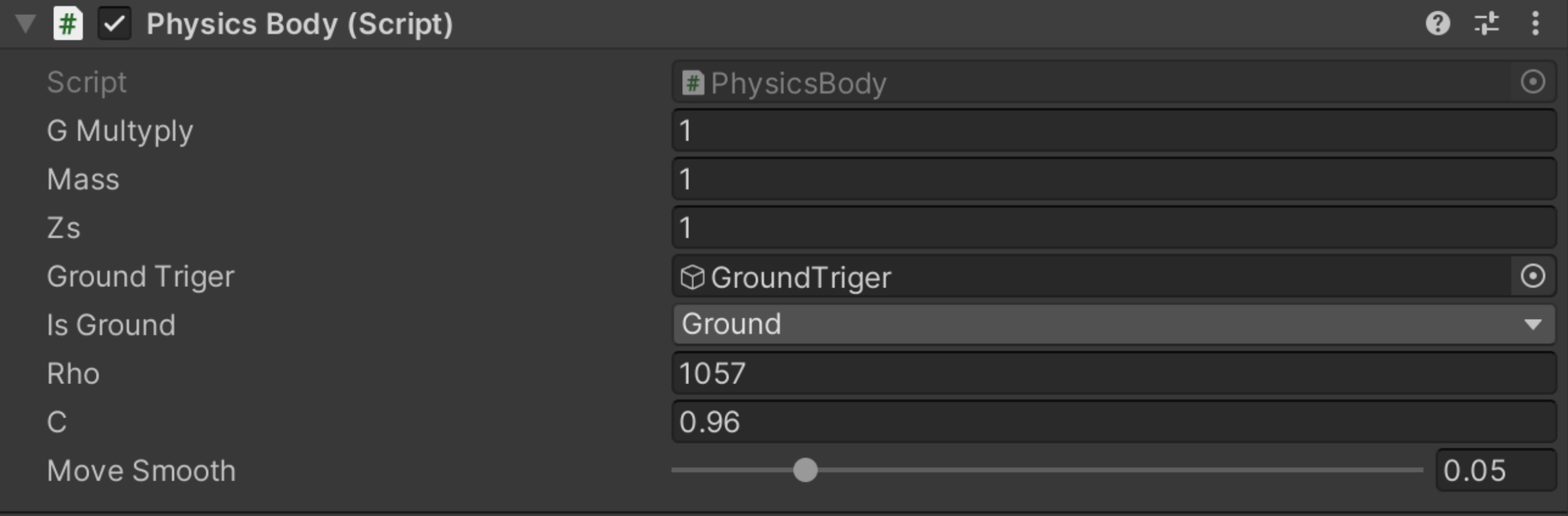
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Script : MonoBehaviour
{
    void Start()
    {
    }

    private void Awake()
    {
    }

    void Update()
    {
    }

    private void FixedUpdate()
    {
    }
}
```



Физическое тело

- Масса
- Радиус цилиндра
- Плотность
- Коэффициент лобового сопротивления

```

public void throwHT()
{
    var k1 = 6 * Math.PI * Environment.Instance.Mu * zs; //Стокс
    var k2 = c * 0.5 * Math.PI * zs * zs * Environment.Instance.Rho;
    var m_ = (rho - Environment.Instance.Rho) * mass / rho;
    Vector3 v = new Vector3((float)(RigidBody.velocity.x + throwSpeed(mass, m_, k1, k2, RigidBody.velocity.x, RigidBody.velocity.y, 0, Time.fixedDeltaTime)),
        (float)(RigidBody.velocity.y + throwSpeed(mass, m_, k1, k2, RigidBody.velocity.y, RigidBody.velocity.x, PhysicsManager.Instance.FreeFallAcs, Time.fixedDeltaTime)),
        RigidBody.velocity.z);
    RigidBody.velocity = Vector3.SmoothDamp(RigidBody.velocity, v, ref V, MoveSmooth);
}

private static double throwSpeed(double m, double m_, double k1, double k2, double v1, double v2, double a, double dt)
{
    var part1 = (-m_ * a - FSoprt(m, m_, k1, k2, v1, v2)) / m;
    var insidespeed1 = v1 + dt * (-m_ * a - FSoprt(m, m_, k1, k2, v1, v2)) / m;
    var insidespeed2 = v2 + dt * (-m_ * a - FSoprt(m, m_, k1, k2, v2, v1)) / m;
    var part2 = (-m_ * a - FSoprt(m, m_, k1, k2, insidespeed1, insidespeed2)) / m;
    return (part1 + part2) * dt / 2;
}

private static double FSoprt(double m, double m_, double k1, double k2, double v1, double v2)
{
    return (k1 + k2 * Math.Sqrt(v1 * v1 + v2 * v2)) * v1;
}

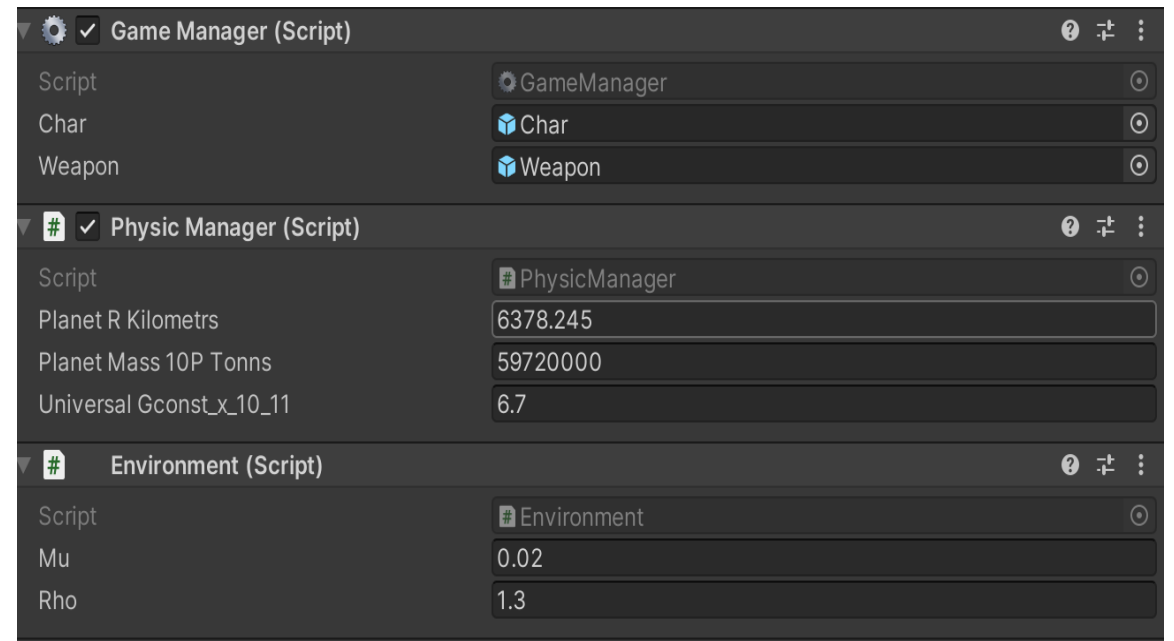
```

Прикладываемые силы

- Сила тяжести $F_{\text{тяж}} = m * g$
- Сила Архимеда $F_{\text{арх}} = \rho * g * V$
- Сила Сопротивления среды $F_{\text{сопр}} = (k1 + k2 * \sqrt{v_1^2 + v_2^2}) * v_1$
 - $k1 = 6 * \pi * \mu * r$
 - $k2 = 0.5 * \pi * r^2 * \rho$
- Закон всемирного тяготения $g = G * \frac{M}{R^2}$

Параметры среды

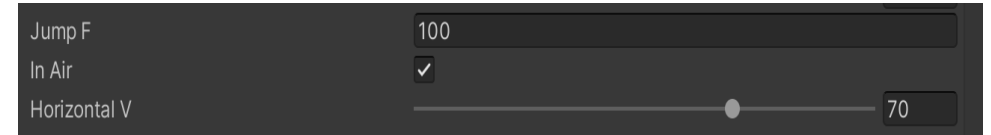
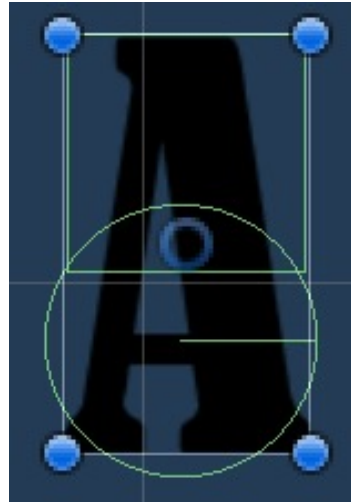
- Шаблон персонажей
- Радиус «Планеты»
- Масса
- Универсальная Гравитационная постоянная
- Вязкость среды
- Плотность среды



Игровые персонажи

Физическое тело +....

- Движение горизонтально
- Движение вертикально
- Сила реакции опоры



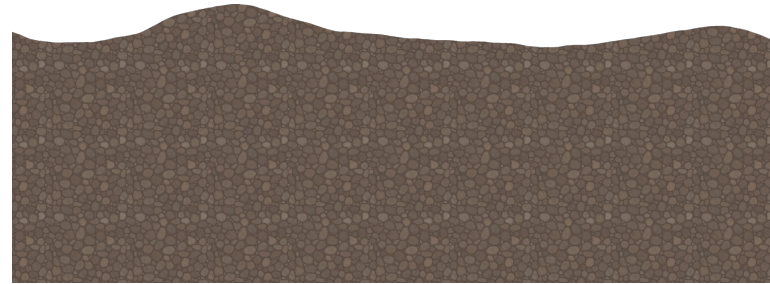
```
public void Move(float move, bool jump)
{
    if (onGround || InAir)
    {
        Vector3 targetVelocity = new Vector2(move*HorizontalV, Rigidbody.velocity.y);
        Rigidbody.velocity = Vector3.SmoothDamp(Rigidbody.velocity, targetVelocity, ref V, MoveSmooth);

        if (move > 0 && !FacingRL)
        {
            Flip();
        }
        else if (move < 0 && FacingRL)
        {
            Flip();
        }
    }
    if (onGround && jump)
    {
        onGround = false;
        Rigidbody.AddForce(new Vector2(0f, jumpF));
    }
}

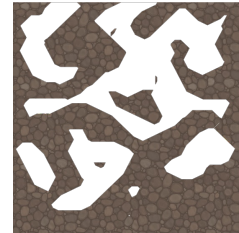
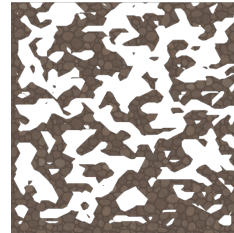
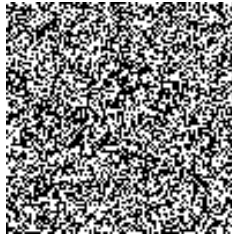
protected void Flip()
{
    FacingRL = !FacingRL;
    Vector3 theScale = transform.localScale;
    theScale.x *= -1;
    transform.localScale = theScale;
}
```


Генерация Ландшафта

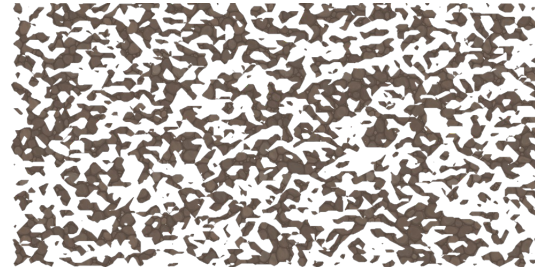
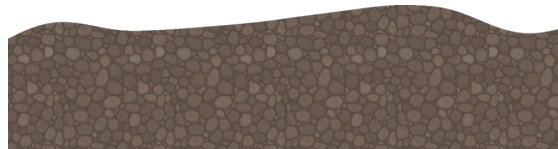
- Шум Перлина



- Клеточный Автомат

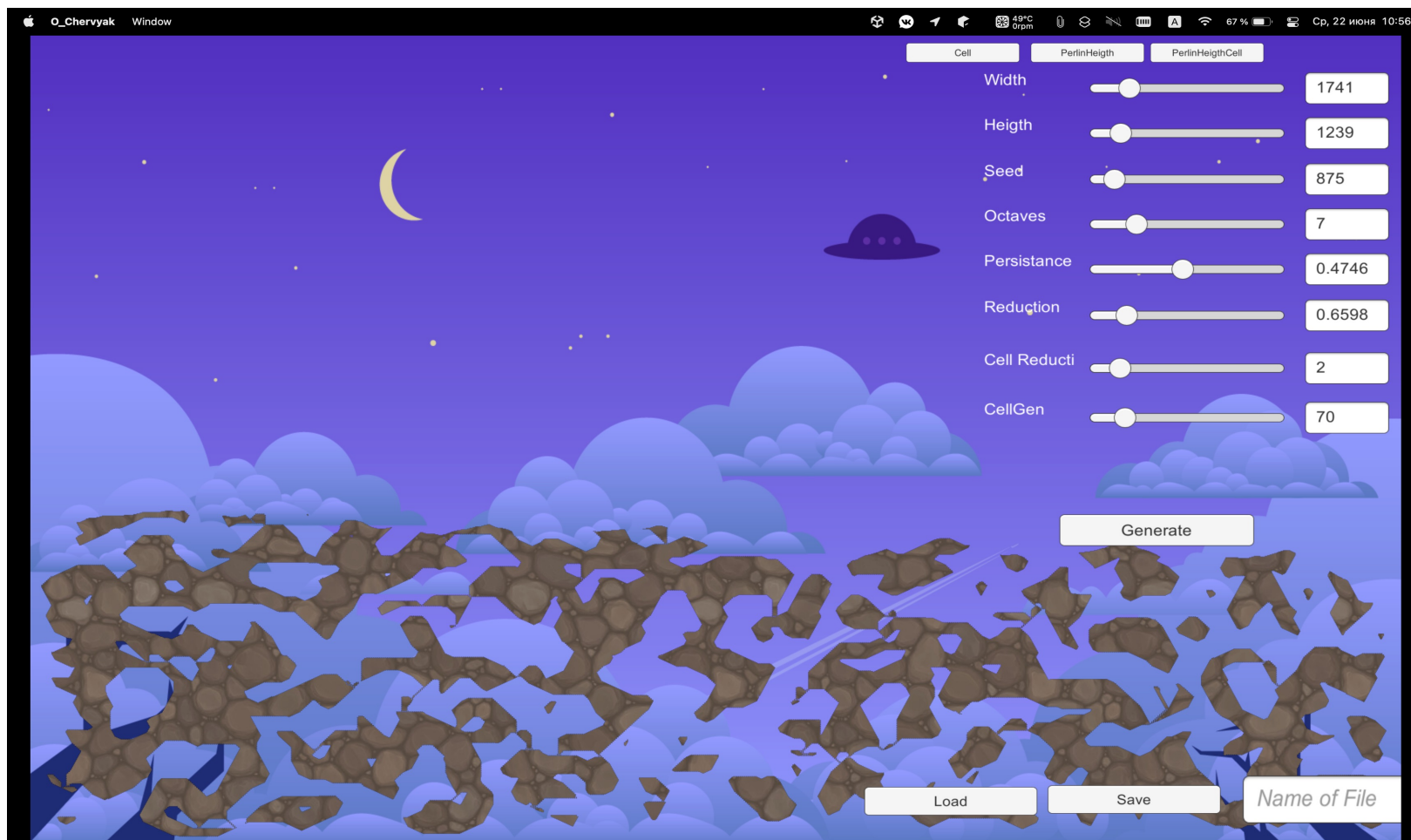


- Объединение



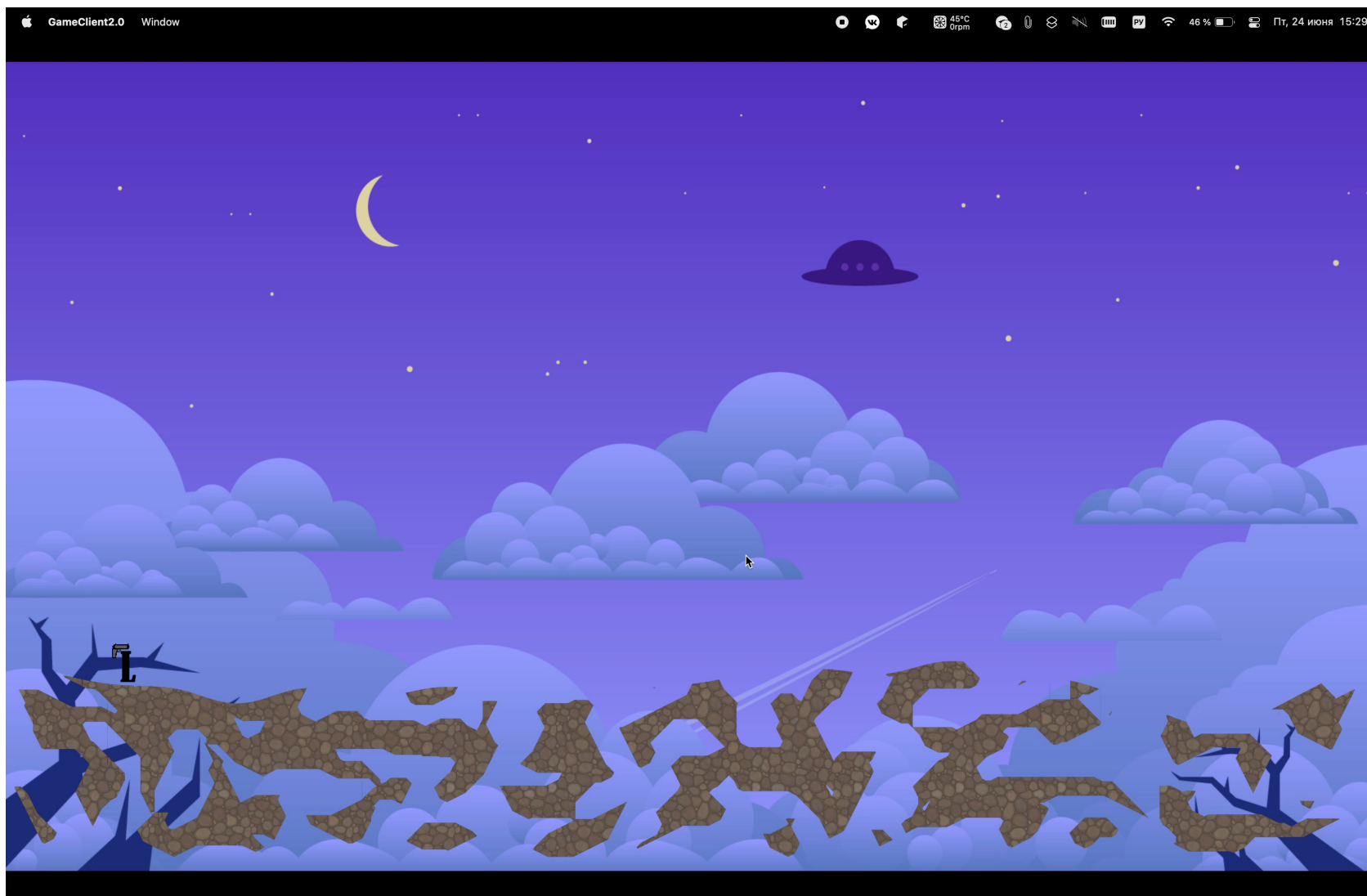
Результаты

Генератор ландшафта



Результат

Симуляция среды



Спасибо за внимание