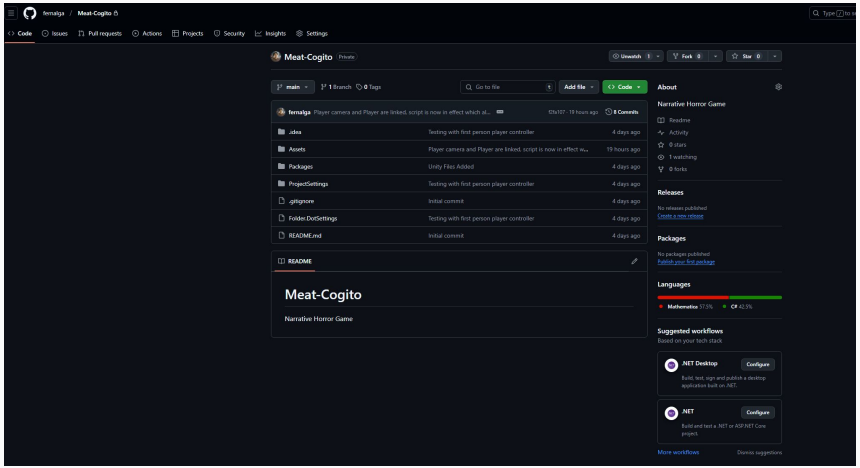
Meat Cogito

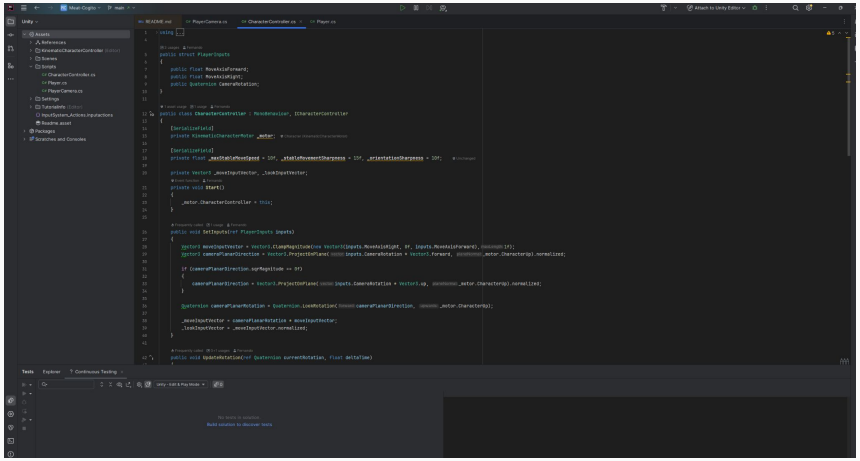
A Narrative Horror Game

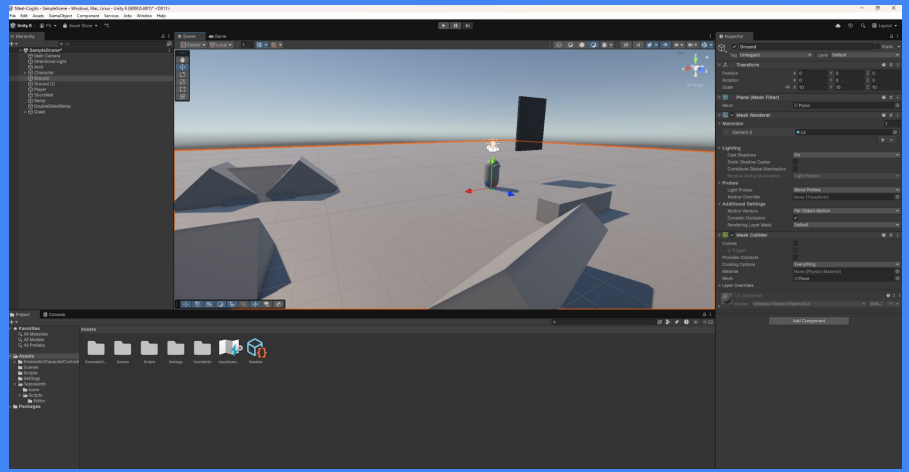
Fernando Villasenor

Rider & Unity + Github

- Rider Scripts
- Unity Game Engine
- GitHub Repo


fernalga / Meat-Cogito
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
Meat-Cogito
Private
main
Add file
Code
About
Narrative Horror Game
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
Mathematica 57.5%
C# 42.5%
Suggested workflows
Based on your tech stack
.NET Desktop
Configure
Build, test, sign and publish a desktop application built on .NET.
.NET
Configure
Build and test a .NET or ASP.NET Core project.
More workflows
Dismiss suggestions
.idea
Assets
Packages
ProjectSettings
.gitignore
Folder.DotSettings
README.md
README
Meat-Cogito
Narrative Horror Game





Player Controller

- Player Camera
- Character Controller
- Player

README.md | PlayerCamera.cs | CharacterController.cs | Player.cs

```cs
using UnityEngine;

1 asset usage  Fernando
public class Player : MonoBehaviour
{
    [SerializeField]
    private PlayerCamera _playerCamera;  Main Camera (PlayerCamera)
    [SerializeField]
    private Transform _cameraFollowPoint;  CameraTarget (Transform)
    [SerializeField]
    private CharacterController _characterController;  Character (CharacterController)

    private Vector3 _lookInputVector;

    Event function  Fernando
    private void Start()
    {
        _playerCamera.SetFollowTransform(_cameraFollowPoint);
    }

    Frequently called  1 usage  Fernando
    private void HandleCameraInput()
    {
        float mouseUp = Input.GetAxisRaw("Mouse Y");
        float mouseRight = Input.GetAxisRaw("Mouse X");

        _lookInputVector = new Vector3(mouseRight, mouseUp, 0f);

        float scrollInput = -Input.GetAxis("Mouse ScrollWheel");
        _playerCamera.UpdateWithInput(Time.deltaTime, scrollInput, _lookInputVector);
    }

    Frequently called  1 usage  Fernando
    private void HandleCharacterInputs()
    {
        PlayerInputs inputs = new PlayerInputs();
        inputs.MoveAxisForward = Input.GetAxisRaw("Vertical");
        inputs.MoveAxisRight = Input.GetAxisRaw("Horizontal");
        inputs.CameraRotation = _playerCamera.transform.rotation;

        _characterController.SetInputs(ref inputs);
    }

    Event function  Fernando
    private void Update()
```

README.md | PlayerCamera.cs | CharacterController.cs | Player.cs

```cs
using ...

1 asset usage  1 usage  Fernando
public class PlayerCamera : MonoBehaviour
{
    [SerializeField] // variables are editable in unity inspector but remain private
    // Camera distance settings
    private float _defaultDistance = 0f,  Unchanged
        _minDistance = 0f,  Unchanged
        _maxDistance = 10f,  Unchanged
        _distanceMovementSpeed = 5f,  Unchanged
        _distanceMovementSharpness = 10f,  Unchanged

        // Rotation settings
        _rotationSpeed = 10f,  Unchanged
        _rotationSharpness = 10000f,  Unchanged

        // Follow smoothing
        _followSharpness = 10000f,  Unchanged

        // Vertical angle limits (camera up/down)
        _minVerticalAngle = -90f,  Unchanged
        _maxVerticalAngle = 90f,  Unchanged
        _defaultVerticalAngle = 20f;  Unchanged

    // Transform references
    private Transform _followTransform;
    private Vector3 _currentFollowPosition, _planarDirection;

    // Distance and angle tracking
    private float _targetVerticalAngle;
    private float _currentDistance, _targetDistance;

    // resets character camera angle/distance/direction back to default on start up
    Event function  Fernando
    private void Awake()
    {
        _currentDistance = _defaultDistance;
        _targetDistance = _currentDistance;
        _targetVerticalAngle = 0f;
        _planarDirection = Vector3.forward;
    }

    // Bridges camera,input, and character controller
    1 usage  Fernando
```

Week1: Pre-Production

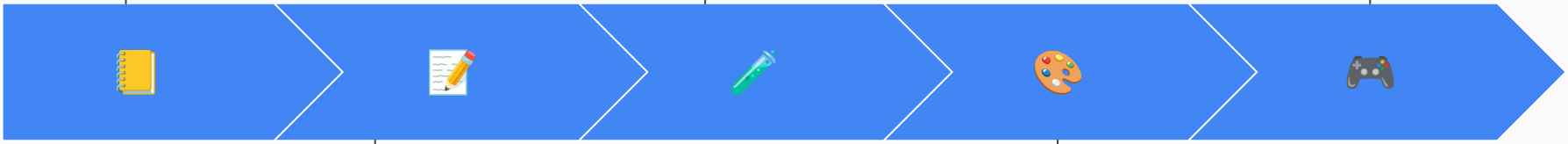
~~Research~~
~~Finalize game concept~~
~~Design document~~

Week 3-4: Prototype Development

~~Build movement~~ and interaction mechanics
~~Implement placeholder assets~~
Develop core horror mechanics

Week 11: Playtesting & Iteration

Conduct playtesting
Gather feedback
Bug fixing



Week 2: Narrative

~~Create & finalize narrative~~

Week 5-10: Asset Creation and Refinement

Design environments
Design characters
Design UI elements
Implement sound design & music
Refine animations & visual effects