ktu
1922

KAUNAS UNIVERSITY OF TECHNOLOGY

FACULTY OF INFORMATICS

T120B166 Development of Computer Games and Interactive Applications

In search of Light

IFF-6/8, Tadas Laurinaitis:

Date: 2019.*03.12*

Kaunas, 2019

Tables of Contents

Tables of Images

Table of Tables/functions

Work Distribution Table:

Name/Surname	Description of game development part
Tadas Laurinaitis	*Did everything.*

Description of Your Game

Description of Your Game.

1. **3D or 2D?** *3D.*
2. What type is your game? *A 3rd person adventure RPG.*
3. What genre is your game? *A 3rd person adventure RPG.*
4. Platforms (mobile, PC or both?) *PC.*
5. Scenario Description. *A spaceship crashes on a distant unknown planet on which the people are still living in the medieval times. The Survivor has to explore the planet, meet the locals, aid them in their quests and find a way to get back to his own home planet.*

Laboratory work #1

List of tasks (main functionality of your project)

1. Implement simple player movement #1
2. Implement 3rd person camera #2
3. Make a temporary map #3

Solution

Task #1. *Implementation of player movement*

Description of the implementation (3-5 sentences). *Firstly, I created a simple box, which will be used as a temporary player. Secondly, I attached Box collider and RigidBody components to that box. Lastly, I wrote a c# script, which makes the player move, when WASD or Arrow keys are pressed.*
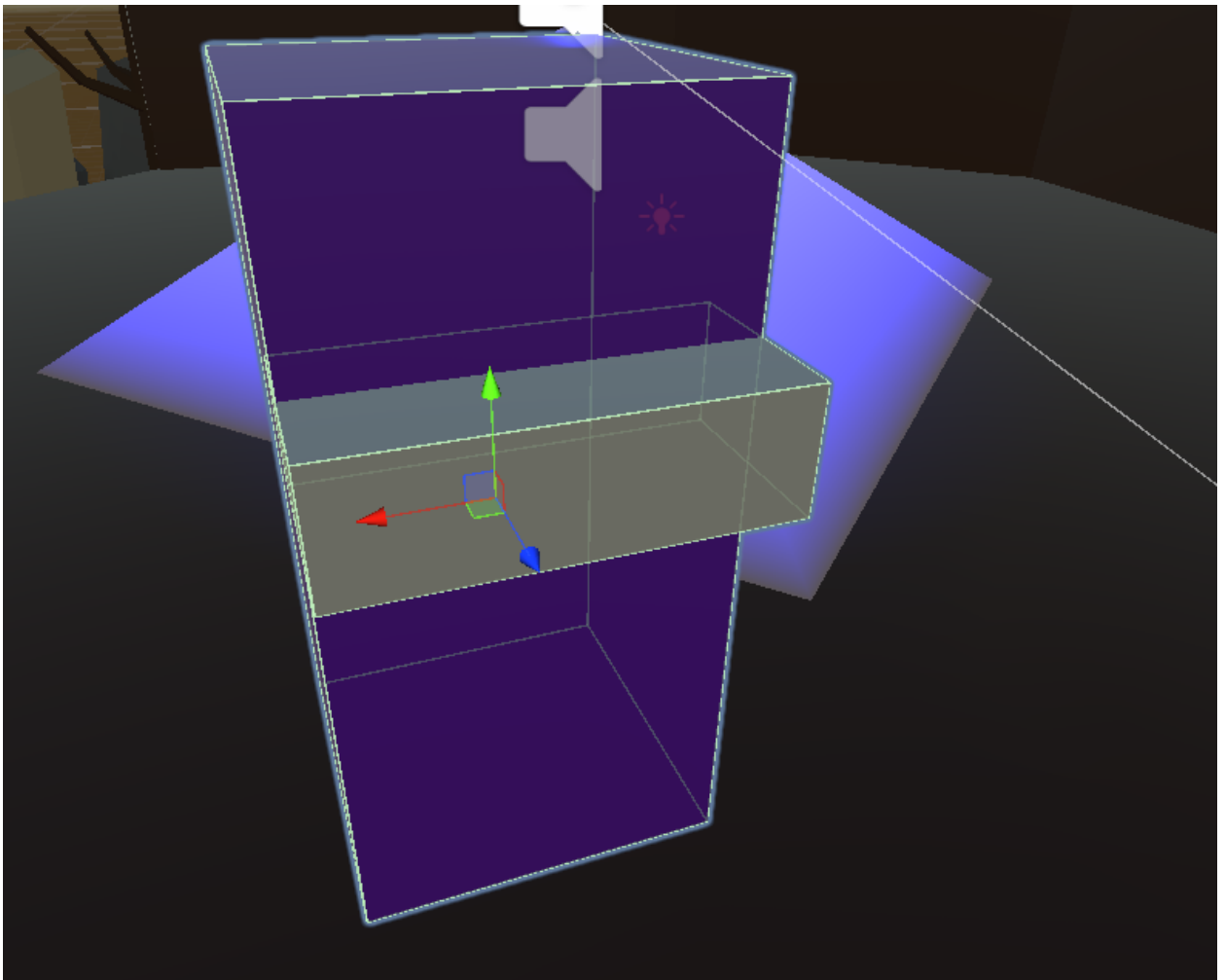
Figure 1. Screenshot #1

In the case of using functions, the description of each main function should be completed with the source code FRAGMENTS (the functions should be indexed in a separate table of contents) [3];

```csharp
    Camera cam;
    public float Speed = 10f;
    public float Gravity = -5f;

    // Start is called before the first frame update
    void Start()
    {
        cam = Camera.main;
    }

    // Update is called once per frame
    void Update()
    {
        MovePlayer();
        //transform.Translate(Input.GetAxis("Horizontal")*Time.deltaTime*speed, 0f,
Input.GetAxis("Vertical") * Time.deltaTime * speed);
        //Rigidbody rigidBody = GetComponent<Rigidbody>();

        //if (Input.GetKeyDown(KeyCode.A))
        //{

        //}

    }

    void MovePlayer()
    {
        float horizontal = Input.GetAxis("Horizontal");
        float vertical = Input.GetAxis("Vertical");
        Vector3 playerMovement = new Vector3(horizontal, 0f, vertical) * Time.deltaTime *
Speed;
        transform.Translate(playerMovement, Space.Self);
    }
```

Table 1. Script, responsible for player movement #1

Task #2. *Implementation of 3ʳᵈ person camera*

Description of the implementation (3-5 sentences). *Firstly I positioned main camera right behind player, so we can fully see him. Secondly, I attached main camera to player. Lastly, I wrote a script, which makes camera and player to turn when it gets input from mouse, and only turn camera when Left Shift button is pressed.*
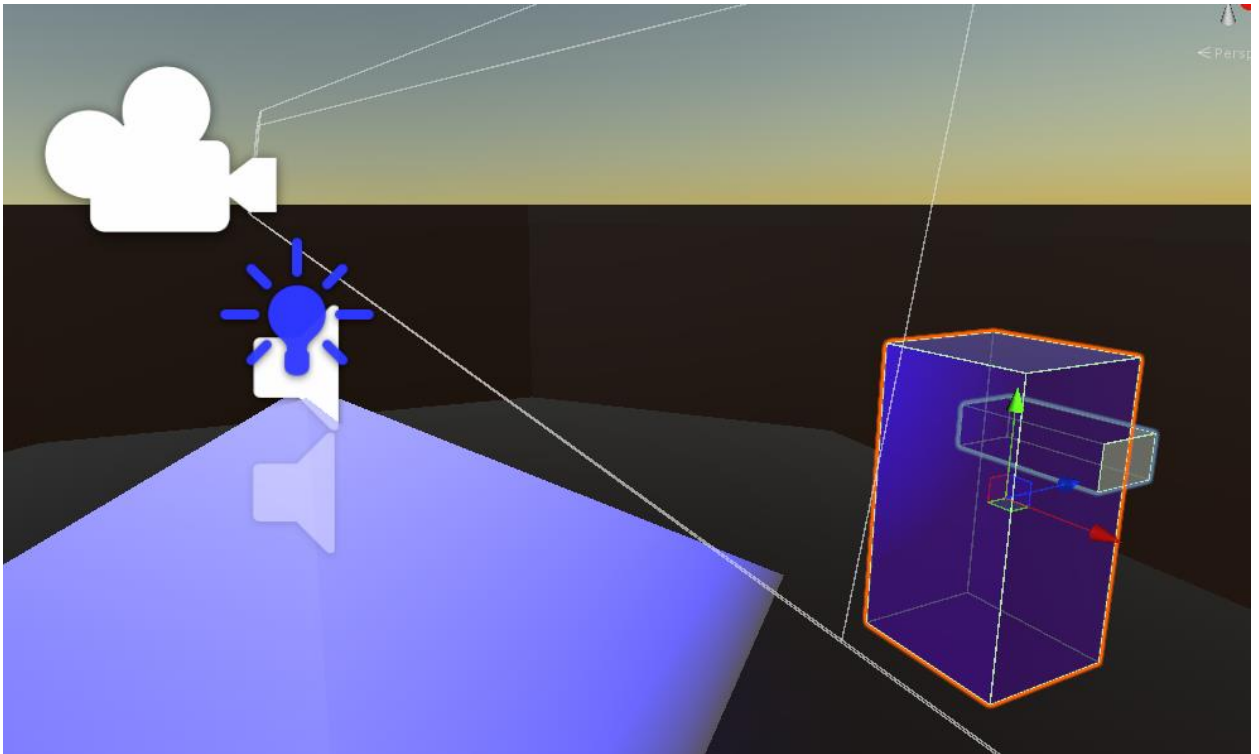
Figure 2. Screenshot #2

In the case of using functions, the description of each main function should be completed with the source code FRAGMENTS (the functions should be indexed in a separate table of contents);

```csharp
public float RotationSpeed = 1f;
public Transform Target;
public Transform Player;
float mouseX, mouseY;

void Start()
{
    Cursor.visible = false;
    Cursor.lockState = CursorLockMode.Locked;
}
void LateUpdate()
{
    MoveCamera();
}

public void MoveCamera()
{
    mouseX += Input.GetAxis("Mouse X") * RotationSpeed;
    mouseY -= Input.GetAxis("Mouse Y") * RotationSpeed;
    mouseY = Mathf.Clamp(mouseY, -35, 60);

    transform.LookAt(Target);

    if (Input.GetKey(KeyCode.LeftShift))
    {
        Target.rotation = Quaternion.Euler(mouseY, mouseX, 0);
    }
    else
    {
        Target.rotation = Quaternion.Euler(mouseY, mouseX, 0);
        Player.rotation = Quaternion.Euler(0, mouseX, 0);
    }
}
```

```
        }
```

Task #3. *Create a temporary map*

Description of the implementation (3-5 sentences). *Firstly I downloaded some models from itch.io and started placing them. Secondly, I downloaded 3D modeling app called Blender and started creating my own models. Lastly, I placed some light effects and sound effects.*
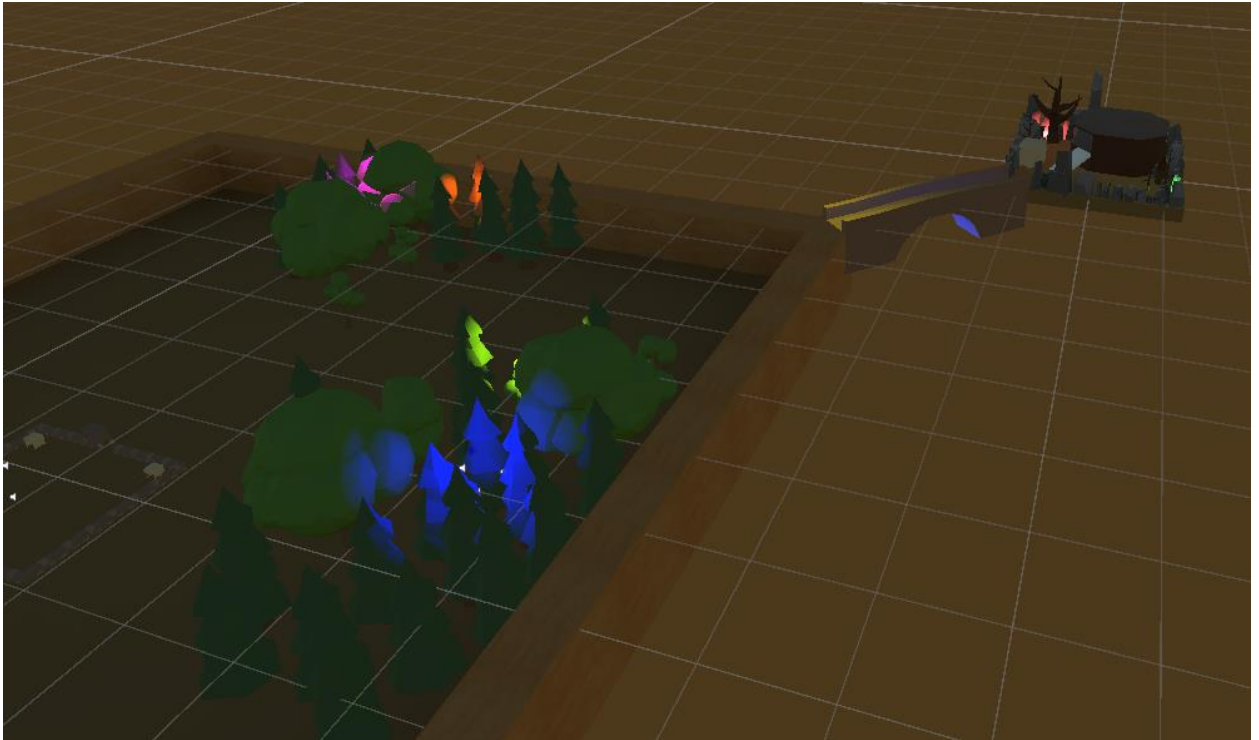

Figure 3. Screenshot #3

Laboratory work #2

List of tasks (main functionality of your project)

1. Title of Task #1
2. Title of Task #2
3. Title of Task #3
4. …

Solution

Task #1. *Title of Task*

Description of implementation (3-5 sentences). *Vestibulum hendrerit felis at turpis ultrices imperdiet. Nulla facilisi curabitur vitae semper nulla. Etiam rhoncus orci dolor, ac dictum erat iaculis sed. Aliquam pulvinar viverra consequat. Nam eu mi in mauris semper pellentesque eget ut erat.*
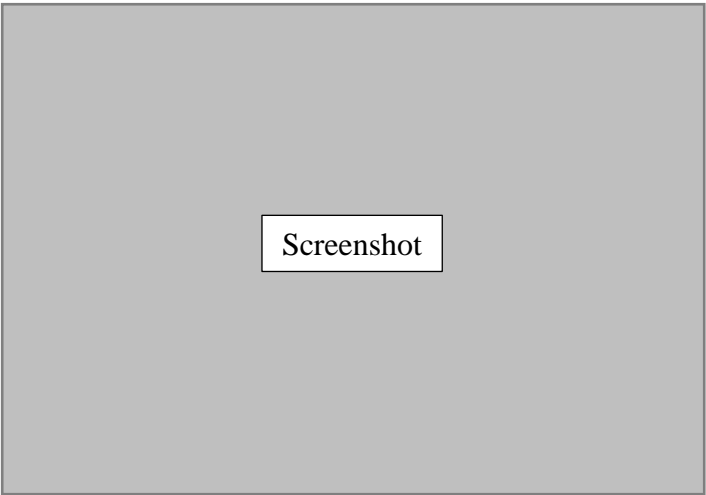
Figure 4. Screenshot #1

In the case of using functions, the description of each main function should be completed with the source code FRAGMENTS (the functions should be indexed in a separate table of contents);

Table 3. Title of fragment #1

Task #2. *Title of Task*

Description of implementation (3-5 sentences). *Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed ultricies nunc sit amet sem blandit, at ultricies nibh commodo. Duis ut mollis risus. Proin hendrerit libero eu felis dapibus imperdiet. Fusce posuere felis ornare luctus molestie. Duis ut odio pretium, bibendum elit et, molestie quam.*
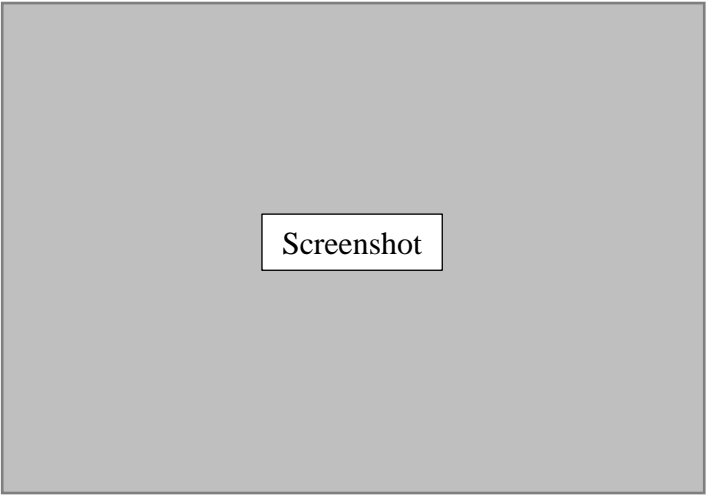

Figure 5. Screenshot #2

In the case of using functions, the description of each main function should be completed with the source code FRAGMENTS (the functions should be indexed in a separate table of contents);

Fragment of Source Code

Table 4. Title of fragment #2

Task #3. *Title of Task*

Description of implementation (3-5 sentences). *Vestibulum hendrerit felis at turpis ultrices imperdiet. Nulla facilisi curabitur vitae semper nulla. Etiam rhoncus orci dolor, ac dictum erat iaculis sed. Aliquam pulvinar viverra consequat. Nam eu mi in mauris semper pellentesque eget ut erat.*

Figure 6. Screenshot #3

In the case of using functions, the description of each main function should be completed with the source code FRAGMENTS (the functions should be indexed in a separate table of contents);

Fragment of Source Code

Table 5. Title of fragment #3

Laboratory work #3

List of tasks (main functionality of your project)

1. Title of Task #1
2. Title of Task #2
3. Title of Task #3
4. …

Solution

Task #1. *Title of Task*

Description of implementation (3-5 sentences). *Vestibulum hendrerit felis at turpis ultrices imperdiet. Nulla facilisi curabitur vitae semper nulla. Etiam rhoncus orci dolor, ac dictum erat iaculis sed. Aliquam pulvinar viverra consequat. Nam eu mi in mauris semper pellentesque eget ut erat.*
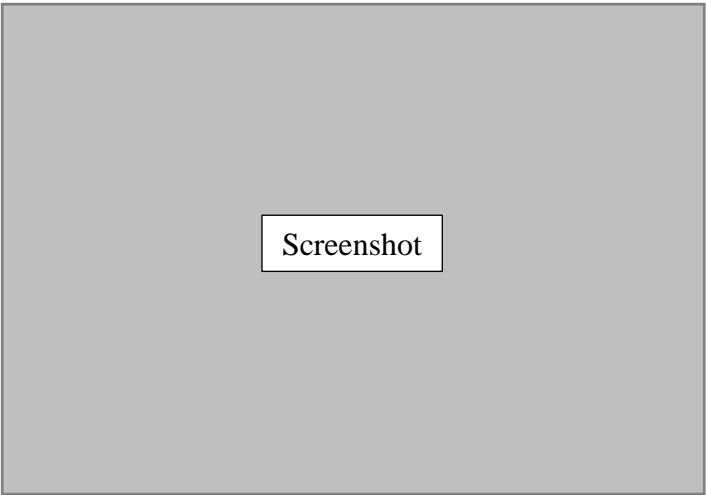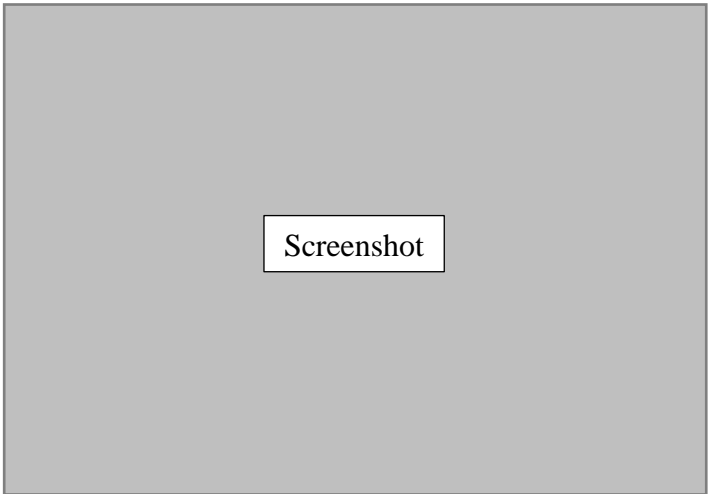
Screenshot

Figure 7. Screenshot #1

In the case of using functions, the description of each main function should be completed with the source code FRAGMENTS (the functions should be indexed in a separate table of contents);

Fragment of Source Code

Table 6. Title of fragment #1

Task #2. *Title of Task*

Description of implementation (3-5 sentences). *Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed ultricies nunc sit amet sem blandit, at ultricies nibh commodo. Duis ut mollis risus. Proin hendrerit libero eu felis dapibus imperdiet. Fusce posuere felis ornare luctus molestie. Duis ut odio pretium, bibendum elit et, molestie quam.*
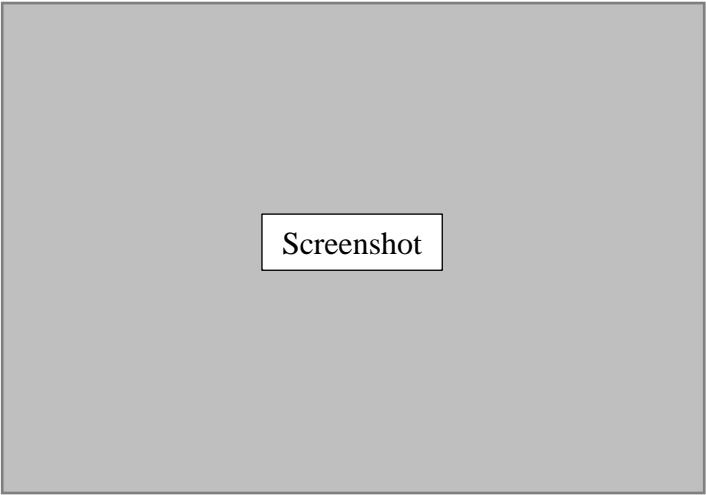
Figure 8. Screenshot #2

In the case of using functions, the description of each main function should be completed with the source code FRAGMENTS (the functions should be indexed in a separate table of contents);

Table 7. Title of fragment #2

Task #3. *Title of Task*

Description of implementation (3-5 sentences). *Vestibulum hendrerit felis at turpis ultrices imperdiet. Nulla facilisi curabitur vitae semper nulla. Etiam rhoncus orci dolor, ac dictum erat iaculis sed. Aliquam pulvinar viverra consequat. Nam eu mi in mauris semper pellentesque eget ut erat.*
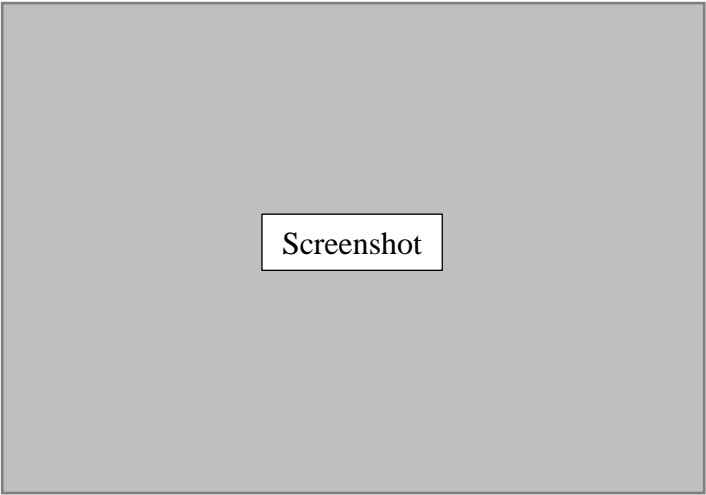

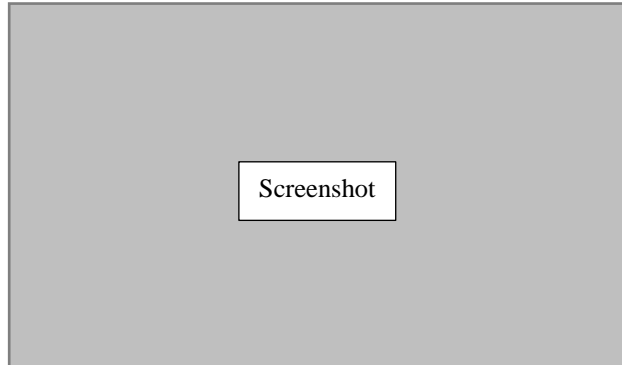
Figure 9. Screenshot #3

In the case of using functions, the description of each main function should be completed with the source code FRAGMENTS (the functions should be indexed in a separate table of contents);

Fragment of Source Code

Table 8. Title of fragment #3

User's manual (for the Individual work defence)

How to play? *Aenean eu quam gravida, laoreet nisl eu, sagittis quam. Donec sit amet nunc nisi. Sed vel ipsum metus. Nullam accumsan vestibulum ex. Aenean eu quam gravida, laoreet nisl eu, sagittis quam. Donec sit amet nunc nisi. Sed vel ipsum metus. Nullam accumsan vestibulum ex.*

Figure 10. Screenshot #5

Nunc vel enim vel magna interdum dapibus id nec nisl. Suspendisse elit augue, accumsan tempor erat sed, gravida suscipit urna. Duis blandit lacus et finibus finibus. Mauris pretium pharetra orci dictum luctus. Nullam commodo magna a tincidunt malesuada.
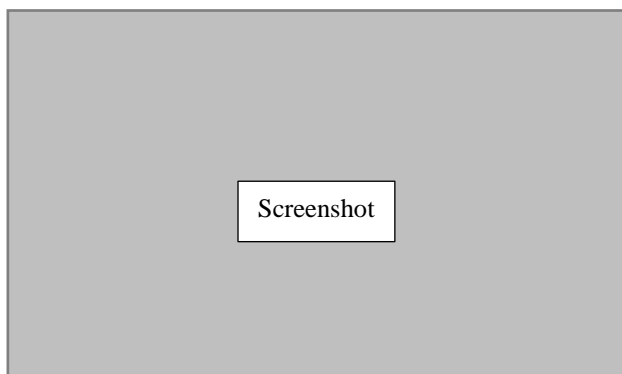
Figure 11. Screenshot #5

Sed sollicitudin justo erat, viverra luctus mi consequat non. Sed ut condimentum libero. Duis rutrum lacus ante, vitae feugiat ex faucibus at. Maecenas pulvinar et augue sed commodo.

Descriptions of the rules of the game. Nunc quis condimentum lacus. Quisque felis neque, ullamcorper vel posuere eget, blandit non neque. Nam in varius erat. Duis molestie sit amet eros vel rhoncus. Nunc quis condimentum lacus. Quisque felis neque, ullamcorper vel posuere eget, blandit non neque. Nam in varius erat. Duis molestie sit amet eros vel rhoncus.

Descriptions of the controls / keys. Donec et lorem vitae ligula bibendum faucibus. Suspendisse interdum quis augue sed luctus. Curabitur ac diam augue. In hac habitasse platea dictumst. Curabitur maximus maximus tortor. Nunc quis condimentum lacus. Quisque felis neque, ullamcorper vel posuere eget, blandit non neque. Nam in varius erat. Duis molestie sit amet eros vel rhoncus.

Literature list

1. Source #1. *https://freesound.org/*
2. Source #2. https://www.youtube.com/watch?v=7nxpDwnU0uU&list=PLrbglElPxW9KvfHztR-8UcZUX0h4jwdgE&index=2&t=0s
3. Source #3 https://www.youtube.com/watch?v=JYj6e-72RDs&index=2&list=PLrbglElPxW9KvfHztR-8UcZUX0h4jwdgE
4. Source #4 https://www.youtube.com/watch?v=Wql4nZkqG0g&list=PLrbglElPxW9KvfHztR-8UcZUX0h4jwdgE&index=3
5. Source #5 https://www.youtube.com/watch?v=69AlA-i1hIU&index=4&list=PLrbglElPxW9KvfHztR-8UcZUX0h4jwdgE

ANNEX

All source code is contained in this part.

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

public class ThirdPersonCamera : MonoBehaviour
{
    public float RotationSpeed = 1f;
    public Transform Target;
    public Transform Player;
    float mouseX, mouseY;

    void Start()
    {
        Cursor.visible = false;
        Cursor.lockState = CursorLockMode.Locked;
    }
    void LateUpdate()
    {
        MoveCamera();
    }

    public void MoveCamera()
    {
        mouseX += Input.GetAxis("Mouse X") * RotationSpeed;
        mouseY -= Input.GetAxis("Mouse Y") * RotationSpeed;
        mouseY = Mathf.Clamp(mouseY, -35, 60);

        transform.LookAt(Target);

        if (Input.GetKey(KeyCode.LeftShift))
        {
            Target.rotation = Quaternion.Euler(mouseY, mouseX, 0);
        }
        else
        {
            Target.rotation = Quaternion.Euler(mouseY, mouseX, 0);
            Player.rotation = Quaternion.Euler(0, mouseX, 0);
        }
    }
}

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

public class PlayerControler : MonoBehaviour
{

    Camera cam;
    public float Speed = 10f;
    public float Gravity = -5f;

    // Start is called before the first frame update
    void Start()
    {
        cam = Camera.main;
    }

    // Update is called once per frame
    void Update()
    {
```

```csharp
        MovePlayer();
        //transform.Translate(Input.GetAxis("Horizontal")*Time.deltaTime*speed, 0f,
Input.GetAxis("Vertical") * Time.deltaTime * speed);
        //Rigidbody rigidBody = GetComponent<Rigidbody>();

        //if (Input.GetKeyDown(KeyCode.A))
        //{

        //}

    }

    void MovePlayer()
    {
        float horizontal = Input.GetAxis("Horizontal");
        float vertical = Input.GetAxis("Vertical");
        Vector3 playerMovement = new Vector3(horizontal, 0f, vertical) * Time.deltaTime *
Speed;
        transform.Translate(playerMovement, Space.Self);
    }
}
```