Computer Graphics and Game Programming – 508837
5th Assignment
Advisor: Assoc. Dr. Murat KURT

5th Assignment Report

06.02.2023

By:
Mehmet ISIKHAN, 91220001906

Table of Contents

Introduction

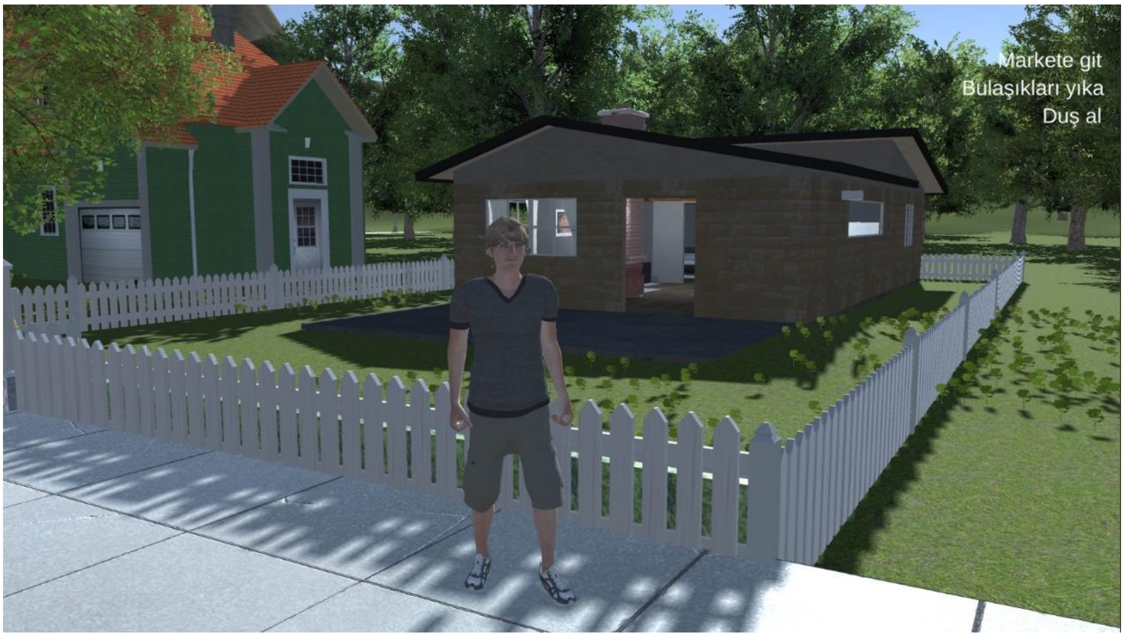
This report includes developing processes, of a house game. This game contains the house’s architecture, house design, furniture inside of the house, and simple house missions of the player inside the game. The player starts on the bedroom, and it should go to fridge for getting the missions because the player’s mom is attached some home mission on the fridge. The house consists of one American kitchen, living room, hall, bedroom, bathroom, and garden. Also, the player can wander his neighborhood and he can buy some things from the neighborhood shop. He should spend his money carefully because his mom is gave a market list to the player as the house mission. The game players can play this game both desktop and mobile platforms (only android devices). The game can be further developed with the new house missions, some jobs, and new money spending areas.

House Game

1. Buildings

a. House

I found the house from the Unity Asset Store [1]. The "asset" includes the ready house and furniture packages, but I redesigned the house according to my pleasure and I added some advanced and normal lights to the house. The house and furniture materials are already included the package, but I changed some textures of the furniture and I added new textures for creating new materials. I designed the garden with the fences, and I determined the house borders with these. I modified the house’s door script for the mobile functuanlity and I provided close and open with a button for the mobile players. Lastly, I attached a note on the fridge for starting process to missions and I designed the paper texture with using the Adobe Photoshop for writing the mission to paper.



Picture.1: The House

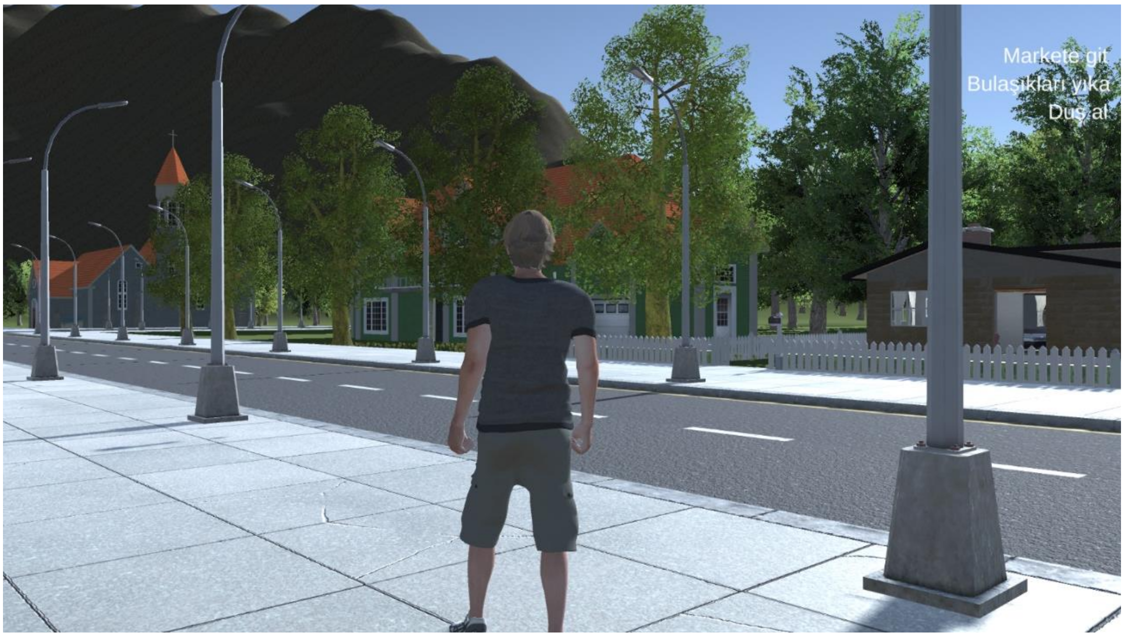
```
using UnityEngine;
using System.Collections;
using UnityEngine.InputSystem;
using UnityEngine.UI;

public class MoveObjectController : MonoBehaviour {
    .
    .
    .
    private Button mobileButton;
    private bool isMobileButtonPressed = false;



    void Start()
    {

        player = GameObject.FindGameObjectWithTag("Player");
        mobileButton =
        GameObject.FindWithTag("MobileButton")?.GetComponent<B
        utton>();
        if(mobileButton != null)
        mobileButton.onClick.AddListener(HandleButtonPress);
        .
        .
        .
        if (Input.GetKeyUp(KeyCode.E)||isMobileButtonPressed){
            anim.enabled = true;
            anim.SetBool(animBoolNameNum,!isOpen);
            msg = getGuiMsg(!isOpen);
            if(mobileButton != null){
            isMobileButtonPressed=!isMobileButtonPressed;
            }
        }
        .
        .
        .
        private void HandleButtonPress(){
            isMobileButtonPressed = true;
        }
        .
        .
        .
```

b. Neighborhood

Firstly, I used terrain asset [2] for the creating ground of the world and I add grass texture for the base texture. Then I made some mountains with using the terrains brushes to around the house and I painted the mountains to the rock texture with again terrains brushes.

Secondly, I added the modular low poly street asset [3] for the making roads. I designed the roads according to the house and I break some spaces for the neighborhood house and its gardens. Then, I added two house assets for the creating neighborhood houses, one of them is low poly buildings lite asset [4] and the other one is house pack asset [5]. The house pack asset [5] contains the same-colored buildings so I changed the houses' material's colors for the variation of the house. Later, I resized, rotated, and positioned the houses around the roads. Moreover, I allocated some space for the neighborhood shop.

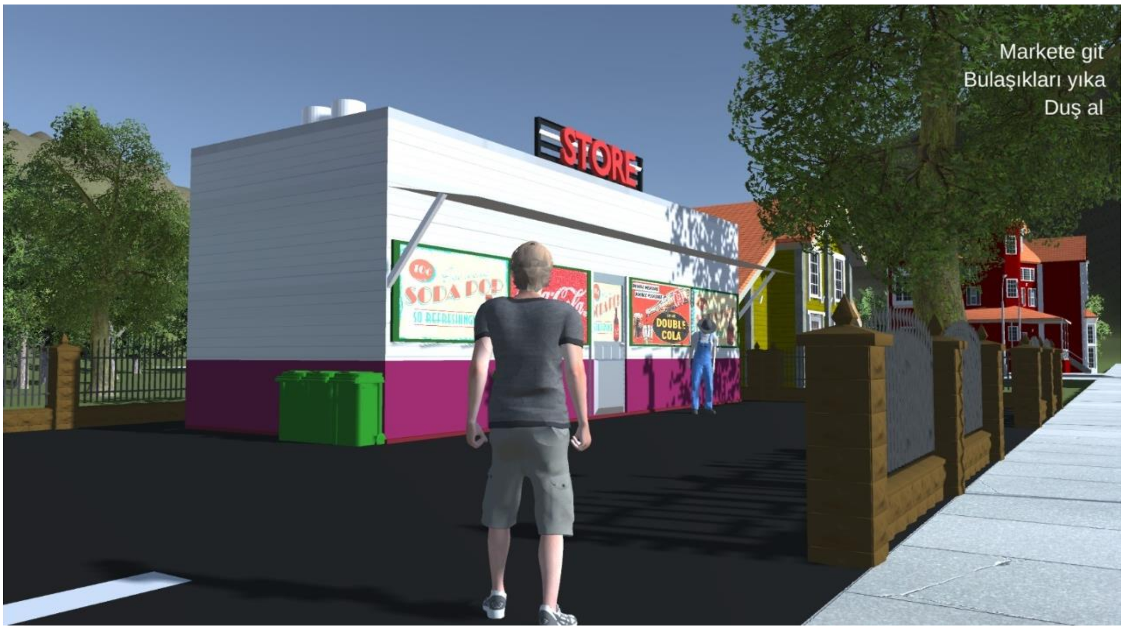
Thirdly, I added different type of trees with using terrain brushes on the plane. In the neighborhood I designed the trees according to buildings and I set new settings for the trees quality and range features to obstruct the high graphical rendering processes.



Picture.2: The Neighborhood

c. Shop

I found the three-dimension shop model on the “cgtrader” web page [6] and I found some advertisement textures for the shop’s windows overlay also I found a wall textures for the shop’s walls from the google images. I create some materials from these textures, and I applied the materials to the shop elements.



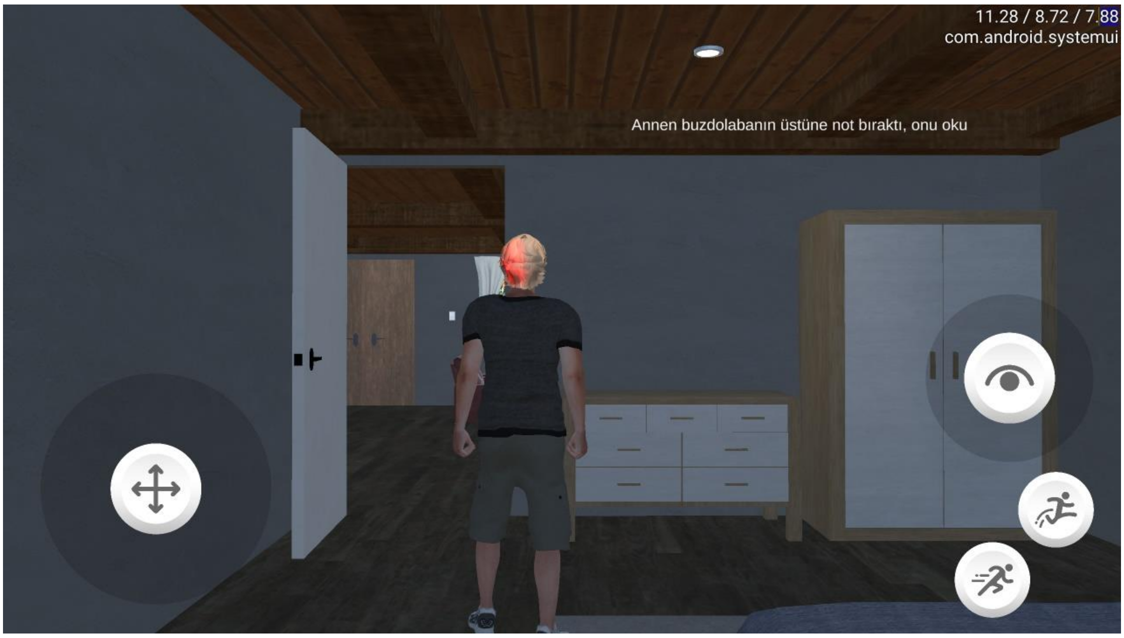
Picture.5: The Shop

2. Characters

a. The Player

Firstly, for the player’s three-dimensional model, I searched on the “cgtrader” web page [6] and I found the model which have the enough body components. I tried to find specially this type of model because the body components are necessary. The player model should be moveable and when the player moving the three-dimensional model components should show these motions, so the body components are important for the creating the "avatar" object in the Unity game engine.

Secondly, for the character controller, I utilized the third-person sample asset provided by Unity [7]. This asset consists of character motions, controllers, and animations. It also includes a sample three-dimensional robot character model, which I replaced with my own three-dimensional model. Additionally, I replaced the "avatar" objects with the new model's motion elements, instead of using the sample robot model [8]. The "avatar" element contains the articulation points of the model for character motions. While the articulation points are generally similar for humanoid models, each model may have some differences in terms of these points.



11.28 / 8.72 / 7.88
com.android.systemui

Annen buzdolabanın üstüne not bıraktı, onu oku

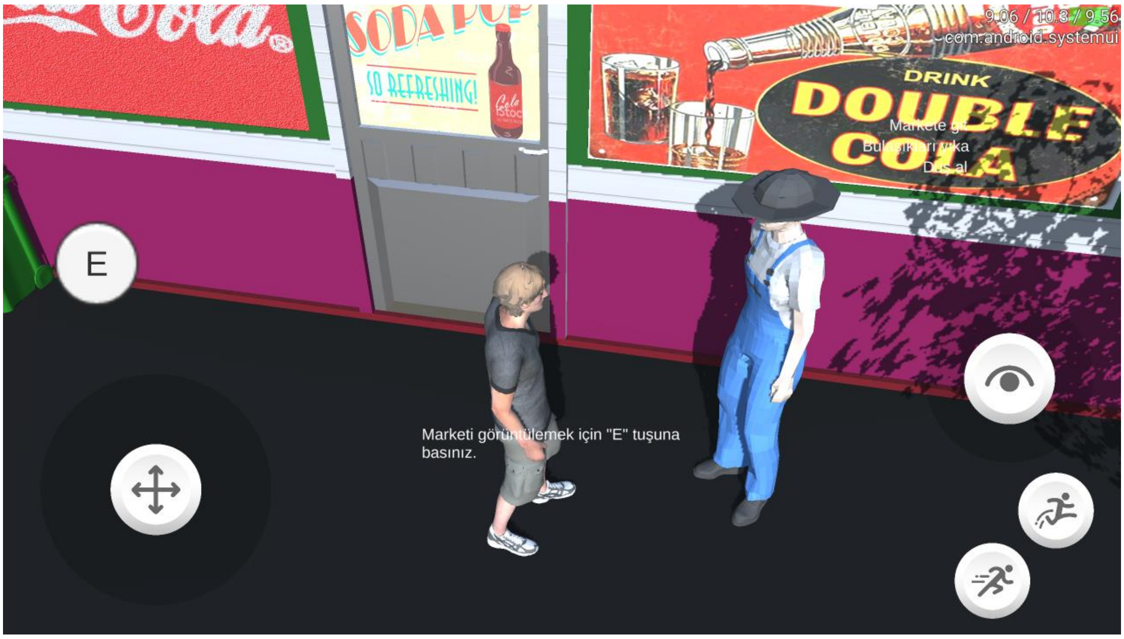
Picture.6: Android Device Character Controller (Without “E” Button)

Lastly, I wrote a script for the activation of the touching player controllers. This script to active the touching player controllers if the device touchable. Else, the system deactivates the touching player controllers, and the players can controller the “W”, “A”, “S”, “D” keys and the mouse. Also, players can control the doors and missions with the “E” key on the desktop version. For the mobile version, there is a “E” button on the screen.

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

public class JoystickController : MonoBehaviour
{

    [SerializeField] private GameObject joystickCanvas;


    void Start()
    {
        if(SystemInfo.deviceType == DeviceType.Handheld){
            joystickCanvas.SetActive(true);
        }else{
            joystickCanvas.SetActive(false);
        }
    }
}
```

b. Seller

Again, I found the seller three-dimensional model for the neighborhood shop from the “cgtrader” web page *[6]*. The t-shirt, overalls, hat and boots materials are the came with the “.fbx” model, but I added new skin material for the seller’s skin. The seller three-dimensional model did not contain the articulation points, so it did not suitable to character animation. But I added a static character animation for the realistic standing in front of the shop. I added this animation from the “mixamo” web page *[9]*. In this web page developed by the Adobe, many character animations are included in it. Also, I added the necessary articulation points thanks to this web page. In this way, the model has had an animation while the standing.



9.06 / 10.8 / 9.56
com.android.systemui

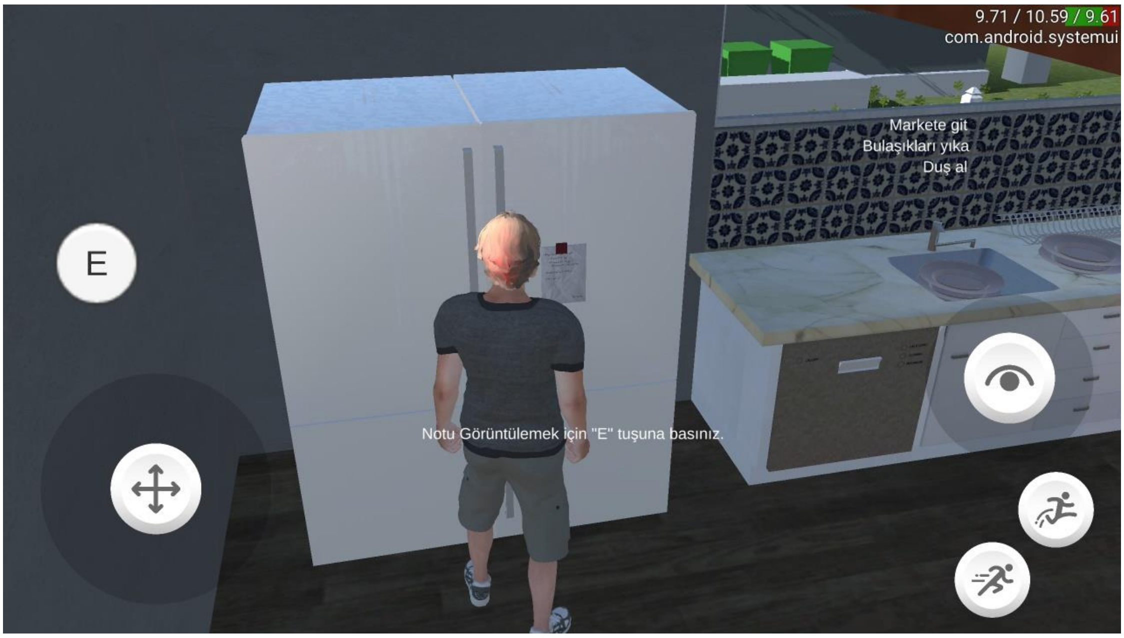
Markete git
Bulaşıkları yıka
Duş al

Marketi görüntülemek için "E" tuşuna basınız.

Picture.7: The Seller Man

3. Missions

When the game starts, players will see a message left top on the screen. Actually, this message about the first mission, and the mission to direct the player to fridge because the character’s mom has attached a note on the fridge. The character's mom has left a note on the fridge, and the objective is for the character to go to the kitchen and read the message.



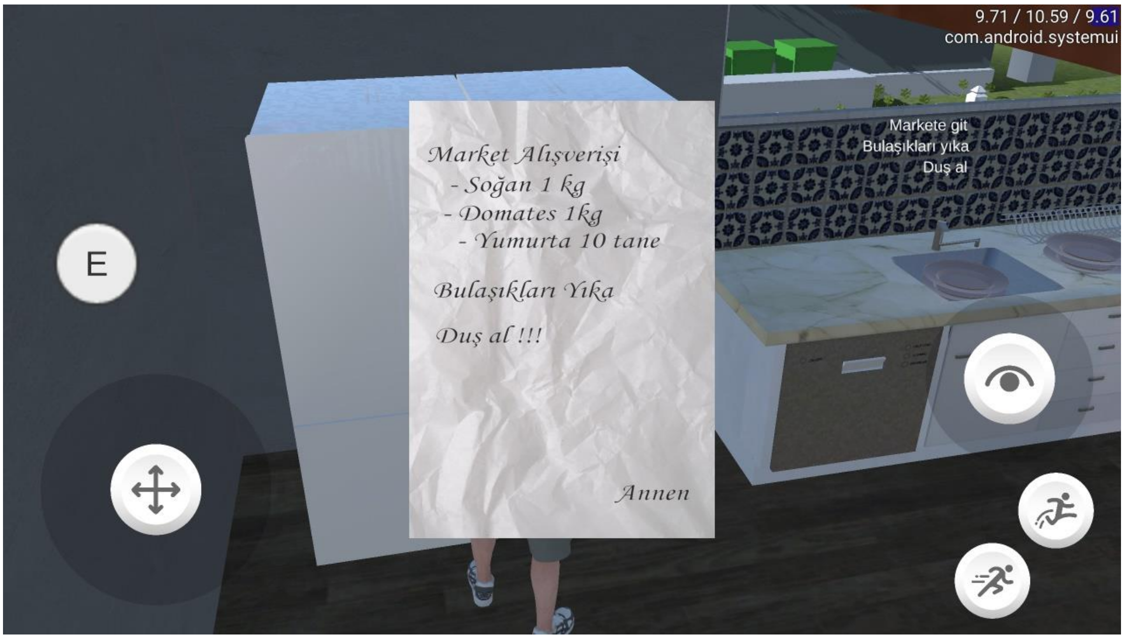
9.71 / 10.59 / 9.61
com.android.systemui

Markete git
Bulaşıkları yıka
Duş al

Notu Görüntülemek için "E" tuşuna basınız.

Picture.7: The First Mission

I have added specific areas on the map for missions. The first area is called the “PaperArea”. When the player enters this area, a script displays an information message instructing them to press the “E” key or button to view a message from the character's mom. The “PaperZone” script is triggered when the character is inside this area, setting the boolean value “paperZone” to true.



Market Alışverişi
- Soğan 1 kg
- Domates 1kg
- Yumurta 10 tane

Bulaşıkları Yıka

Duş al !!!

Annen

Markete git
Bulaşıkları yıka
Duş al

Picture.8: Mission Paper

In the “Update” function, the script checks for a press of the “E” key/button. If the player triggers this event, the “paper” game object is activated and the information message for the paper area is deactivated. Additionally, the static boolean variable “isMomMissionsActive” is set to true when the player displays the mom's message paper. Subsequently, other missions become activated.

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

public class PaperZone : MonoBehaviour
{

    [SerializeField] private UnityEvent zoneEnter;
    [SerializeField] private UnityEvent zoneExit;

    [SerializeField] private GameObject starterInfo;
```

```
[SerializeField] private GameObject paperInfo;
[SerializeField] private GameObject paper;

[SerializeField] private GameObject shopMission;
[SerializeField] private GameObject dishMission;
[SerializeField] private GameObject showerMission;

private bool paperZone = false;
public static bool isMomMissionsActive = false;


void OnTriggerEnter(Collider other)
{
    paperZone = true;
    zoneEnter.Invoke();
    Debug.Log("paperzone in !!!");
}


void OnTriggerExit(Collider other)
{
    paperZone = false;
    zoneExit.Invoke();
    Debug.Log("paperzone out !!!");
}


public void showInfo(){
    paperInfo.SetActive(true);
}

public void closeInfo(){
    paperInfo.SetActive(false);
}

public void showPaper(){
    paper.SetActive(true);
}
public void closePaper(){
    paper.SetActive(false);
}

private void addMissions(){
    starterInfo.SetActive(false);
    shopMission.SetActive(true);
    dishMission.SetActive(true);
    showerMission.SetActive(true);
}
```

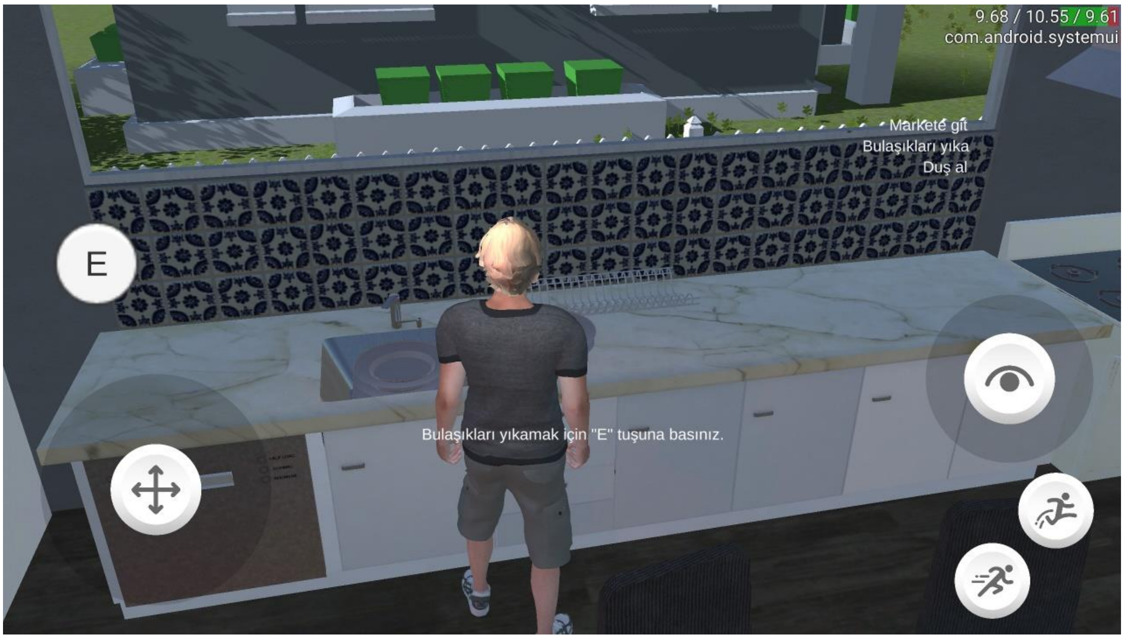
```
    void Update(){
        if (Input.GetKeyUp(KeyCode.E) && paperZone)
        {
            isMomMissionsActive = true;
            addMissions();
            if(paper.active){
                closePaper();
                showInfo();
            }else{
                closeInfo();
                showPaper();
            }

        }
    }

    public void paperButton(){
        if (paperZone)
        {
            isMomMissionsActive = true;
            addMissions();
            if(paper.active){
                closePaper();
                showInfo();
            }else{
                closeInfo();
                showPaper();
            }

        }
    }

}
```

a. Dish Washing Mission

The nearest mission from the fridge. When the player came in front of the sink, it means inside the “DishArea” trigger object, again the mission information shows to the player. When the player presses the “E” key or button, the main character which has controller and free look cam is deactivated and another character object which has dish washing animation I found from the “mixamo” web page activated. Also, new static camera activates for the new character and the players wait 30 second for the dish washing mission. After mission completed, the dirty dishes object set deactivated and clean dishes object activate. End of the mission, the "isDishMissionDone" boolean object set to true for preventing the players to do the same mission again.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Events;
using UnityEngine.SceneManagement;
using static PaperZone;

public class DishZone : MonoBehaviour
{

    [SerializeField] private UnityEvent zoneEnter;
    [SerializeField] private UnityEvent zoneExit;
    [SerializeField] private GameObject dishMission;
    [SerializeField] private GameObject dishMissionInfo;
    [SerializeField] private GameObject dirtyDishes;
    [SerializeField] private GameObject cleanDishes;
    [SerializeField] private GameObject mainPlayer;
    [SerializeField] private GameObject washerPlayer;

    private bool dishZone = false;
    private bool isDishMissionDone = false;
    private bool isDishMissionStart = false;
    private float dishWashingTime = 30;

    void OnTriggerEnter(Collider other)
    {
        zoneEnter.Invoke();
        if(PaperZone.isMomMissionsActive){
            dishZone = true;
            if(!isDishMissionDone)
                showInfo();
            //Debug.Log("dish zone in !!!");
        }

    }

    void OnTriggerExit(Collider other)
    {
        zoneExit.Invoke();
        if(PaperZone.isMomMissionsActive){
             dishZone = false;
             closeInfo();
            //Debug.Log("dish zone out !!!");
        }

    }

    public void showInfo(){
        dishMissionInfo.SetActive(true);
    }
    public void closeInfo(){
        dishMissionInfo.SetActive(false);
    }

    void startDishMission(){
        mainPlayer.SetActive(false);
        washerPlayer.SetActive(true);
    }

    void missinDone(){
        closeInfo();
        mainPlayer.SetActive(true);
        washerPlayer.SetActive(false);
        dirtyDishes.SetActive(false);
        cleanDishes.SetActive(true);
    }

    void Update(){
        if (Input.GetKeyUp(KeyCode.E) && dishZone &&
(!isDishMissionDone) && (!isDishMissionStart))
        {
            isDishMissionStart = true;
            startDishMission();
        }

        if(isDishMissionStart){
            dishWashingTime -= Time.deltaTime;
            Debug.Log(dishWashingTime);
            if(dishWashingTime < 0)
            {
                isDishMissionStart = false;
                isDishMissionDone = true;
                missinDone();
            }
        }
    }

    public void dishButton(){
        if (dishZone && (!isDishMissionDone) &&
(!isDishMissionStart))
        {
            isDishMissionStart = true;
            startDishMission();
        }
    }

}
```

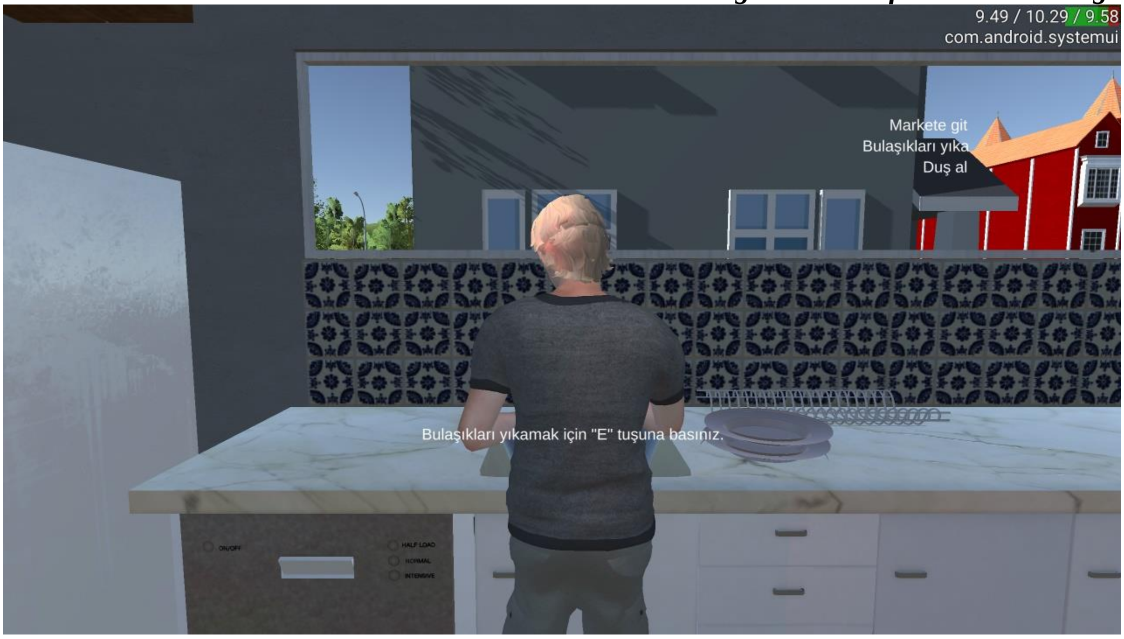
Also, the script checks mission start or not for preventing the pressing “E” button or key while mission doing.



Markete git
Bulaşıkları yıka
Duş al

Bulaşıkları yıkamak için "E" tuşuna basınız.

Picture.9: Dish Washing Mission Information Message



Markete git
Bulaşıkları yıka
Duş al

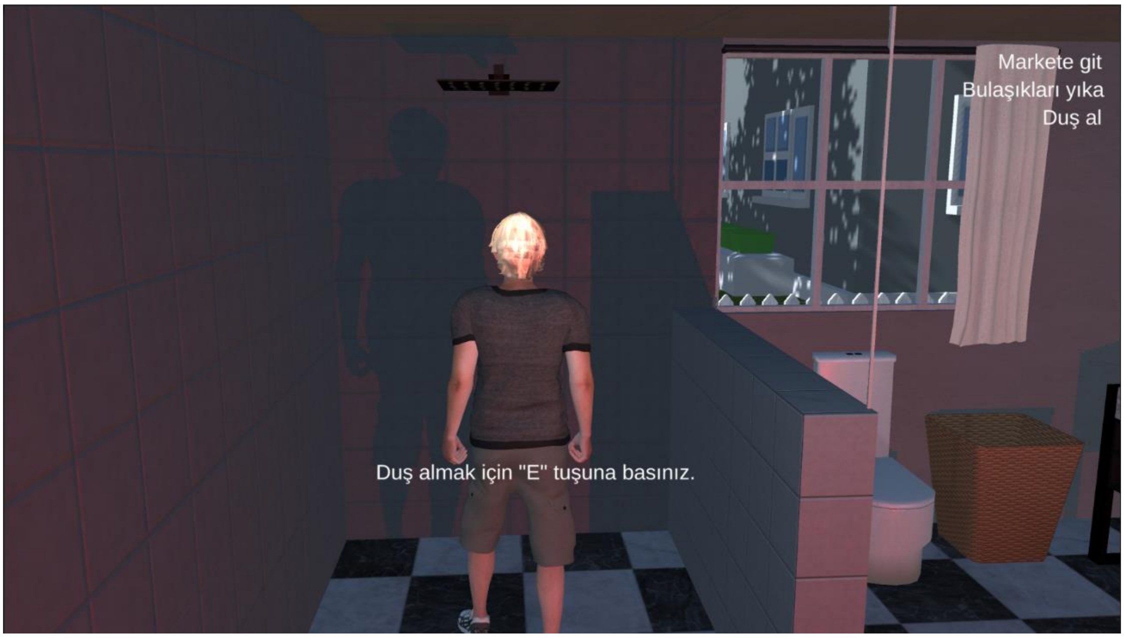
Bulaşıkları yıkamak için "E" tuşuna basınız.

Picture.10: Dish Washing Mission

b. Shower Mission

This mission involves multiple animation combinations. When the player presses the "E" button/key in the shower, the main character, who is currently active, is deactivated, and the shower character becomes active with multiple animations. These animations include the shower character turning back, walking to the bathroom door, and closing it. The camera, which is active with the shower character, follows the character until the bathroom door is closing and the player waiting 45 seconds for the player's shower. After the mission is completed, the shower character is deactivated, and the main character becomes active again.

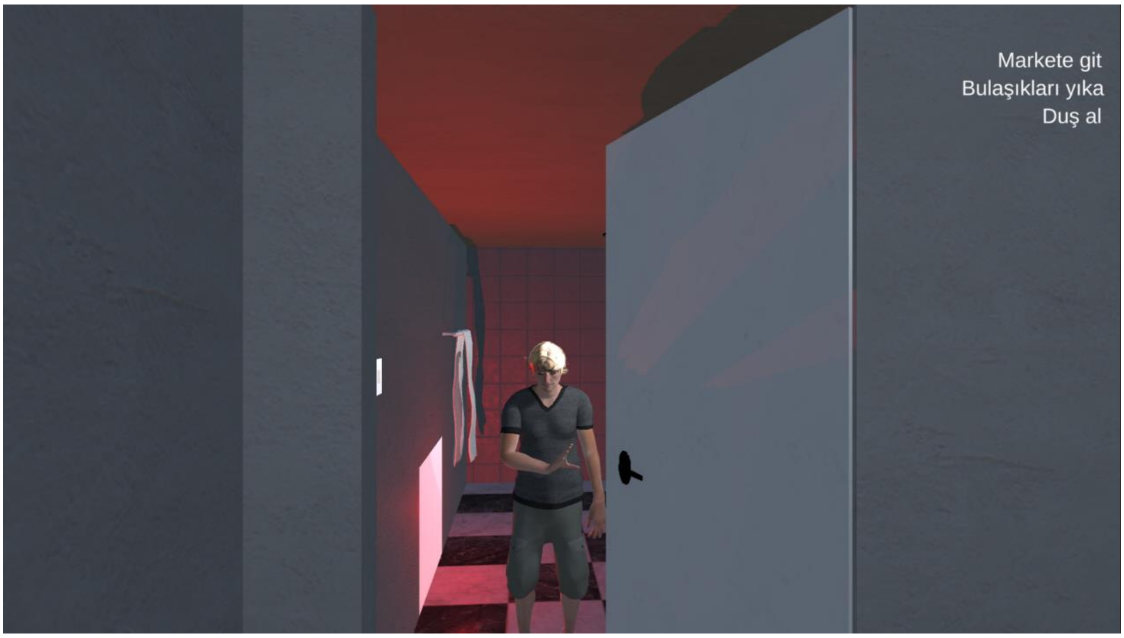
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Events;

public class ShowerZone : MonoBehaviour
{
    [SerializeField] private UnityEvent zoneEnter;
    [SerializeField] private UnityEvent zoneExit;
    [SerializeField] private GameObject showerMissionInfo;
    [SerializeField] private GameObject mainPlayer;
    [SerializeField] private GameObject showerPlayer;
    [SerializeField] private Animator showerAnimator;
    private int cameraAnimationHash;
    [SerializeField] private GameObject showerDoor;
    [SerializeField] private GameObject showerCamera;
    private bool showerZone = false;
    private bool isShowerMissionDone = false;
    private bool isShowerMissionStart = false;
    private float showerTime = 45;

    private void Start()
    {
        cameraAnimationHash = 
Animator.StringToHash("ShowerDoorClose"); // Replace with the name 
of your animation state
    }

    void OnTriggerEnter(Collider other)
    {
        zoneEnter.Invoke();
        if(PaperZone.isMomMissionsActive && 
(!isShowerMissionDone)){
            showerZone = true;
            showInfo();
            //Debug.Log("dish zone in !!!");
        }

    }

    void OnTriggerExit(Collider other)
    {
        zoneExit.Invoke();
        if(PaperZone.isMomMissionsActive){
             showerZone = false;
             closeInfo();
            //Debug.Log("dish zone out !!!");
        }

    }
    public void showInfo(){
        showerMissionInfo.SetActive(true);
    }

    public void closeInfo(){
        showerMissionInfo.SetActive(false);
    }

    void startShowerMission(){
        mainPlayer.SetActive(false);
        showerPlayer.SetActive(true);
        showerDoor.SetActive(true);
        showerCamera.SetActive(true);
    }

    void missionDone(){
        mainPlayer.SetActive(true);
        showerPlayer.SetActive(false);
        showerDoor.SetActive(false);
        showerCamera.SetActive(false);
    }

    void Update(){
        AnimatorStateInfo stateInfo = 
showerAnimator.GetCurrentAnimatorStateInfo(0);
        if (Input.GetKeyUp(KeyCode.E) && showerZone && 
(!isShowerMissionDone) && (!isShowerMissionStart))
        {
            isShowerMissionStart = true;
            closeInfo();
            startShowerMission();
        }

        if(isShowerMissionStart){
            showerTime -= Time.deltaTime;
            Debug.Log(showerTime);
            if(showerTime < 0)
            {
                isShowerMissionStart = false;
                isShowerMissionDone = true;
                missionDone();
            }
        }
    }

    public void showerButton(){
        if (showerZone && (!isShowerMissionDone) && 
(!isShowerMissionStart))
        {
            isShowerMissionStart = true;
            closeInfo();
            startShowerMission();
        }    }   }
```



Markete git
Bulaşıkları yıka
Duş al

Duş almak için "E" tuşuna basınız.

Picture.11: Shower Mission

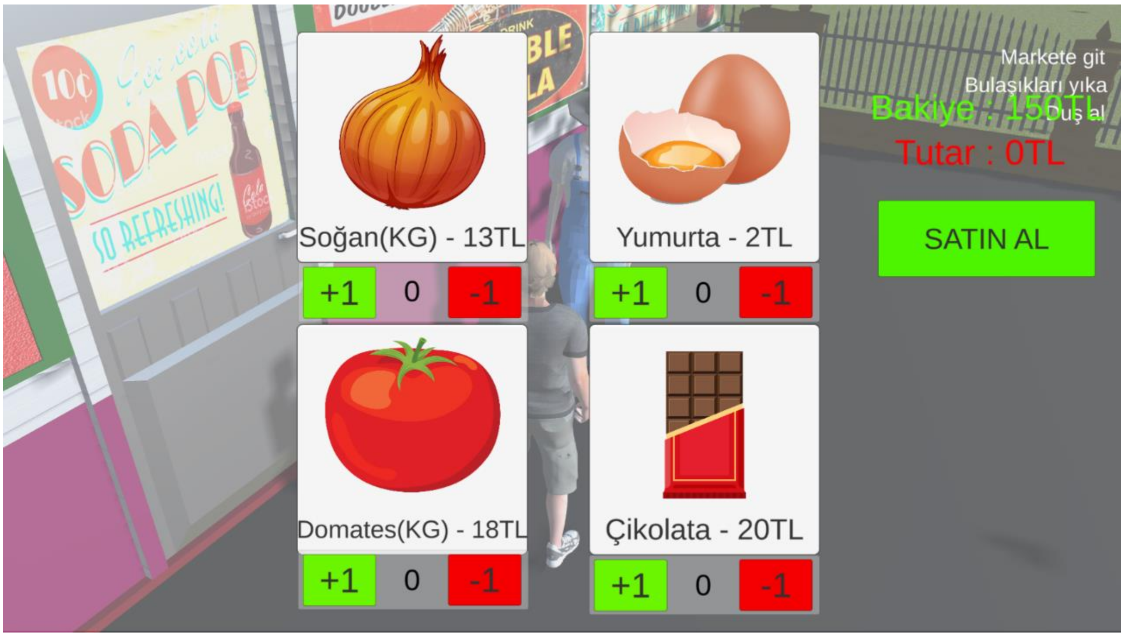


Markete git
Bulaşıkları yıka
Duş al

Picture.12: Shower Mission

c. Shopping Mission

This mission is outside the house, so the player must go to the shop which on the other side of the road. The player has “150TL” beginning of the game and this value is storing to “balance” variable in the “Market” script. This script contains the shopping interface and buying process. All market products have “plus” and “minus” buttons, the player can preference these products with some limitation and money limit. When the player buys a product, the determined product cost is adding to general cost variable and if the player presses the buy button the “cost” variable is deducted from the player’s “balance” value. The player should buy the mission products from the shop.



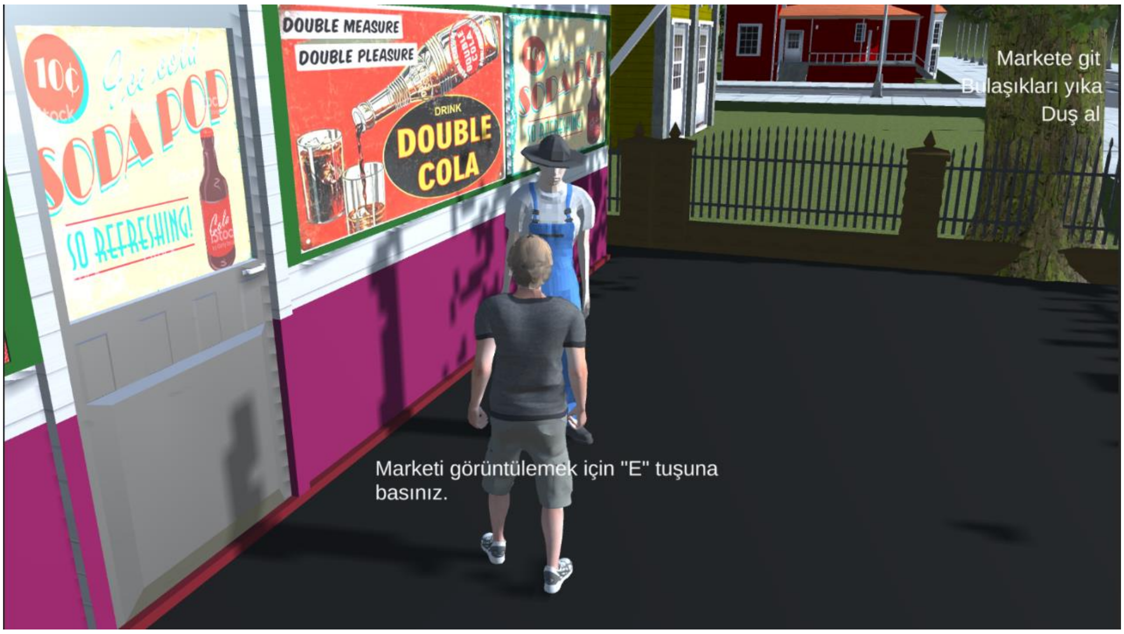
Picture.13: Shop Menu

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

public class Market : MonoBehaviour
{
    [SerializeField] private TextMeshProUGUI onion;
    [SerializeField] private TextMeshProUGUI egg;
    [SerializeField] private TextMeshProUGUI tomato;
    [SerializeField] private TextMeshProUGUI chocolate;
    [SerializeField] private TextMeshProUGUI costTxt;
    [SerializeField] private TextMeshProUGUI balanceTxt;
```

```
private static int balance = 150;
private static int numberOfOnion = 0;
private static int numberOfTomato = 0;
private static int numberOfEgg = 0;
private static int numberOfChocolate = 0;
private static int cost = 0;

public void onionPlus(){
    if(numberOfOnion != 10){
        numberOfOnion++;
        cost+=13;
        onion.text = numberOfOnion.ToString();
        costTxt.text = "Tutar : "+cost.ToString()+"TL";
    }
}

public void onionMinus(){
    if(numberOfOnion != 0){
        numberOfOnion--;
        cost-=13;
        onion.text = numberOfOnion.ToString();
        costTxt.text = "Tutar : "+cost.ToString()+"TL";
    }
}

public void eggPlus(){
    if(numberOfEgg != 10){
        numberOfEgg++;
        cost+=2;
        egg.text = numberOfEgg.ToString();
        costTxt.text = "Tutar : "+cost.ToString()+"TL";
    }
}

public void eggMinus(){
    if(numberOfEgg != 0){
        numberOfEgg--;
        cost-=2;
        egg.text = numberOfEgg.ToString();
        costTxt.text = "Tutar : "+cost.ToString()+"TL";
    }
}

public void tomatoPlus(){
    if(numberOfTomato != 10){
        numberOfTomato++;
        cost+=18;
        tomato.text = numberOfTomato.ToString();
        costTxt.text = "Tutar : "+cost.ToString()+"TL";
    }
}
```

```
    public void tomatoMinus(){
        if(numberOfTomato != 0){
            numberOfTomato--;
            cost-=18;
            tomato.text = numberOfTomato.ToString();
            costTxt.text = "Tutar : "+cost.ToString()+"TL";
        }
    }

    public void chocolatePlus(){
        if(numberOfChocolate != 5){
            numberOfChocolate++;
            cost+=20;
            chocolate.text = numberOfChocolate.ToString();
            costTxt.text = "Tutar : "+cost.ToString()+"TL";
        }
    }

    public void chocolateMinus(){
        if(numberOfChocolate != 0){
            numberOfChocolate--;
            cost-=20;
            chocolate.text = numberOfChocolate.ToString();
            costTxt.text = "Tutar : "+cost.ToString()+"TL";
        }
    }

    public void buy(){
        if(balance > cost){
            balance = balance - cost;
            costTxt.text = "Tutar : 0TL";
            balanceTxt.text = "Bakiye : "+balance.ToString()+"TL";
            chocolate.text = "0";
            tomato.text = "0";
            egg.text = "0";
            onion.text = "0";
        }
    }
}
```

In order to open the market menu, the player needs to be near the seller man who is located in front of the shop. In this case, the “ShopeZone” script acts like the other mission scripts but when the players press the “E” key on the desktop, it means when the shop menu active, the cursor set active for purchasing products. Also, when the player gets away from the seller man the shop menu is closing. The players can close and open the shop menu with the “E” key/button.



Markete git
Bulaşıkları yıka
Duş al

Marketi görüntülemek için "E" tuşuna basınız.

Picture.14: Shopping Mission

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

using static PaperZone;

public class ShopZone : MonoBehaviour
{

    [SerializeField] private UnityEvent zoneEnter;
    [SerializeField] private UnityEvent zoneExit;


    [SerializeField] private GameObject shopMissionInfo;
    [SerializeField] private GameObject shopMenu;


    private bool shopZone = false;
    private bool isMenuOpenned = false;
```

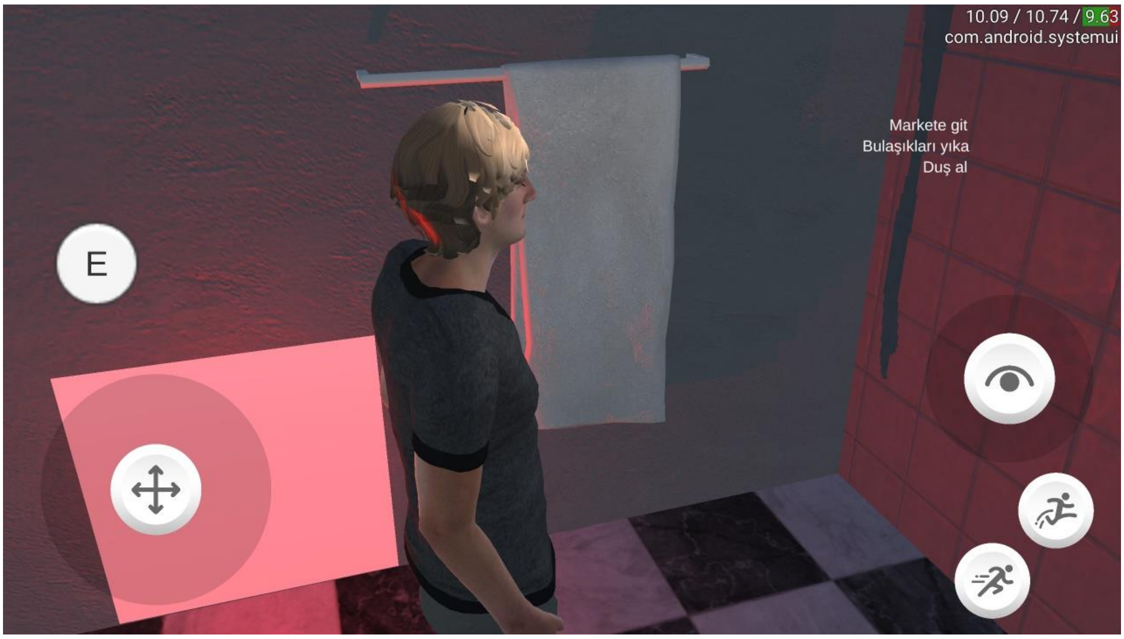
```
void OnTriggerEnter(Collider other)
{
    zoneEnter.Invoke();
    if(PaperZone.isMomMissionsActive && (!isMenuOpenned)){
        shopZone = true;
        showInfo();
    }
}

void OnTriggerExit(Collider other)
{
    zoneExit.Invoke();
    if(PaperZone.isMomMissionsActive){
         shopZone = false;
         isMenuOpenned = false;
         closeInfo();
         closeShopMenu();
    }
}

public void showInfo(){
    shopMissionInfo.SetActive(true);
}

public void closeInfo(){
    shopMissionInfo.SetActive(false);
}

private void openShopMenu(){
    Cursor.visible = true;
    Cursor.lockState = CursorLockMode.None;
    shopMenu.SetActive(true);
}

private void closeShopMenu(){
    Cursor.visible = false;
    Cursor.lockState = CursorLockMode.Locked;
    shopMenu.SetActive(false);
}

public void shopButton(){
    if (shopZone && (!isMenuOpenned))
    {
        isMenuOpenned = true;
        closeInfo();
        openShopMenu();
    }else if(shopZone && (isMenuOpenned)){
        isMenuOpenned = false;
        closeShopMenu();
        showInfo();
    }
}
```

```
       void Update(){
         if (Input.GetKeyUp(KeyCode.E) && shopZone &&
(!isMenuOpenned))
         {
             isMenuOpenned = true;
             closeInfo();
             openShopMenu();
         }else if(Input.GetKeyUp(KeyCode.E) && shopZone &&
(isMenuOpenned)){
             isMenuOpenned = false;
             closeShopMenu();
             showInfo();
         }
     }
}
```

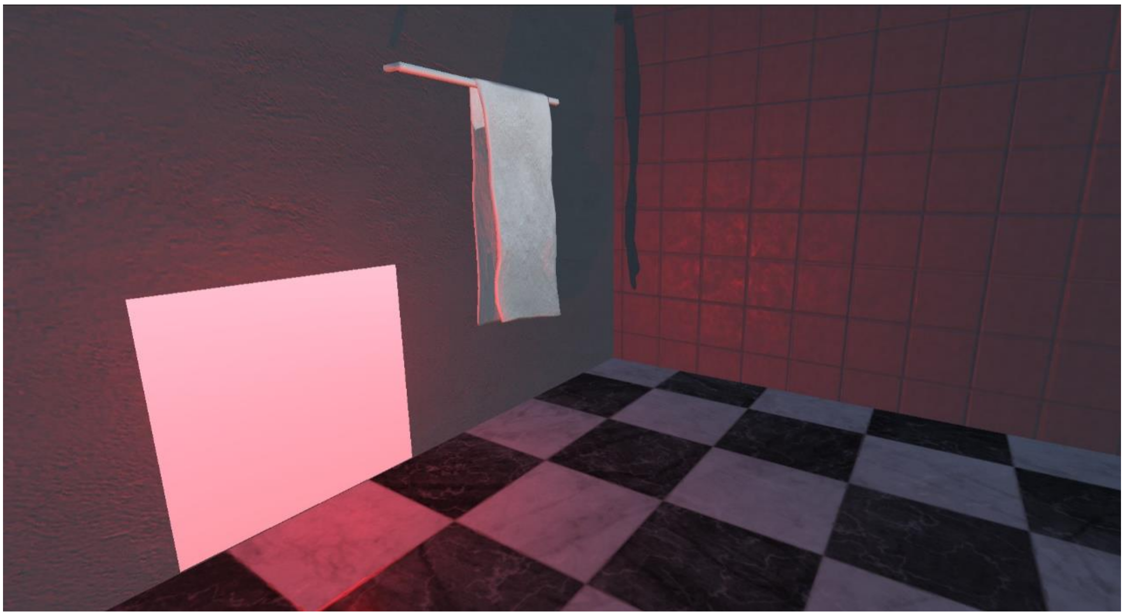
4. Lighting

I added a directional light for the general lighting both house and neighborhood components (trees, houses etc.) and in the house, I have simple spotlights on the ceiling for the house simple lighting. Also, I added advance lighting panel to the bathroom with red colored.

Picture.15: Spotlight

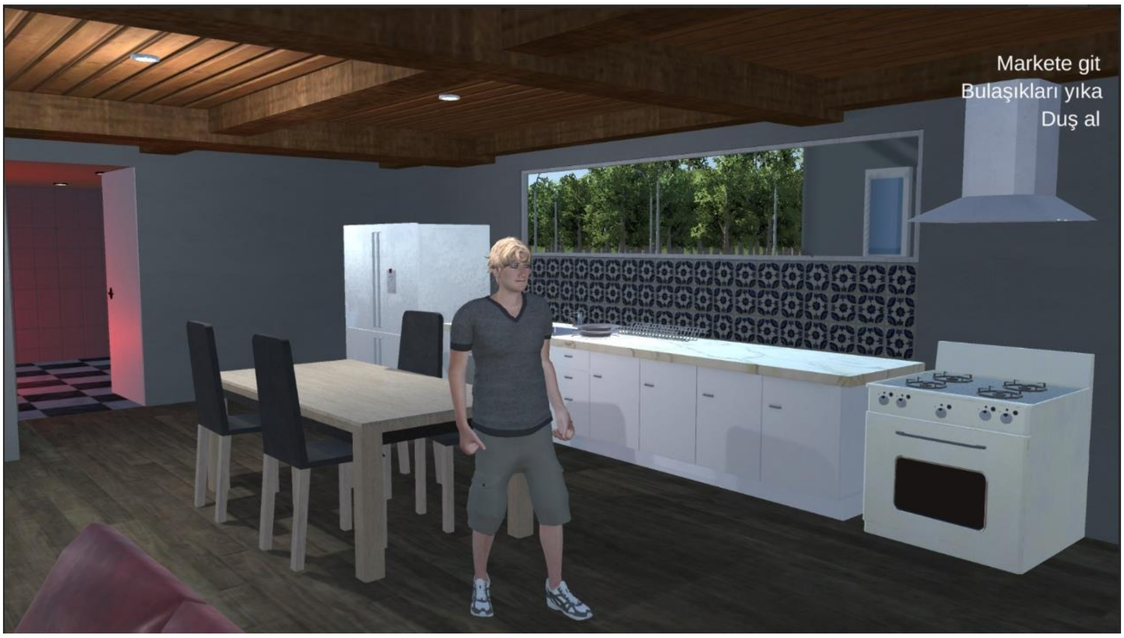


Picture.16: Advance Red Loess Light Panel



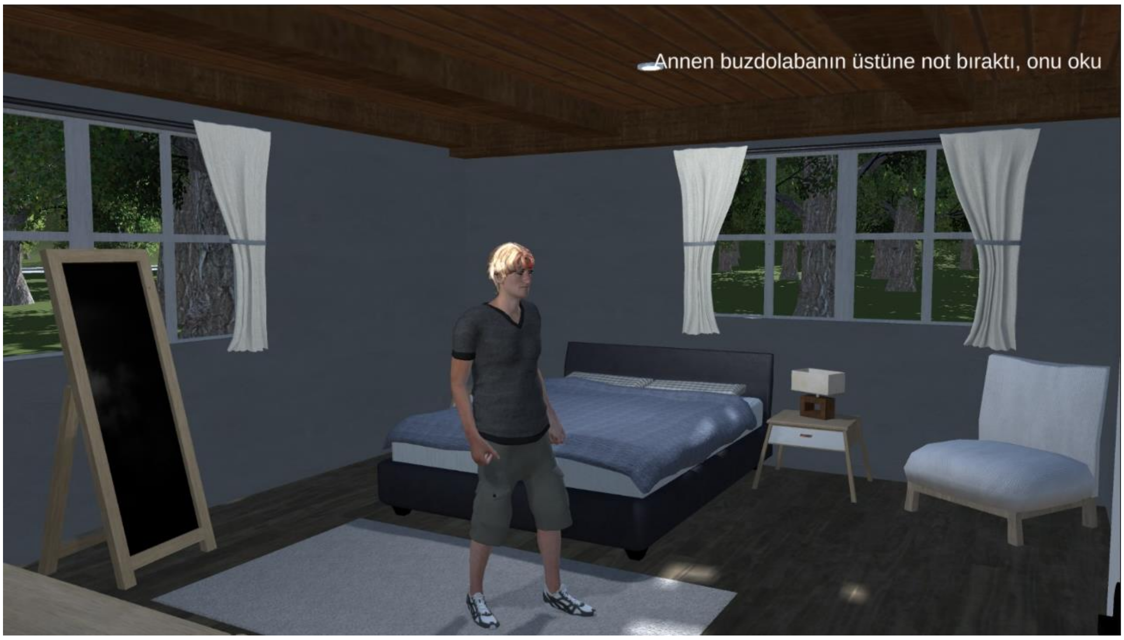
Picture.17: Advance Red Loess Light Panel

Screen Shots



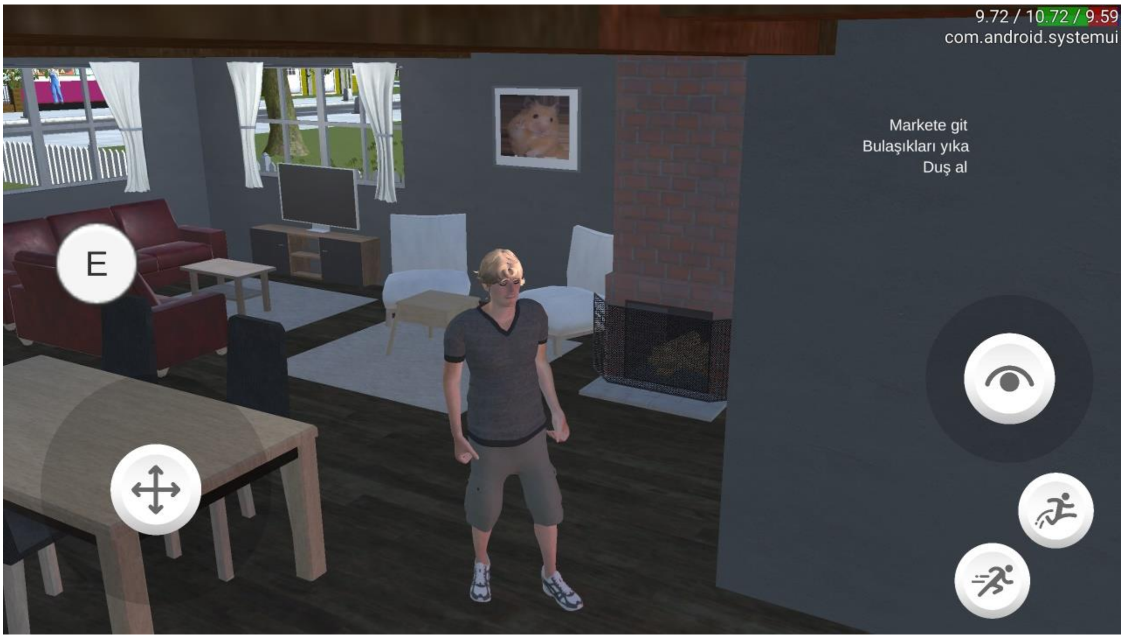
Markete git
Bulaşıkları yıka
Duş al

Picture.18: Kitchen



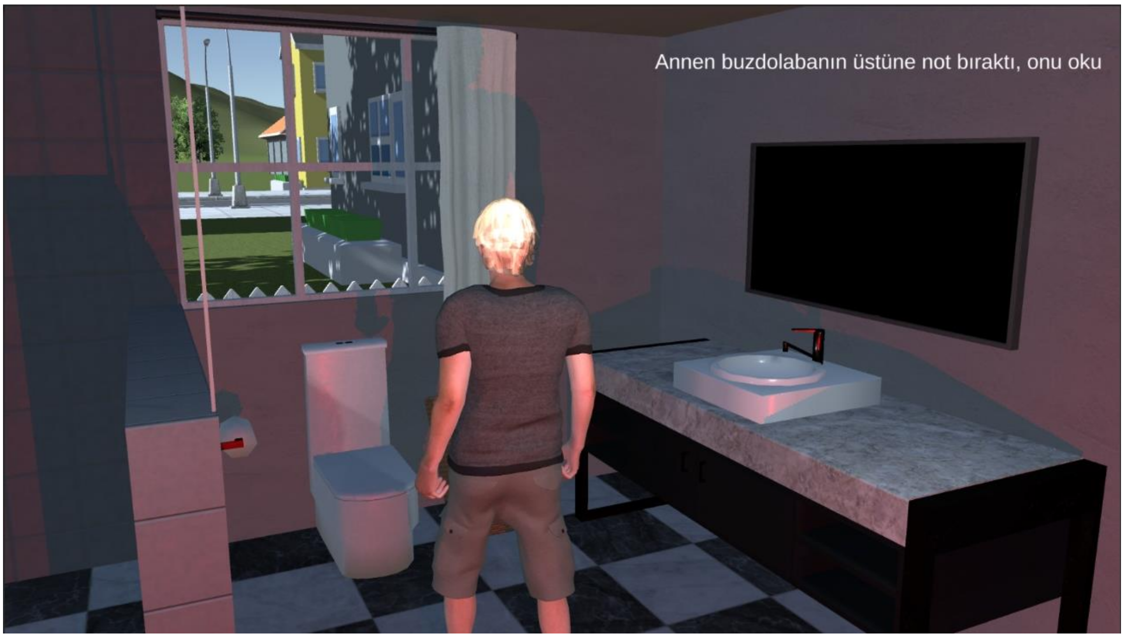
Annen buzdolabanın üstüne not bıraktı, onu oku

Picture.19: Bedroom



Markete git
Bulaşıkları yıka
Duş al

Picture.20: Livingroom



Annen buzdolabanın üstüne not bıraktı, onu oku

Picture.20: Bathroom

References

1. https://assetstore.unity.com
2. https://assetstore.unity.com/packages/3d/environments/landscapes/terrain-sample-asset-pack-145808
3. https://assetstore.unity.com/packages/p/modular-lowpoly-streets-free-192094
4. https://assetstore.unity.com/packages/3d/environments/low-poly-buildings-lite-98836
5. https://assetstore.unity.com/packages/3d/environments/low-poly-buildings-lite-98836
6. https://www.cgtrader.com/3d-models
7. https://assetstore.unity.com/packages/essentials/starter-assets-third-person-character-controller-196526
8. https://medium.com/nerd-for-tech/how-to-replace-unitys-3rd-person-armature-9339dbaabc48#:~:text=With%20the%20Player%20Armature%20object,about%20with%20your%20new%20model!
9. https://www.mixamo.com