



Our City Our Water

Presented by Group 5: Gundo,
Xolani, Thato, Oeratile and
Marco

Our Team

Developer



Thato

Developer



Oreratile

UI designer



Marco

3D Modeler



Xolani

Sound Designer



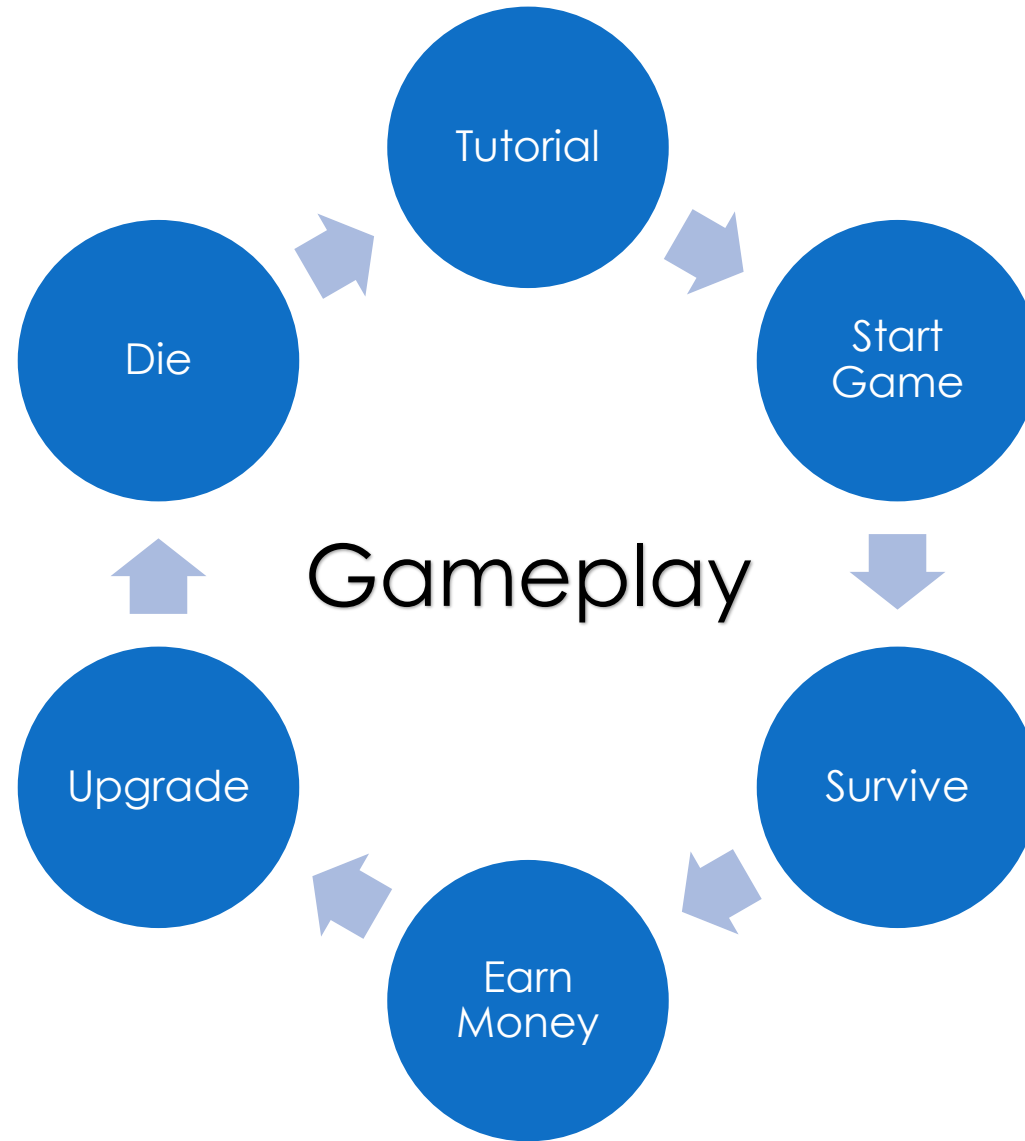
Gundo



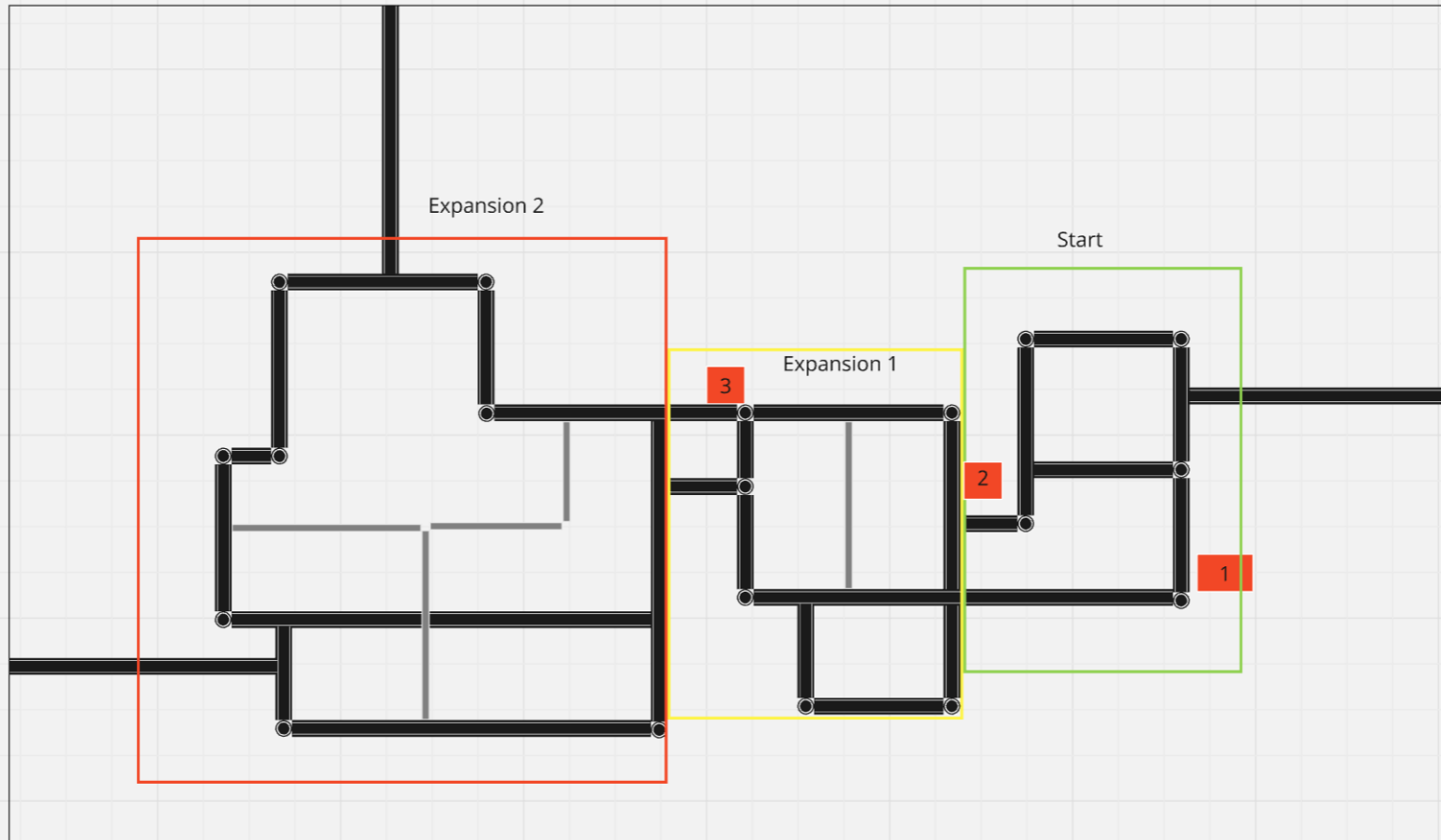
What is Our City Our Water?



Keep the citizens happy
Keep the manholes clean

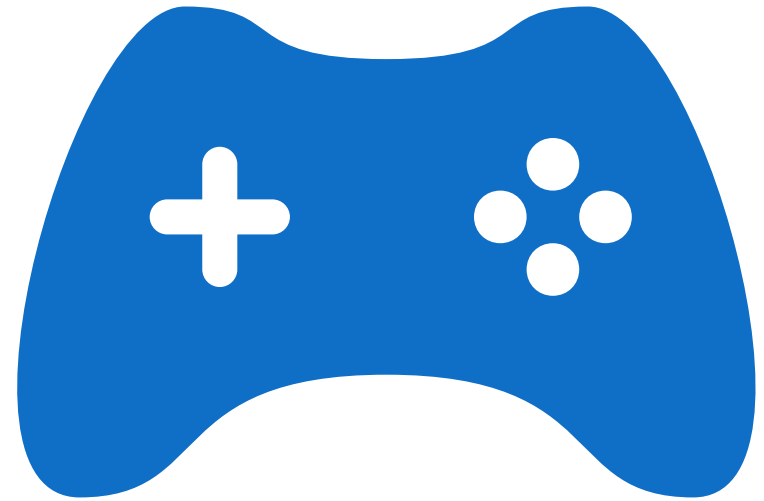


The city layout



Move of the
location from
expansion
addition

Code



```

public class BuyTruck : MonoBehaviour
{
    public TruckImageManager TruckImageManager;
    public TMP_Text ButtonText;
    public int TruckCost = 5;
    //Gundo edit
    private void Start()
    {
        ButtonText.text = "Buy Truck: \n" + TruckCost;
    }
    public void BuyNewTruck()
    {
        SRC.instance.ButtonPress();
        if(TruckImageManager.TrucksAvailable < 5)
        {
            if(GameManager.instance.coins >= TruckCost)
            {
                Spawner.instance.spawnCell(Selecting.Instance.placeholder);
                Station.instance.SendOut(Selecting.Instance.placeholder);
                GameManager.instance.UseCoins(TruckCost);
                GameManager.instance.TruckCount++;
                TruckImageManager.TrucksAvailable++;
                TruckImageManager.TrucksInside++;
                TruckCost += 10;
                ButtonText.text = "Buy Truck: \n" + TruckCost;
                TruckImageManager.ChangeImage();
            }
        }
    }
}

```

```

public int HappinessMeter = 100;
public float changeSpeed; //Gradual Change in Slider
public List<GameObject> ManHolesList = new List<GameObject>();
public static SliderManager instance;
private void Awake()
{
    instance = this;
}

void Update()
{
    HappinessCalculationAndMeterUpdater();
}

public void addManHoleToList(GameObject manHole)
{
    ManHolesList.Add(manHole);
}

void HappinessCalculationAndMeterUpdater()
{
    int avg = 0;
    for (int i = 0; i < ManHolesList.Count; i++)
    {
        avg += (int)ManHolesList[i].GetComponent<ManHoles>().damageMeter;
    }

    avg /= ManHolesList.Count;
    HappinessMeter = avg;
    HappinessMeterSlider.value = Mathf.MoveTowards(HappinessMeterSlider.value, HappinessMeter, changeSpeed * Time.deltaTime);

    if (HappinessMeter > 70)
    {
        moodIndicatorImage.sprite = happy;
        fillColour.color = Color.green;
    }
    else if (HappinessMeter >= 40)
    {
        moodIndicatorImage.sprite = moderate;
        fillColour.color = Color.yellow;
    }
    else if (HappinessMeter < 40)
    {
        moodIndicatorImage.sprite = angry;
        fillColour.color = Color.red;
    }
}

```



```

public class TruckImageManager : MonoBehaviour
{
    public static TruckImageManager instance;
    //Truck Image List
    public List<GameObject> Trucks = new List<GameObject>();

    //White - When the truck is inside the building
    //Gray - When the truck is out on repair
    [SerializeField] Sprite WhiteTruck;
    [SerializeField] Sprite GrayTruck;

    public int TrucksAvailable;
    public int TrucksOutside;
    public int TrucksInside;

    private void Awake()
    {
        instance = this;
    }

    public void ChangeImage()
    {
        if(TrucksAvailable == 1)
        {
            Trucks[0].GetComponent<Image>().sprite = WhiteTruck;
        }
        else if(TrucksAvailable == 2)
        {
            Trucks[1].GetComponent<Image>().sprite = WhiteTruck;
        }
        else if(TrucksAvailable == 3)
        {
            Trucks[2].GetComponent<Image>().sprite = WhiteTruck;
        }
        else if(TrucksAvailable == 4)
        {
            Trucks[3].GetComponent<Image>().sprite = WhiteTruck;
        }
        else if(TrucksAvailable == 5)
        {
            Trucks[4].GetComponent<Image>().sprite = WhiteTruck;
        }
    }
}

```

```

public class Trucks : MonoBehaviour
{
    public GameObject target;
    NavMeshAgent agent;
    GameObject station;
    GameObject truckDisplay;
    [SerializeField] Sprite whiteTruck, GrayTruck;
    [SerializeField] float repairTime;
    float repairTimeReset;

    void Start()
    {
        agent = GetComponent<NavMeshAgent>();
        station = this.transform.parent.gameObject;
        repairTimeReset = repairTime;
        LinkToImage();
    }

    void Update()
    {
        if(target != null)
        {
            truckDisplay.GetComponent<Image>().sprite = GrayTruck;
            agent.destination = target.transform.position;
            if (Vector3.Distance(transform.position, target.transform.position) <= 2)
            {
                //What must happen when the truck gets there
                repairTime -= Time.deltaTime;
                if(repairTime <= 0)
                {
                    target.GetComponent<ManHoles>().RepairManHole();
                    repairTime = repairTimeReset;
                    target = null;
                }
            }
        }
        if(target == null)
        {
            agent.destination = station.transform.position;
            if(Vector3.Distance(transform.position, transform.parent.position) <= 2)
            {
                Station.instance.LineUp(gameObject);
                truckDisplay.GetComponent<Image>().sprite = whiteTruck;
                gameObject.SetActive(false);
            }
        }
    }
}

```

```

}

void LinkToImage()
{
    if(GameManager.instance.TruckCount == 1)
    {
        truckDisplay = GameObject.FindGameObjectWithTag("Image1");
    }
    else if (GameManager.instance.TruckCount == 2)
    {
        truckDisplay = GameObject.FindGameObjectWithTag("Image2");
    }
    else if (GameManager.instance.TruckCount == 3)
    {
        truckDisplay = GameObject.FindGameObjectWithTag("Image3");
    }
    else if (GameManager.instance.TruckCount == 4)
    {
        truckDisplay = GameObject.FindGameObjectWithTag("Image4");
    }
    else if (GameManager.instance.TruckCount == 5)
    {
        truckDisplay = GameObject.FindGameObjectWithTag("Image5");
    }
}

```

```

void Start()
{
    upgradeCost = 2;
    RepairCost = 1;
    damageMeterSlider = gameObject.transform.GetChild(0).gameObject.transform.GetChild(0).GetComponent<Slider>();
    viewCanvas = gameObject.transform.GetChild(0).GetComponent<Canvas>();
    GameManager.instance.population += 200;
    SliderManager.instance.addManHoleToList(gameObject);
}

void Update()
{
    DamageMeterUpdater();
    if(degrade == true && damageMeter > 0)
    {
        StartCoroutine(manHoleDeterioration());
        degrade = false;
    }
    if(damageMeter < 0)
    {
        damageMeter = 0;
    }
}

void DamageMeterUpdater()
{
    viewCanvas.transform.LookAt(Camera.main.transform);
    damageMeterSlider.value = damageMeter;
}

IEnumerator manHoleDeterioration()
{
    yield return new WaitForSeconds(degradeRate);
    damageMeter -= wearAndTear;
    degrade = true;
}

public void RepairManHole()
{
    damageMeter = 100f;
    StopCoroutine(manHoleDeterioration());
    degrade = true;
    //give player Money after repair
    GameManager.instance.coins += (int)repairReward;
}

public void UpgradeManHole()
{
    GameManager.instance.UseCoins(upgradeCost);
    manHoleLevel++;
    degradeRate += 3;
    if(wearAndTear > 3)
    {
        wearAndTear -= 1;
    }
    RepairCost += 3;
}

```

```

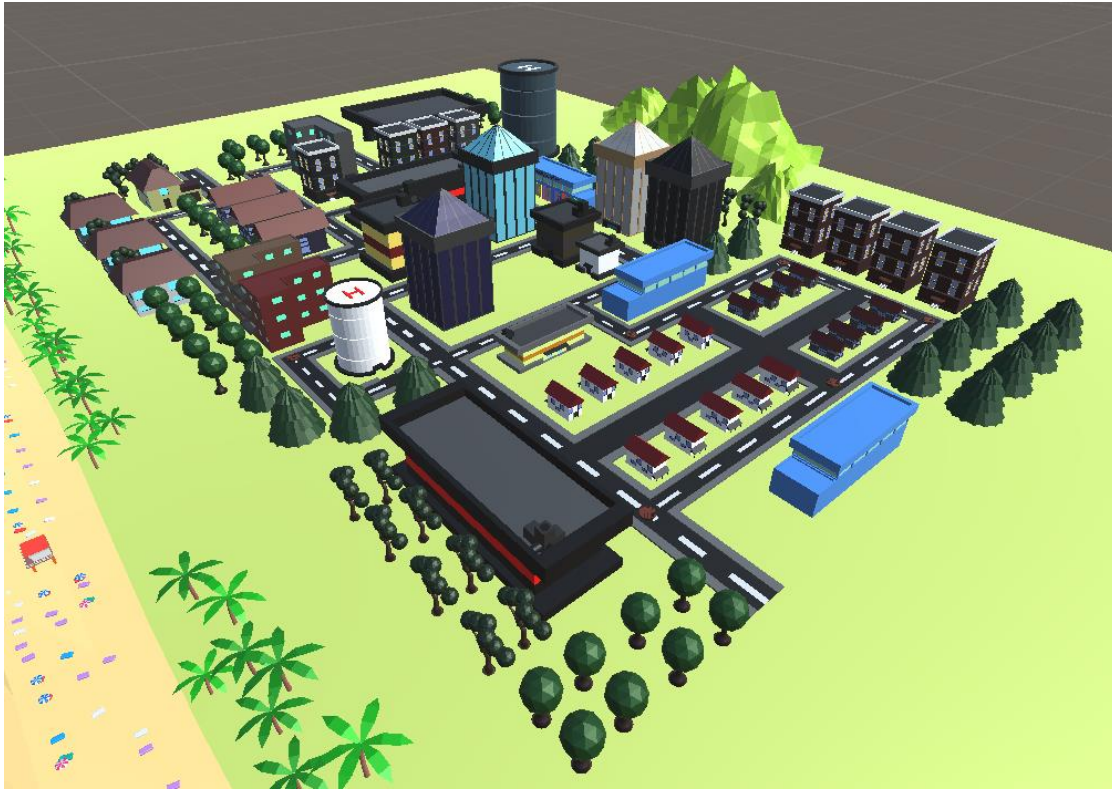
public class ExpantionControler : MonoBehaviour
{
    [SerializeField] List<GameObject> Areas = new List<GameObject>();
    [SerializeField] GameObject stationMovePoint1, stationMovePoint2;
    [SerializeField] GameObject truckStation;
    Vector3 offset = new Vector3(0, 0.5f, 0);
    // Update is called once per frame
    void Update()
    {
        if(GameManager.instance.population >= 3000)
        {
            Areas[0].gameObject.SetActive(true);
            truckStation.transform.position = stationMovePoint1.transform.position + offset;
        }
        if (GameManager.instance.population >= 6000)
        {
            Areas[1].gameObject.SetActive(true);
            truckStation.transform.position = stationMovePoint2.transform.position + offset;
        }
    }
}

```

3D Art

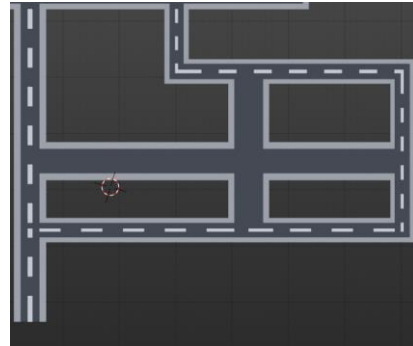
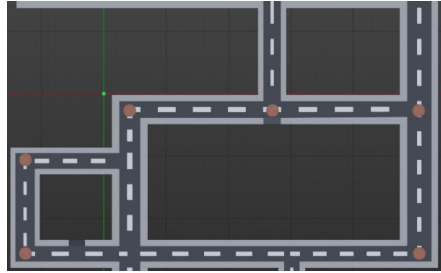
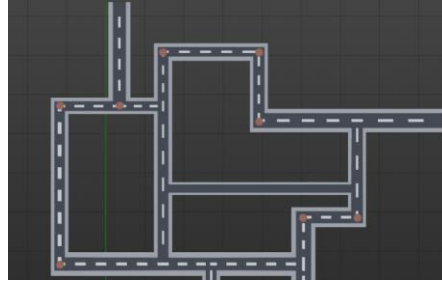
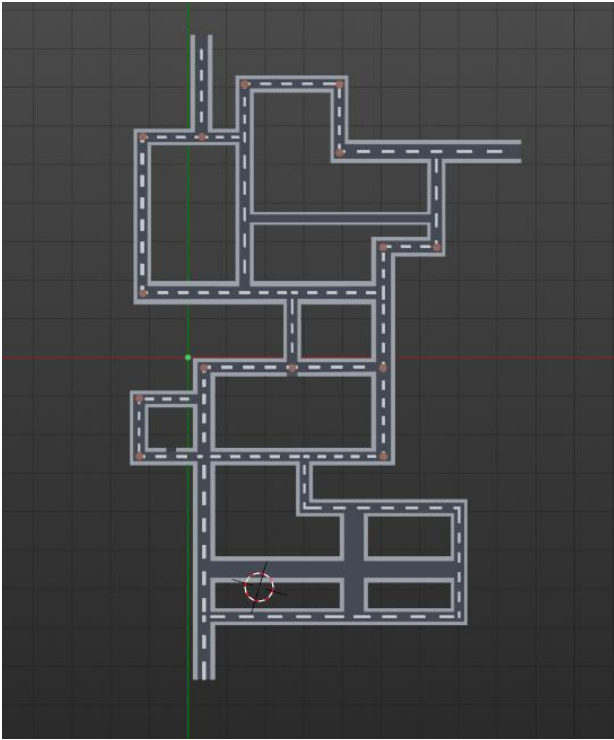


The City



3D Low Poly Objects

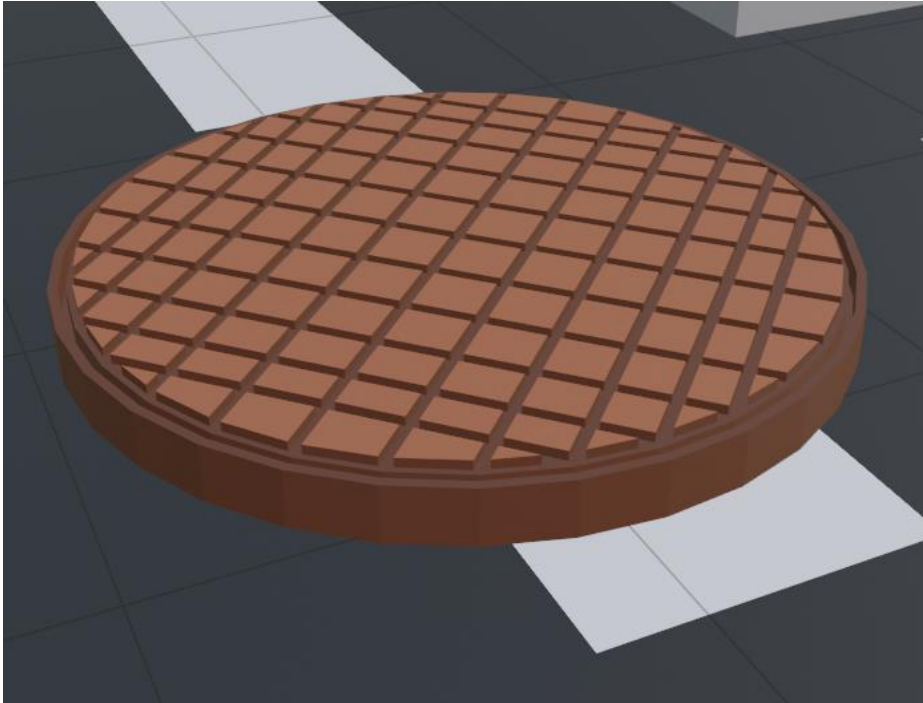
The Layout



- Player Engagement
- Expansion Design
- Areas

3D Low Poly Objects

Manholes



- The problem
- Happiness influencer
- Multiple in number

3D Low Poly Objects

OurCityOurWater Building

- Player Visualisation
- Spawn for trucks
- Colour Scheme
- Relocation



3D Low Poly Objects

Truck

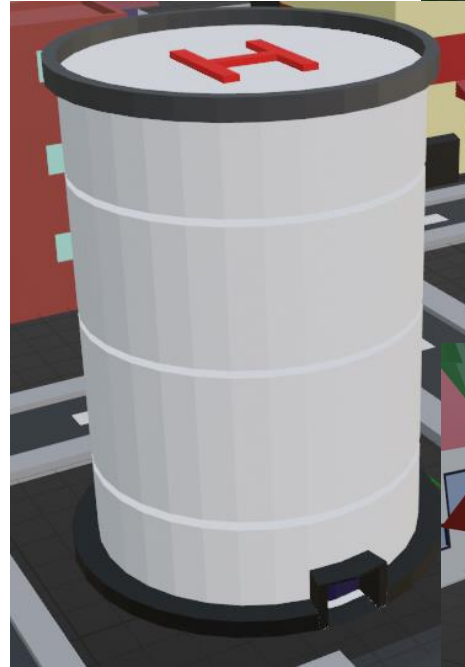
- The Solution
- Player Character
- Total of five
- Upgradeable



3D Low Poly Objects

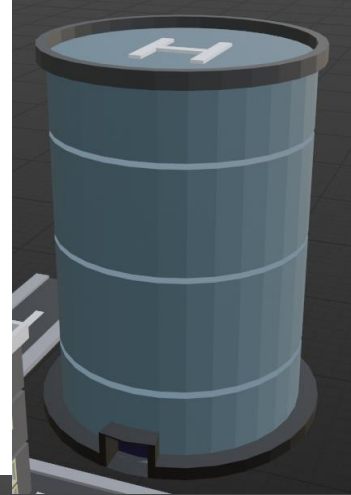
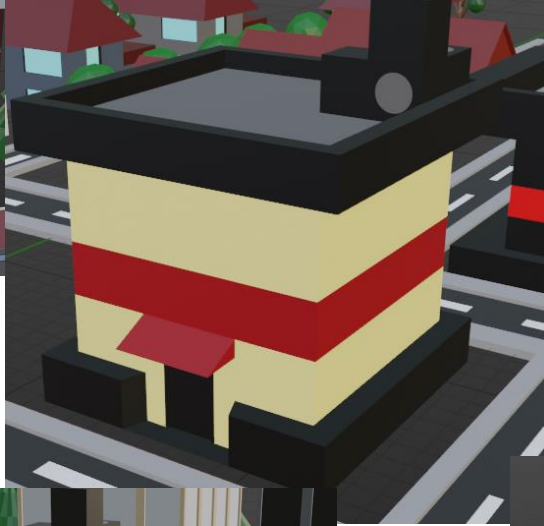
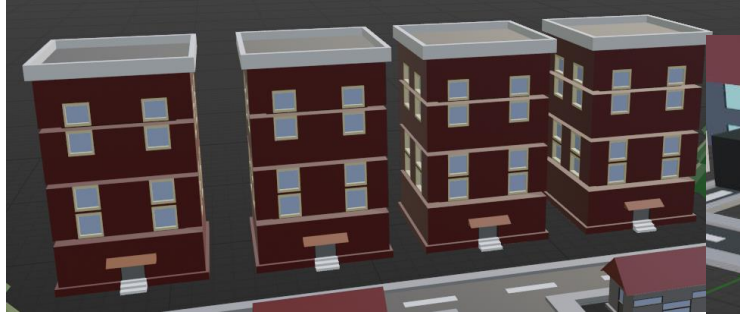
Buildings

- Sense of Reality
- Player Satisfaction
- Variation
- Correlation with area



3D Low Poly Objects

Buildings

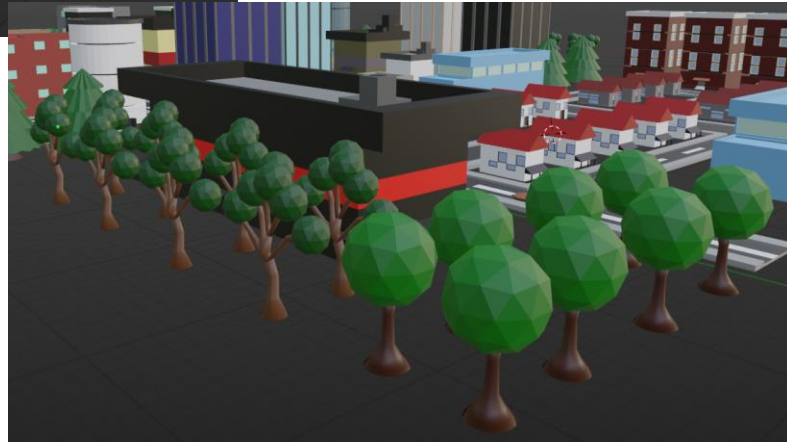


- Sense of Reality
- Player Satisfaction
- Variation
- Correlation with area



3D Low Poly Objects

Foliage



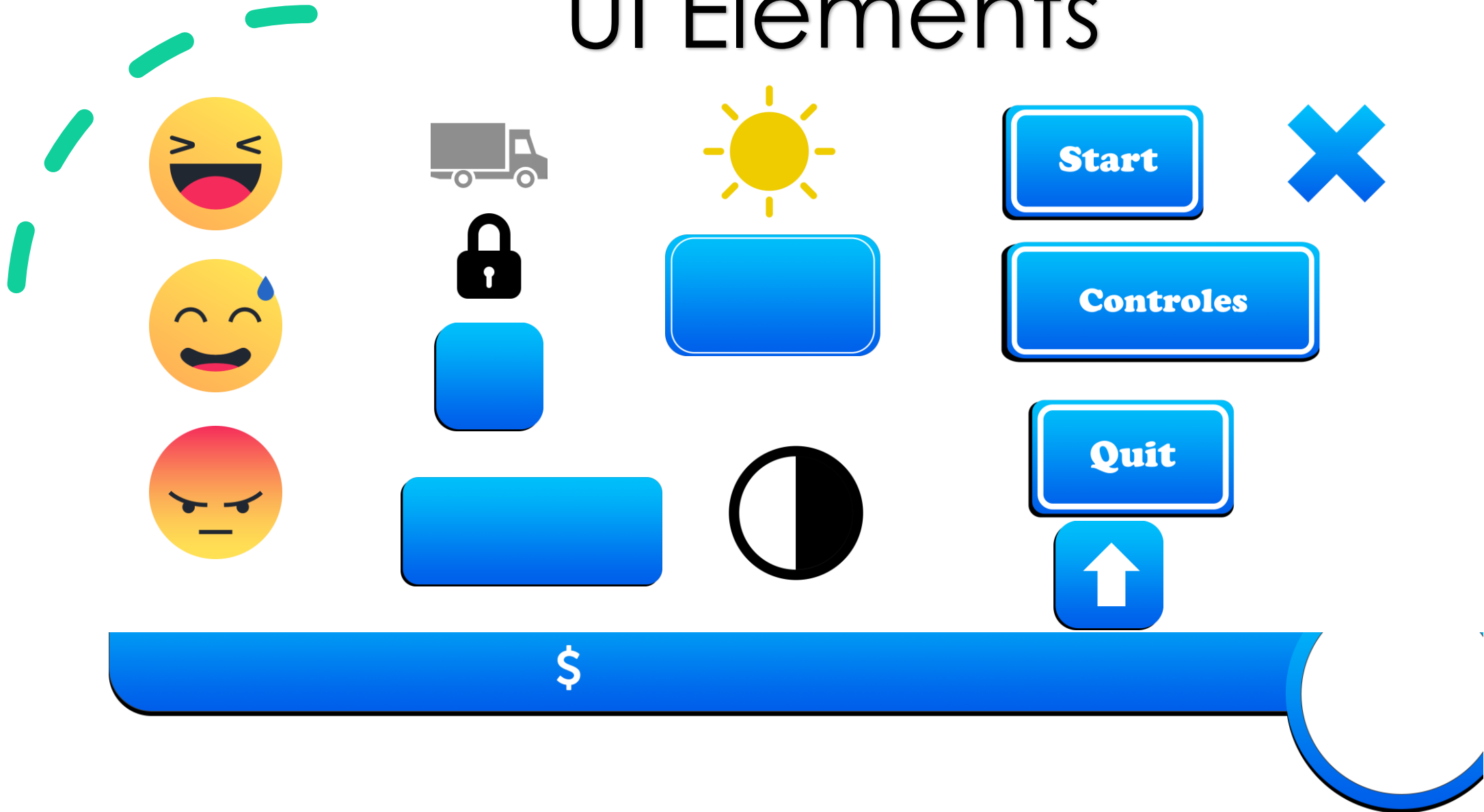
- Creative Decision
- Visualisation for players

3D Low Poly Objects

User Interface



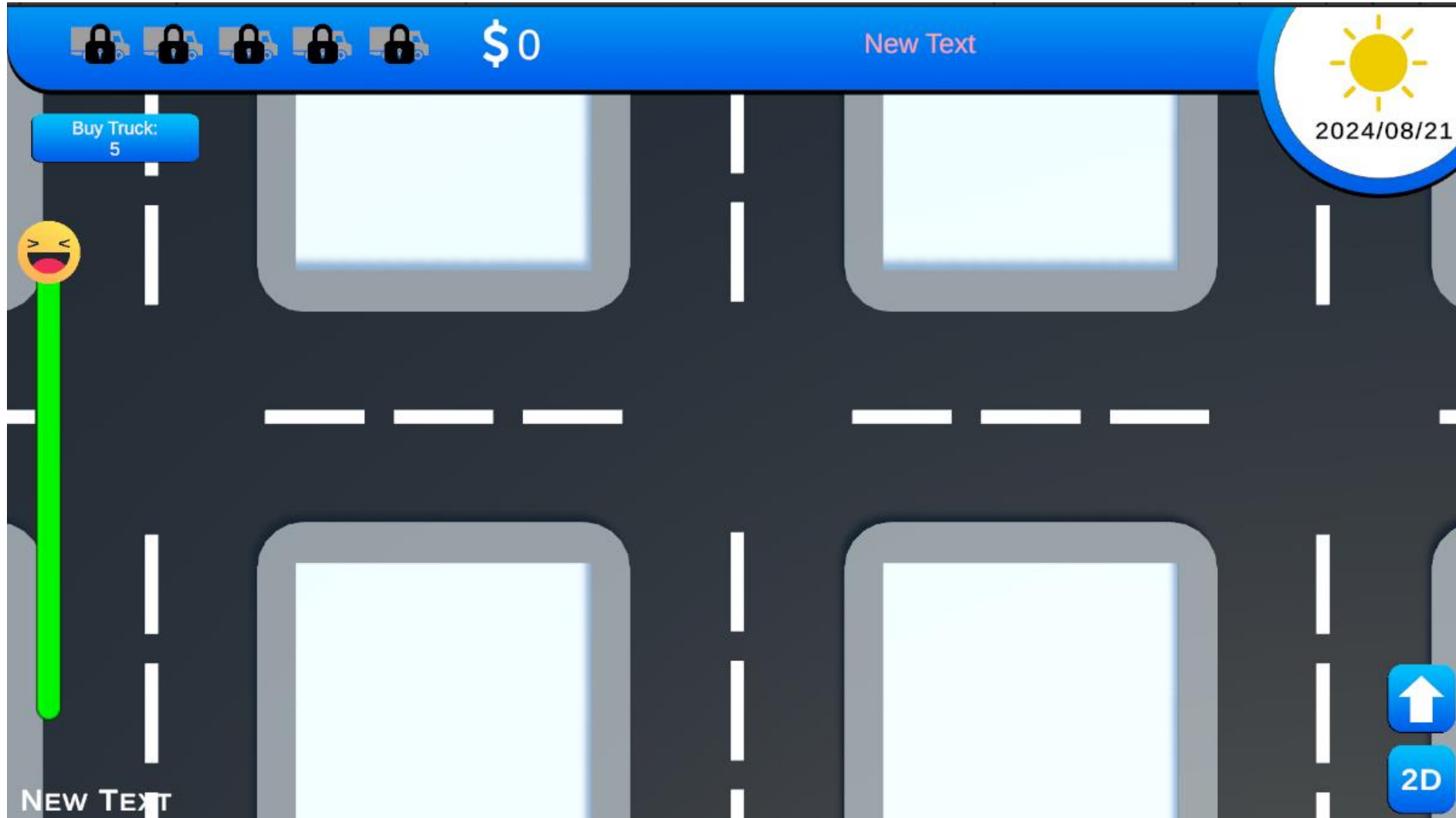
UI Elements



Main Menu



Game Scene





Audio



Music And Sound Effects

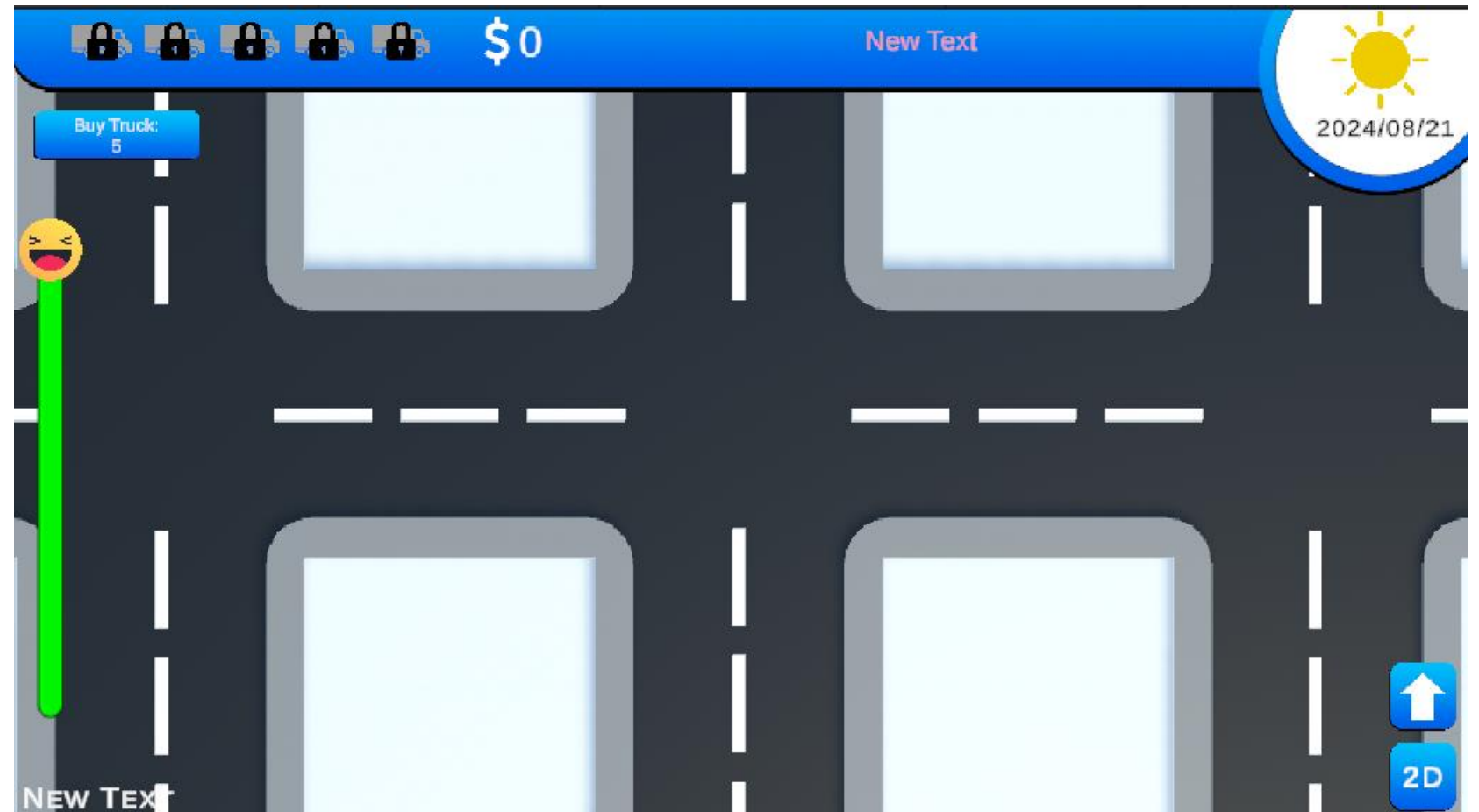
Background: Main Menu



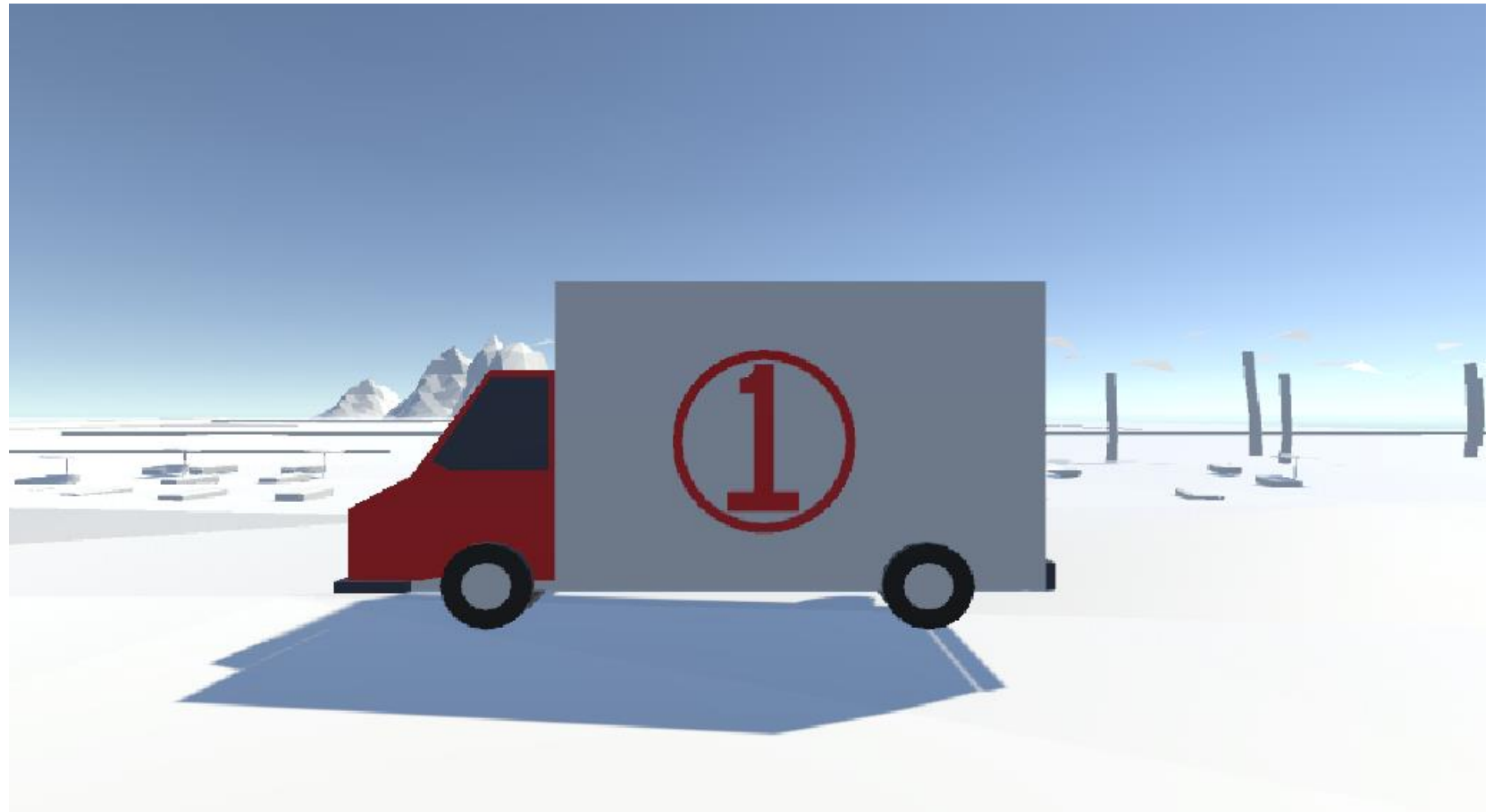
Button Click: Main menu and Gamplay



Background: Gameplay



Gameplay: Truck Noise





What we are
asking for:

The background features a large blue semi-circle on the left side. In the top left corner, there is a small green circle and a yellow triangle. In the top right, there is a large green circle and a vertical yellow dashed line. In the bottom right, there is a yellow square and a diagonal yellow dashed line.

Thank you for
listening