

AR Dragon

1. Propunerea temei de proiect

Proiectul propus constă în dezvoltarea unui joc de realitate augmentată (RA) pentru platforma Android, utilizând Unity și ARCore. Jocul are ca scop plasarea unor capre virtuale pe o suprafață detectată de sistem, urmate de apariția unui dragon care lansează bile de foc. Jucătorul poate interacționa cu caprele prin rotire, scalare și mutare pe cele trei axe. Obiectivul principal al jocului este eliminarea tuturor caprelor de pe hartă, iar progresul este contorizat printr-un sistem de scor. Această temă de proiect este aleasă datorită popularității realității augmentate în jocuri și potențialului său de a oferi experiențe interactive captivante. Jocul propune un mediu dinamic, combinând elemente de strategie și acuratețea recunoașterii suprafețelor din ARCore pentru o experiență realistă.

2. Analiza și specificarea soluției

2.1 Descrierea și analiza jocului

Jocul se desfășoară într-un mediu de realitate augmentată, utilizând Unity și ARCore pentru a recunoaște suprafețe plane. Jucătorul plasează capre pe această suprafață, iar un dragon virtual apare în fața camerei. Dragonul are o animație de zbor continuu și poate declanșa un atac la comanda utilizatorului. Atunci când un proiectil lovește o capră, aceasta este eliminată, iar scorul jucătorului crește. După ce toate caprele sunt eliminate, jucătorul primește o notificare de finalizare. Mecanica jocului este simplă, dar oferă un nivel ridicat de interactivitate prin posibilitatea manipulării caprelor în timp real.

2.2 Prototipizarea inițială (schița pe hârtie)



Această schemă conține așezarea în pagină a jocului, pe mijlocul ecranului ar trebuie sa fie imaginea camerei, care detectează planuri, precum și elementele virtuale dragonul și caprele.

2.3 Descrierea scenariilor de joc RA

Scenariul principal presupune plasarea caprelor pe suprafața detectată de ARCore, urmată de apariția dragonului. La apăsarea butonului "Attack", dragonul lansează o bilă de foc spre capre. Dacă bila lovește o capră, aceasta este eliminată, iar scorul este actualizat. Odată ce toate caprele sunt distruse, apare un mesaj de notificare a câștigului.

3. Proiectarea jocului RA

3.1 Scena de obiecte reale și virtuale

Scena de joc include:

- Suprafețe reale detectate cu ARCore;
- Capre virtuale care pot fi mutate, scalate și rotite;
- Dragonul care apare în fața camerei și atacă caprele;
- Bile de foc lansate de dragon.

3.2 Strategia de joc

Jucătorul trebuie să plaseze strategic caprele în așa fel încât să reziste atacurilor dragonului. Această mecanică introduce un element de strategie, oferind o experiență interactivă și captivantă.

3.3 Tehnici de interacțiune utilizator

- **Meniu principal (jos, centru):** Selectare capră
- **Plasare obiecte:** După detectarea unui AR Plane, caprele pot fi plasate
- **Interacțiune cu capra:**
- **Meniu de editare:** Apare la atingere
- **Opțiuni:** Transformare, scalare, rotire, ștergere
- **Butoane laterale (dreapta):**
- **Reset:** Resetează scorul și reînvie caprele
- **Start:** Face dragonul să apară și afișează joystick-ul
- **Attack:** Dragonul lansează bile de foc la apăsarea butonului
- **Joystick (stânga):** Controlează mișcarea dragonului

3.4 Tehnici de interacțiune între obiecte

- Detectarea coliziunii bilelor de foc cu caprele;
- Animarea morții caprelor și actualizarea scorului;
- Afișarea și actualizarea scorului
- Afișarea notificării de finalizare a jocului.

4. Considerații de implementare

4.1 Prezentarea și motivarea soluțiilor alese

Unity și ARCore oferă un mediu robust pentru implementarea realității augmentate. Sistemul de fizică Unity permite coliziuni realiste, iar interfața intuitivă asigură o experiență plăcută pentru utilizatori.

4.2 Implementarea scenelor de obiecte reale și virtuale

Scena de joc este construită folosind detecția de suprafațe **ARCore**.



Planul este înconjurat de un contur negru, de asemenea am folosit un free asset ca să aplic un material transparent, dar care generează umbre **Shadow Receiver URP**.

4.3 Implementarea interacțiunii utilizator cu obiecte virtuale

Pentru a plasa caprele, am folosit un alt free asset care se numește **AR Magic Bar Lite**. Caprele, precum și animațiile lor au fost preluate din pachetul **Animated Goat and Sheep- 3D lowpoly-FREE**.

Dragonul este plasat în mod dinamic în fața camerei, cu fața spre cameră cu ajutorul unui script scris de mine denumit **DragonCreator**. Obiectul cât și animațiile le-am luat din pachetul **Dragon for Boss Monster: HP**. Mișcarea dragonului o fac cu ajutorul unui Fixed JoyStick, obiect importat din pachetul **Joystick Pack**, dar și cu ajutorul scriptului **DragonController**.



Pentru mișcarea mingii de foc, am scris scriptul **ShootBall**.

4.4 Implementarea interacțiunii obiectelor virtuale cu obiecte reale

Coliziunile sunt gestionate de motorul fizic Unity. Pentru asta a trebuit să mă asigur că obiectele au o componentă de Rigidbody și că au și o componentă de Collider.

4.5 Implementarea scenariilor de joc RA

Logica jocului a fost scrisă în câteva scripturi, de asemenea am și unele importate.

Mare parte a logicii a fost scrisă în C# scripts, care au fost atașate unui gameObject empty cu denumirea de GameLogic, exemple de scripturi:

```
public class ShootBall : MonoBehaviour
{
    [SerializeField] private GameObject ball;
    [SerializeField] private float ballForwardForce = 500f;

    void Start()
    {
        UIButtonHandler.OnShootButtonClick += ShootBallOnClick;
    }

    private void ShootBallOnClick()
    {
        GameObject dragon = GameObject.FindWithTag("BlueDragon");
        Vector3 spawnPosition = dragon.transform.position + dragon.transform.forward
        * 0.1f;
        spawnPosition.y += 0.4f;
        Quaternion spawnRotation = dragon.transform.rotation;
```

```

        Debug.Log(spawnPosition);

        GameObject spawnedBall = Instantiate(ball, spawnPosition, spawnRotation);
        Rigidbody rb= spawnedBall.GetComponent<Rigidbody>();
        Vector3 modifiedForward = Quaternion.AngleAxis(20f, dragon.transform.right)
* dragon.transform.forward;

        if (rb != null )
        {
            rb.AddForce(modifiedForward * ballForwardForce);
        }

        Destroy(spawnedBall, 5f);
    }
}

```

Funcția de identificare a unei coliziuni cu o capră, această funcție face parte dintr-un script care este atașat mingii de foc.

```

private void OnTriggerEnter(Collider other)
{
    if(other.gameObject.CompareTag("BlackGoat"))
    {
        Animator animator = other.gameObject.GetComponent<Animator>();
        if (animator != null && !animator.GetBool("IsDead"))
        {
            animator.SetBool("IsDead", true);
            KillManager.Instance.IncreaseKillCount();
        }
    }
}

```

5. Evaluarea jocului

5.1 Evaluarea corectitudinii funcționale

Pentru a verifica buna funcționare a jocului, s-au efectuat teste asupra interacțiunii dintre utilizator, dragon și capre. Coliziunile dintre bilele de foc și capre au fost atent monitorizate pentru a asigura eliminarea corectă a caprelor și actualizarea scorului. De asemenea, s-au testat toate butoanele din interfață pentru a evita erori de funcționalitate.

5.2 Evaluarea utilizabilității

Pentru a îmbunătăți experiența utilizatorului, am cerut feedback de la mai mulți utilizatori care au testat jocul pe diferite dispozitive Android. Aceștia au oferit sugestii privind plasarea și manipularea caprelor, utilizarea joystick-ului pentru controlul dragonului și claritatea interfeței. Pe baza acestui feedback, am ajustat sensibilitatea comenzilor și am optimizat meniurile pentru o

utilizare mai intuitivă. În plus, am analizat alte jocuri similare de realitate augmentată pentru a identifica cele mai eficiente metode de interacțiune.

5.3 Evaluarea performanțelor

Testele de performanță au vizat optimizarea resurselor grafice și reducerea consumului de baterie. Am monitorizat utilizarea procesorului și a memoriei RAM pentru a ne asigura că jocul rulează fluent fără întreruperi sau întârzieri. De asemenea, s-au efectuat teste pe diferite modele de telefoane Android pentru a garanta compatibilitatea și stabilitatea aplicației. Optimizările realizate au contribuit la îmbunătățirea fluidității animațiilor și a timpului de răspuns al interacțiunilor.