ORIGINALAN NAUČNI RAD – ORIGINAL SCIENTIFIC PAPER 10.7251/ZRPIM2101508L

RAZVOJ PROGRAMSKE PODRŠKE ZA IGRU „SEAWAYS"

Blagodar Lovčević[1], Miodrag Milićević[1], Nikola Marković[1], Marko Boljanić[1], Mladen Lovčević[2]

[1]Akademija strukovnih studija Šabac, Dobropoljska 5, 15 000 Šabac, Srbija,
blagodarlovcevic@gmail.com
[2]Galeb Group d.o.o, Pocerska 111, 15 000 Šabac, Srbija

SAŽETAK

Rad je posvećen razvoju programske podrške za ulogu glavnog igrača u igri Seaways, realizovan u Unity Game Engine-u korišćenjem programskog jezika C#. Na početku rada date su osnovne karakteristike Unity Game Engine-a. Zatim, u radu je prikazan postupak kreiranja korisničkog interfejsa i programskog koda koji obezbeđuje funkcionisanje igre. Pored Unity Game Engine-a, korišćeni su i sledeći programi: Blender, GIMP i Audacity. Igra je namenjena igračima svih uzrasta i služi da se korisnik programa (igrač) zabavi u slobodno vreme. Kompjuterske igre postaju sve kompleksnije, jer ciljna populacija korisnika nisu samo deca, već i odrasli, koji prema različitim istraživanjima čak i više vremena od dece provode igrajući se. U radu dajemo i predlog daljeg unapređenja igre.

Ključne reči: Programska podrška, Kompjuterska igra, Unity Game Engine.

UVOD

Prvenstvena svrha kompjuterskih igara jeste zabava za korisnike, ali i zarada za kreatore igre. Istraživanja su pokazala da je prosečna starost gejmera 34 godine. Video igre pozitivno utiču na zadovoljstvo krajneg korisnika, na razvoj njegovih kognitivnih sposobnosti, kao i na njegovu kreativnost i opažanje. Ljudi koji igraju igre donose odluke znatno brže, uz približno istu tačnost.

UNITY GAME ENGINE

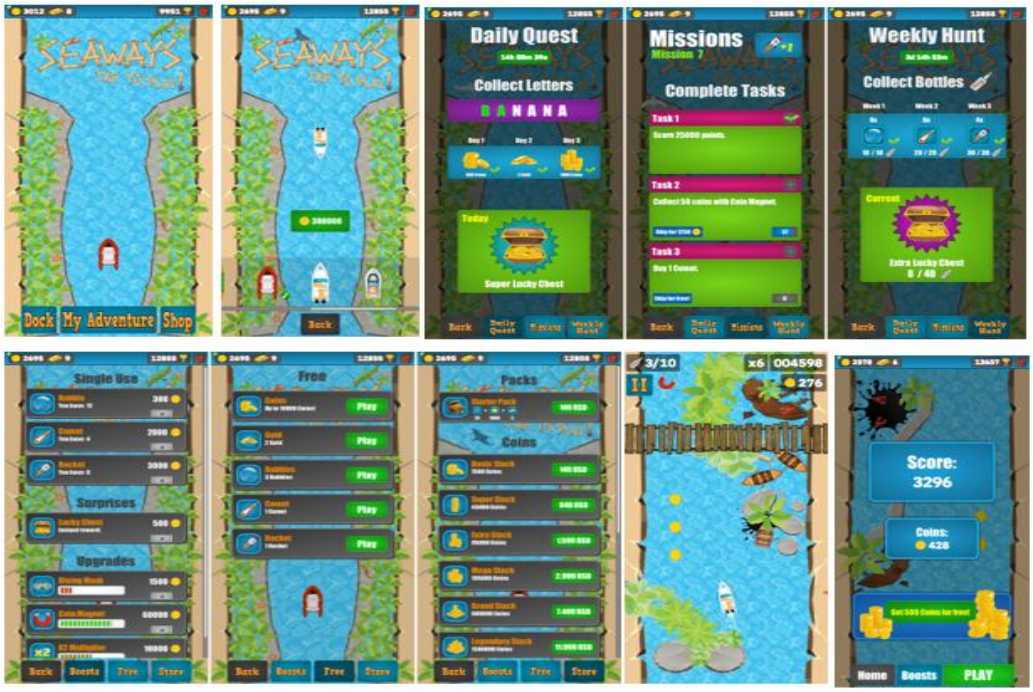
Unity je game engine koji je razvijen od strane kompanije Unity Technologies, a prvi put je najavljen i objavljen u junu 2005. godine na svetskoj konferenciji za razvojne kompanije Apple kao Mac OS – X gane engine, a kasnije je dodata podrška za Microsoft Winndows i Web pretraživače. Unity 2.0 lansiran je 2007. godine, i uključivalo je optimizacije za detaljna 3D okruženja, dinamičke senke u realnom vremenu, smerna svetla i reflektore, reprodukciju videa i mrežni sloj za kreiranje multiplayer igara. Unity 3.0 je lansiran 2010. godine sa funkcijama koje proširuju grafičke karakteristike za desktop računare i konzole. Pored Android podrške, _Unity 3.0 je između ostalog sadržao „Illuminate Labs' Beast Lightmap" alat za odloženo prikazivanje, ugrađeni uređivač stabala, prikazivanje izvornih fontova, automatsko UV mapiranje i audio filter. Unity 4.0 je lansiran 2012. godine i sadržao je podršku i za DirectX i Adobe Flash, kao i alate za animaciju pod nazivom „Mecanim" i pristup Linux-u. Unity 5.0 je ponudio preglede svetlosnih mapa, Unity Cloud, novi audio sistem i Nvidia PhysX 3.3 engine. Unity 5.6 je doneo nove efekte osvetljenja, ažurirao ukupne performanse i dodao podršku za Nitendo Switch, Facebook Gameroom, Google Daydream VR i Vulkan graphics API. Predstavio je i 4K video plejer koji može da pušta snimke u 360 stepeni u VR modu. Nakon Unity 5.6 verzije, postoji verzija Unity 2017.1 koja je sadržala alate za grafičko prikazivanje u realnom vremenu, izgradnju svetova, analitiku operacija ućivo i izveštavanje o performansama. Unity 2017.2 dobija nove alate kao što su „Timeline" koji omogućava „drag-and-drop" sistem kod animacija, „Cinemachine" i pametni sistem kamera. Unity 2017.2 je integrisao #D Max i Maya alate.

Od 2018. godine proširen je na više od 25 platformi. Unity se može koristiti za kreiranje trodimenzionalnih svetova, dvodimenzionalnih svetova, virtuelnih realnosti i igara proširene stvarnosti, kao i simulacija i drugih doživljaja. Pored industrije video igara, Unity je usvojen i u

ostalim industrijama kao što su: film, automobilska industrija, arhitektura, inženjering i građevinarstvo. Unity Game Engine je pokrenut sa ciljem da „demokratizuje“ razvoj igara tako što će ga učiniti dostupnim većem broju programera.

IGRA „SEAWAYS“

Igra je napravljena za Android platformu pomoću Unity Game Engine-a. Igra je potpuno F2P (free to play), s tim što postoji mogućnost da korisnik za pravi novac kupi sredstva za plaćanje u igri ili da odgleda reklamu da bi dobio manju količinu sredtava. U opisu korisničkog interfejsa, posebno ističemo forme: Početni ekran, Prodavnica, Gejmplej i Rezultat igre.



Slika 1. Korisnički interfejs.
Figure 1. User interface.

Kada igrač pokrene igru, pojavljuje se početni ekran na kojem se nalaze tri ponuđena dugmeta. Ako igrač dodirne dugme „Dock“, pojavljuje se panel sa nekoliko brodova koji mogu da se kupe. Ako igrač dodirne dugme pod nazivom „My Adventure“, otvara se panel sa misijama i dnevnim i nedeljnim zadacima koje treba da reši: „Daily Quest“, „Missions“ i „Weekly Hunt“. Forma „Daily Quest“ se bazira na sakupljanju slova tokom igranja. Napravljene su reči koje se sastoje od slova koje igrač sakuplja tokom igre, i one se generišu svakog dan, pa samim tim igraču se daje povod da svakodnevno posećuje igru bi dobijao vredne nagrade. Forma „missions“ je deo prodavnice koji čini da igra ne može da dodsadi igraču, tako što mu stalno daje nove misije koje igrač treba da obavi, i time ostvaruje što bolji rezultat za što kraće vreme (score multiplier). Forma Weekly Hunt“ je napravljena sa ciljem da igraču održi pažnju u većim vremenskim intervalima, jer igrač dobija zadatak da svake nedelje obavi neku radnju u igri i dobije i dobije veće nagrade kao rezultat posete.Dodirom dugmeta „Shop“, igrač ulazi u prodavnicu gde ga čeka veliki izbor „power up-ova“ koji mu pomažu da što lakše savlada prepreke u igri i da ostvare što bolji rezultat. Forme „Boosts“, „Free“ i „Store“ pružaju igraču mogućnost da izabere šta želi da kupi ili nadogradi u igri kako bi omogućio sebi da što lakše savlada prepreke u igri. U formi „Boosts“ igrač može da potroši svoje novčiće kupovinom elemenata koji će mu olakšati igru ili da unapredi elemente koji se sami geberišu ispred igrača u toku igre. U formi „Free“, gledanjem online reklame koja traje u

proseku 15-30 sekundi, igrač dobija element koji želi, besplatno. U formi „Store“, igraču se pruža mogućnost da za realan novac kupi novac u igri.

Dodirom ekrana van dugmića, igrač započinje igru., i brod počinje da se kreće napred. U početku sporijom brzinom, a kako sve više i više odmiče, tako i ubrzava. Na moru se nalaze razne prepreke i neprijateljski brodovi koji se kreću u suprotnom smeru ometajući igrača da se kreće po jednoj traci. Igrač ima mogućnost da se kreće u jednoj od tri trake: leva, desna i srednja. Da bi igrač promenio traku, potrebno je da prevuče prstom preko ekrana, s leva na desno da bi se pomerio za jednu traku u desno ili s densna na levo da bi se pomerio za jednu traku u levo. Cilj igrača je da se što duže zadrži na moru, a da ne udari ni na jednu prepreku i tako ostvari što veći rezultat. Pored prepreka, na moru se nalaze novčići koje igrač treba da sakuplja kako bi mogao da napreduje u igri i razni korisni elementi koji pomažu igraču da ostvari što bolji rezultat. Igra pripada kategoriji „Endless Runner“ igara, što znači da igra nema kraj, odnosno da igrač nikada ne može stići do cilja. Ako igrač udari na neku od prepreka, ima pet sekundi da odluči da li će igru nastaviti sa mesta na kom je izgubio. To može učiniti „odgledanjem“ reklame ili plaćanjem. Ukoliko se igrač ne odluči ni za jednu od ove dve ponude, na ekranu se pojavljuje panel sa postignutim rezultatima i brojem sakupljnenih novčića. Nakon toga, igrač može započeti igru ispočetka.

Programska podrška je napisana u C# programskom jeziku pomoću razvojnog okruženja Visual Studio 2019.

Programski kod za formu početni ekran:

```
using UnityEngine;
using UnityEngine.EventSystems;

public class TapToPlay : MonoBehaviour, IPointerUpHandler
{
   public Scripts s;
   public GameObject[] animals;
   bool play;

   void Start()
   {
      Invoke("Play", 0.5f);
   }

   void Play()
   {
      play = true;
   }

   public void OnPointerUp(PointerEventData data)
   {
      if (play && Vector2.Distance(data.pressPosition, data.position) < 7.5f)
      {
         s.pc.Go();
      }
   }

   void Animals()
   {
      float f1 = Random.value > 0.5f ? -0.1f : 1.1f;
      float f2 = Random.Range(-0.1f, 1.1f);
      Vector2 v2 = Random.value > 0.5f ? new Vector2(f1, f2) : new Vector2(f2, f1);
```

```
        GameObject g = Instantiate(animals[Random.Range(0, 2)], Camera.main.ViewportToWorldPoint(v2), Quaternion.identity);
        Vector2 target = Camera.main.ViewportToWorldPoint(new Vector2(Random.Range(0.1f, 0.9f), Random.Range(0.1f, 0.9f)));
        g.transform.rotation = Quaternion.Euler(Vector3.forward * Mathf.Atan2(target.y - g.transform.position.y, target.x - g.transform.position.x) * Mathf.Rad2Deg);
    }
}
```

Deo programskog koda za formu Gejmplej:

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

public class PlayerController : MonoBehaviour
{
    public Transform[] sea;
    public Transform playerTransform, diveTransform, wordBar, trailPos;
    public GameObject debug, colliders, waves, balloons, star, mudTrail, fireExplosion, waterExplosion, splash, runBar, overallBar, mainButtons;
    public Text totalCoinText, totalGoldText, highScoreText, currentCoinText, currentScoreText, multiplierText, bottlePicked, bottleTotal;
    public Animator anim, camAnim;
    public Scripts s;
    [HideInInspector] public List<Enemy> enemies;
    [HideInInspector] public Transform mudTransform, stormTransform;
    [HideInInspector] public int coin = 0, score = 0, missionMultiplier = 1, x2 = 1, rocket = 0, multiplier;
    [HideInInspector] public float stabilizedSpeed;
    [HideInInspector] public bool forward, accelerate, comet, mask, mud, jump, live, storm, blinking, playing, started;
    Vector2 whirlpoolPos, vVelocity = Vector2.zero;
    public float minSpeed, maxSpeed, speed;
    int t = 3;
    float slowingSpeed, slowingStep, fVelocity = 0f;
    bool island, whirlpool;

    void Start()
    {
        if (s.dd.go)
        {
            Go();
        }
        else
        {
            Status();
            s.ttp.InvokeRepeating("Animals", 7.5f, 7.5f);
            s.dd.ShowInterstitial();
        }
    }

    public void Go()
```

```
{
    s.ttp.CancelInvoke("Animals");
    if (s.dd.tExtra > 170f) s.dd.tExtra = 150f;
    multiplierText.color = s.pUps.c0;
    s.aud.Bubbles();
    started = true;
    playing = true;
    s.panels.tapToPlayPanel.SetActive(false);
    mainButtons.SetActive(false);
    s.bill.coinBill.SetActive(false);
    s.bill.goldBill.SetActive(false);
    s.panels.settingsPanel.SetActive(false);
    overallBar.SetActive(false);
    bottlePicked.text = s.dd.bottlePicked[s.dd.week].ToString();
    bottleTotal.text = "/" + (s.dd.week + 1) * 10;
    missionMultiplier = s.dd.missionMultiplier;
    runBar.SetActive(true);
    forward = true;
    live = true;
    jump = true;
    s.pUps.Invoke("CometRocket", 1f);
    Equipment();
}

public void Reborn()
{
    s.uw.underwater.SetActive(true);
    s.uw.anim.Play("Reborn", 2);
    s.players.currentPlayer.enabled = true;
    anim.enabled = true;
    blinking = true;
    anim.Play("Blinking", 1, 0f);
    anim.Play("Idle", 0, 0f);
    camAnim.Play("Idle", 0, 0f);
    s.com.currentRoute = 1;
    anim.SetFloat("Turn", -1f);
    playing = true;
    live = true;
    forward = true;
    accelerate = true;
    s.com.leftRight = true;
    whirlpool = false;
    mud = false;
    island = false;
    waves.SetActive(true);
    s.aud.anim.Play("Waves Play", 1, 0f);
    s.pause.pauseButton.interactable = true;
    s.pUps.ContinuePowerUps();
    for (int i = 0; i < enemies.Count; i++) enemies[i].ContinueEnemies();
}

public void Status()
{
```

```
        totalCoinText.text = s.dd.totalCoin.ToString();
        totalGoldText.text = s.dd.totalGold.ToString();
        highScoreText.text = s.dd.highScore.ToString();
    }

    void Equipment()
    {
        colliders.transform.GetChild(s.dd.selectedPlayer).gameObject.SetActive(true);
        waves.transform.GetChild(s.dd.selectedPlayer).gameObject.SetActive(true);
        waves.GetComponent<Animator>().enabled = true;
        s.aud.anim.Play("Waves Play", 1, 0f);
        s.pUps.balloon = balloons.transform.GetChild(s.dd.selectedPlayer).gameObject;
    }
```

…..

Deo programskog koda za formu Dock:

```
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.EventSystems;

public class Players : MonoBehaviour, IPointerDownHandler, IPointerUpHandler
{
    public Scripts s;
    public SpriteRenderer currentPlayer;
    public Sprite[] playerSprites;
    public RectTransform content, playersButtons;
    public ScrollRect sr;
    public Scrollbar sb;
    [HideInInspector] public bool aim;
    [HideInInspector] public float oldPos;
    bool fire;
    float newPos;

    void Start()
    {
        int player = int.Parse(currentPlayer.sprite.name);
        sb.value = 0.2f * player;
        content.GetChild(player).GetChild(0).gameObject.SetActive(true);
    }
```

……

Deo programskog koda za formu My adventure:

```
public void Enable()
    {
        CheckDay();
        t = StartCoroutine(Timer());

        if (s.dd.day == 3)
        {
```

```
        bool allDays = true;
        for (int i = 0; i < 3; i++) { if (!s.dd.completedDay[i]) { allDays = false; break; } }

        if (allDays)
        {
           days[3].GetChild(4).gameObject.SetActive(true);
.....
```

Deo programskog koda za formu Rezultat igre:

```
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class Achievements : MonoBehaviour
{
   public Scripts s;
   public GameObject newHighScore;
   public Text scoreText, coinText;
   public Boost[] singleUse, upgrades;
   int targetScore;
   float tRegulate = 1f;

   public void AchievementsPanel()
   {
      if (s.pc.storm)
      {
         s.pc.storm = false;
         s.aud.anim.Play("Storm Stop", 2, 0f);
         s.aud.anim.Play("Chirping Play", 0, 0f);
         s.dd.anim.Play("Music Play", 1);
         s.pc.stormTransform.GetComponent<Animator>().Play("Hide", 1, 0f);
      }

      s.pc.started = false;
      s.panels.achievementsPanel.SetActive(true);
      s.dd.oldTotalCoin = s.dd.totalCoin;
      s.dd.oldTotalGold = s.dd.totalGold;
      s.dd.totalCoin += s.pc.coin;
      scoreText.text = s.pc.score.ToString();
      targetScore = s.pc.score + s.pc.coin * 2;

      if (targetScore > s.dd.highScore)
      {
         s.aud.NewHighScore();
         s.dd.highScore = targetScore;
         s.tasks.Mission(1);
         Instantiate(newHighScore, Camera.main.ViewportToWorldPoint(new Vector2(1.1f,
0.825f)), Quaternion.identity);
      }

      if (s.pc.coin > 0)
      {
```

```
            s.dd.Invoke("Regulate", 1f);
            Invoke("Regulate", 1f);
        }
    }

    void Regulate()
    {
        tRegulate = 0f;
    }

    void Update()
    {
        if (tRegulate < 1f)
        {
            tRegulate += Time.deltaTime * 2f;
            scoreText.text = ((int)Mathf.Lerp(s.pc.score, targetScore, tRegulate)).ToString();
            coinText.text = ((int)Mathf.Lerp(0f, s.pc.coin, tRegulate)).ToString();
        }
    }
.....
```

UNAPREĐENJE I MOGUĆNOSTI ZA DALJI RAZVOJ

Postoji mogućnost da se igra transformiše u igru sa realnim 3D objektima i kao takva da bude još prihvatljivija krajnim korisnicima (Bogosavljević i Tartalja, 2020).

Dalji razvoj igre može biti baziran na marketing miksu: proizvoda, tj. usluge, cene, promocije i distribucije u sklopu prodavnice i prodaje proizvoda u njoj. Pod proizvodom se podrazumeva sve ono što može biti predmet razmene. U ovom slučaju roba je digitalnog tipa, cena treba da bude srazmerna zadovoljstvu korisnika prilikom korišćenja, promocija bazirana pre svega ka ciljnim populacijama i distribucija na društvenim mrežama. Za unapređenje kompjuterske igre je presudno stvaranje novih mogućnosti i njihova neprestana promocija igračima, koji će uvideti kvalitet igre kroz sopstveno iskustvo i zadovoljstvo.

Igra je za sada razvijena za Androuid platformu i nalazi se na Google Play Store-u, na adresi: https://play.google.com/store/apps/details?id=com.dzontraSoft.Seaways

U budućnosti igra će biti razvijana i za ostale platforme i konzole, prvenstveno za IOS mobilne uređaje.

ZAKLJUČCI

Svet kompjuterskih igara je složen i zahtevan. Programiranje kompjuterskih igara predstavlja kreativan i stvaralački rad projektnog tipa. Zadovoljstvo korisnika igre je primarni cilj svake kompjuterske igre.

Unity Game Engine omogućava da se kreira zanimljiv korisnički interfejs, da se funkcionalnost programskog koda lako testira, da se unapređenja lako ugrađuju.

Nedugo, nakon objavljivanja na Google Play Store-u igra je stekla više desetina igrača. Mnogi od njih su fascinirani “Cartoon” grafikom i bojama u igri Seaways.

LITERATURA

Bogosavljević V. i Tartalja, I. (2020). Inkrementalni razvoj 3D video-igre *Arena* kroz praktičan rad studenata. *U Zborniku radova sa 64. Nacionalne konferencije ETRAN (124_RT1.1).* Društvo za elektroniku, telekomunikacije, računrstvo, automatiku i nuklearnu tehniku.

SOFTWARE DEVELOPMENT FOR THE GAME 'SEAWAYS'

Blagodar Lovčević[1], Miodrag Milićević[1], Nikola Marković[1], Marko Boljanić[1], Mladen Lovčević[2]

[1]Academy of Vocational Studies Šabac, Dobropoljska 5, 15 000 Šabac, Serbia,
blagodarlovcevic@gmail.com
[2]Galeb Group d.o.o, Pocerska 111, 15 000 Šabac, Serbia

ABSTRACT

This paper is deducated to the development of software for the role of the main player in the game Seaways, which was created in Unity Game Engine using the C# programming language. At the beginning of the paper, the basic characteristics of the Unity Game Engine are presented. Then, the paper presents the procedure of creating the user interface and the program code which makes sure that the game works. In addition to Unity Game Engine, the following programs were also used: Blender, GIMP and Audacity. The game is intended for players of all ages and it serves to make the user of the program (the player) have fun in one's free time. Computer games are becoming more and more complex, because the target population of users is not only children, but also adults, who according to various research spend even more time playing games than children. In this paper, we also give a proposal for further improvement of the game.

Keywords: Software, Computer Game, Unity Game Engine.