***UNIVERZITET U KRAGUJEVCU FAKULTET TEHNIČKIH NAUKA U ČAČKU***



Game design document (GDD)

*Pixel Runner*

***Studijski program****: Osnovne akademske studije Informacione tehnologije*

***Predmet****: Razvoj digitalnih igara*

***Profesor****:* ***Student****:*

*dr Veljko Aleksić, docent Teodora Popović 78/2018*

*Rade Petrović 100/2018*

*Čačak, 2022. godina*

Sadržaj

[Uvod 3](#_Toc102651010)

[Opis softvera 3](#_Toc102651011)

[Assets 3](#_Toc102651012)

[Audio 4](#_Toc102651013)

[Fontovi 5](#_Toc102651014)

[Prefabs 5](#_Toc102651015)

[Scene 5](#_Toc102651016)

[*Početni meni* 6](#_Toc102651017)

[*Meni sa opcijama* 6](#_Toc102651018)

[*Level 1* 7](#_Toc102651019)

[*Level 2* 9](#_Toc102651020)

[*Završni meni* 12](#_Toc102651021)

[Scripts 12](#_Toc102651022)

[*StartMenu* 13](#_Toc102651023)

[*EndMenu* 14](#_Toc102651024)

[*PlayerLife* 14](#_Toc102651025)

[*ItemCollector* 15](#_Toc102651026)

[*PlayerMovement* 16](#_Toc102651027)

[*PointFollower* 18](#_Toc102651028)

[*StickyPlatform* 18](#_Toc102651029)

[*Rotate* 19](#_Toc102651030)

[*Finish* 19](#_Toc102651031)

[*CameraController* 20](#_Toc102651032)

[*DisplayTime* 21](#_Toc102651033)

[Zaključak 22](#_Toc102651034)

[Literatura 22](#_Toc102651035)

[Reference 22](#_Toc102651036)

[Spisak slika 22](#_Toc102651037)

# Uvod

Pixel runner je 2d platformna igra sa retro grafikom u kojoj je cilj prelazak nivoa uz izbegavanje prepreka i sakupljanje poena tj. jagoda. Glavni karakter se susreće sa preprekama poput rotirajućih, cirkularnih testera koje se pomeraju i šiljaka. Unutar igrice je takođe omogućena funkcija tajmera i penjanja na platforme.

# Opis softvera

U cilju uspešne implementacije igrce, korišćena je Unity Game Engine 2020.3.32f1 LTS (Long Term Support) verzija koja poseduje dobru kompatibilnost sa Visual Studio okruženjem. Kroz Visual Studio pisan je kod pomoću kojeg je odrađen veći deo mehanike igre.

Unity je mehanizam za kreiranje igara kojeg je razvio Unity Technologies. Prvi put je najavljen i objavljen u junu 2005. godine na Svetskoj konferenciji programera kompanije Apple Inc. kao Mac OS X-exclusive engine za igre. Od tada je engine postepeno proširivan tako da podržava razne platforme za desktop, mobilne uređaje, konzole i VR. Posebno je popularan za razvoj iOS i Android mobilnih igara i koristi se za igre poput Heartstone, Monument Valley, Call of Duty: Mobile, Guns of Boom i Cuphead. Unity je popularan za razvoj nezavisnih igara i jednostavan je za upotrebu.

# Assets

U cilju kreiranja igre u Unity okruženju, potrebno je definisati organizaciju arhitekture kako bi kreiranje projekta bilo lakše. Sama arhitektura omogućava lako praćenje i organizaciju fajlova koji služe za razvoj igrice. U Pixel Runner-u, arhitektura je organizovana na sledeći način:

* Assets - sadrži sve neophodne elemente igrice kao što su animacije, zvuci, fontovi, prefabs (prepreke), scene, skripte i teksture.
* Library – sadrži neophodne biblioteke za Unity
* Logs – logovi koji se čuvaju prilikom akcija kao što je nadogradnja paketa
* Packages - sadrži podatke o paketima
* ProjectSettings – podešavanja projekta
* UserSettings – korisnička podešavanja

# Audio

Svaka igrica gubi interesovanje korisnika ukoliko nema zvuk. Pixel Runner koristi 6 audio fajlova:

BGM\_01 – muzika na kraju igrice

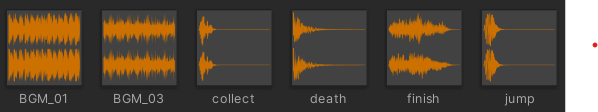
BGM\_03 – muzika na početku igrice

Collect – zvuk prilikom skupljanja poena

Death – zvuk ukoliko heroj dotakne prepreku

Finish – zvuk na završetku nivoa

Jump – zvuk prilikom akcije skoka heroja



Slika 1. Audio

# Fontovi

Kao i zvuk, odgovarajući font doprinosi poboljšanju korisničkog iskustva tokom igranja igrice. Pixel Runner koristi PressStart 2P retro font koji doprinosi osećaju igranja klasične platformne igrice.



Slika 2. Prikaz PressStart 2P retro fonta

# Prefabs

Svaka igra sadrži elemente koji se ponavljaju. Ti elementi se nalaze u prefabs folderu i koriste se za višekratnu upotrebu prilikom dizajniranja nivoa.



Slika 3. Prefabs elementi

# Scene

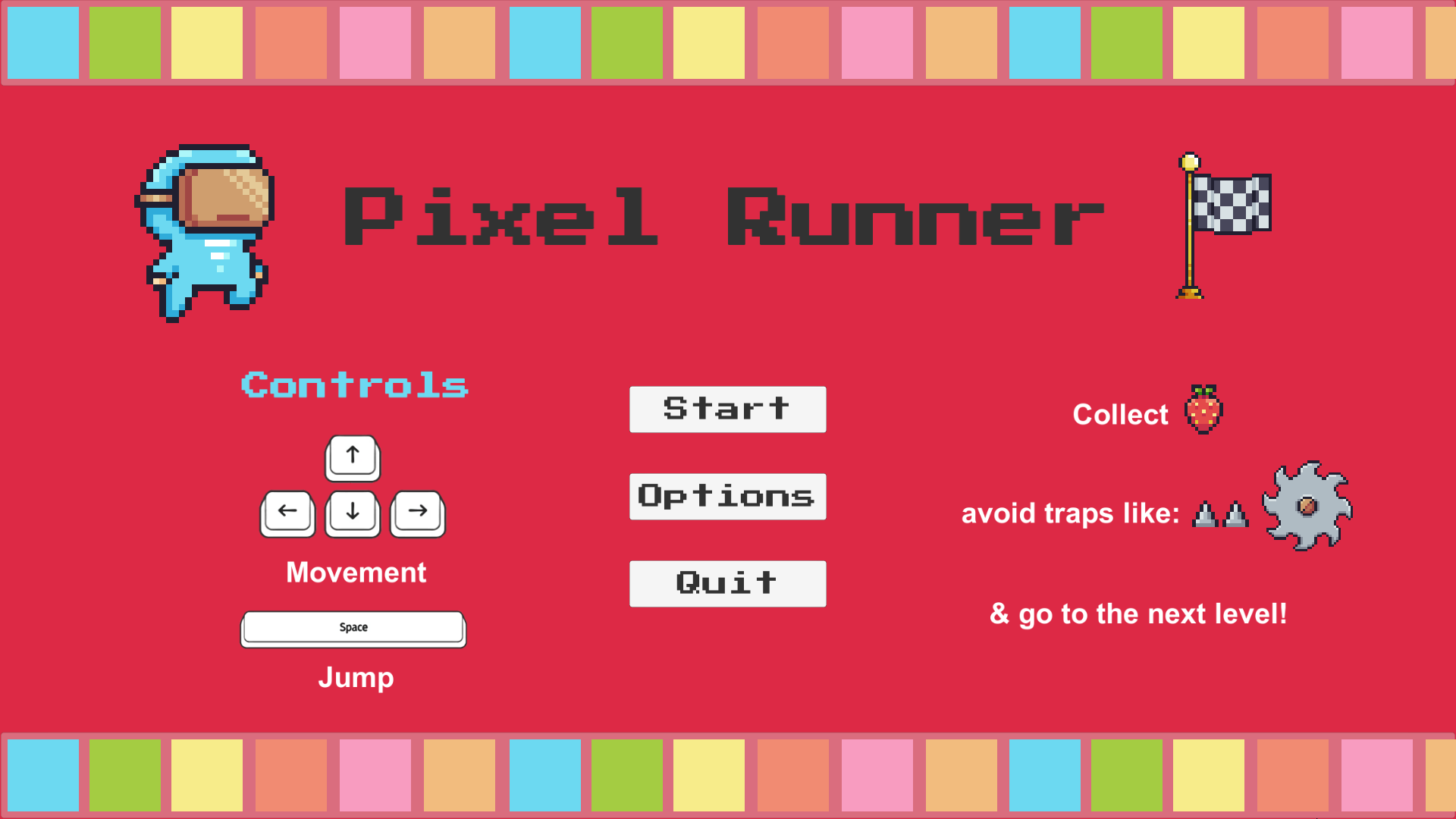
U igri su implementirane različite scene koje predstavljaju tok igre od početka do kraja. Scene pružaju osećaj igraču da se kreće kroz igricu u realnom vremenu. Pixel Runner koristi 4 scene, od čega su dve scene nivoa a dve služe za početni i završni ekran.



Slika 4. Scene Pixel Runner-a

## *Početni meni*

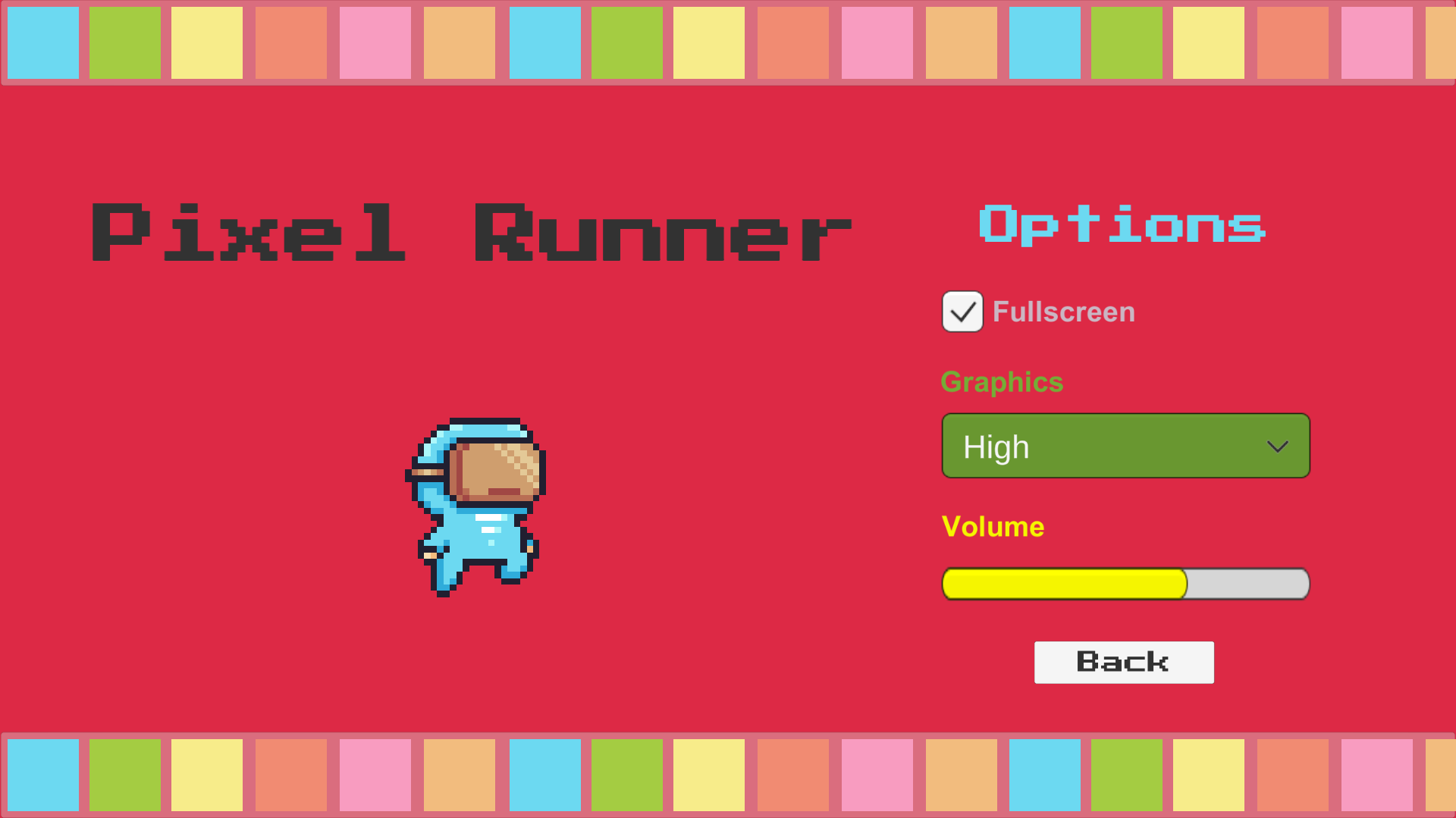
Kada korisnik pokrene igru, prikazuje mu se početni meni gde može započeti sa igranjem, podesiti opcije, ili izaći iz igrice. Dato je i kratko uputstvo, kao i kontrole za igranje.



Slika 5. Početni meni

## *Meni sa opcijama*

Klikom na dugme Options iz glavnog menija dobija se meni sa opcijama za podešavanje celog ekrana, grafike i volumena.



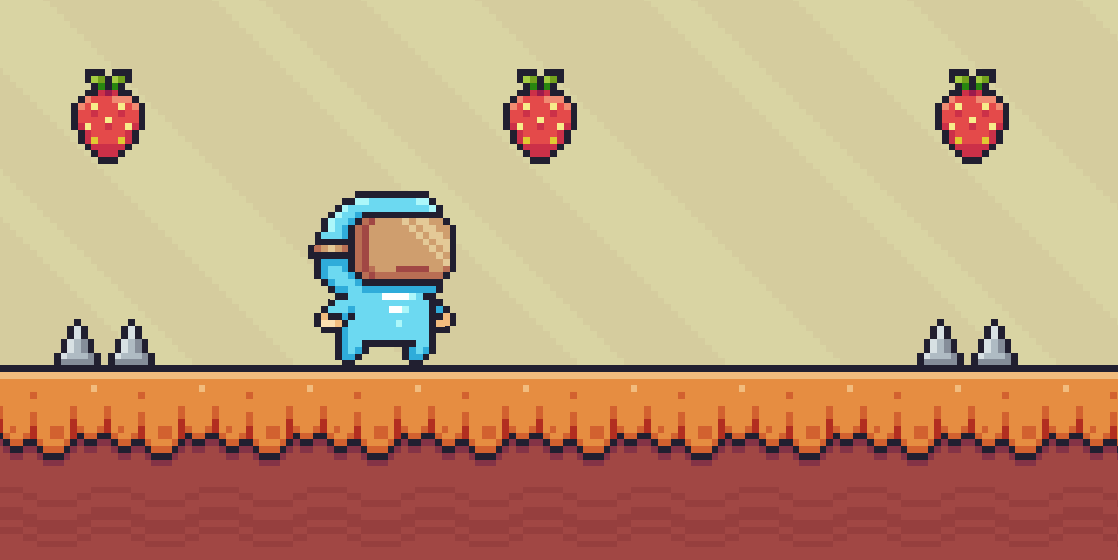
Slika 6. Meni sa opcijama

## *Level 1*

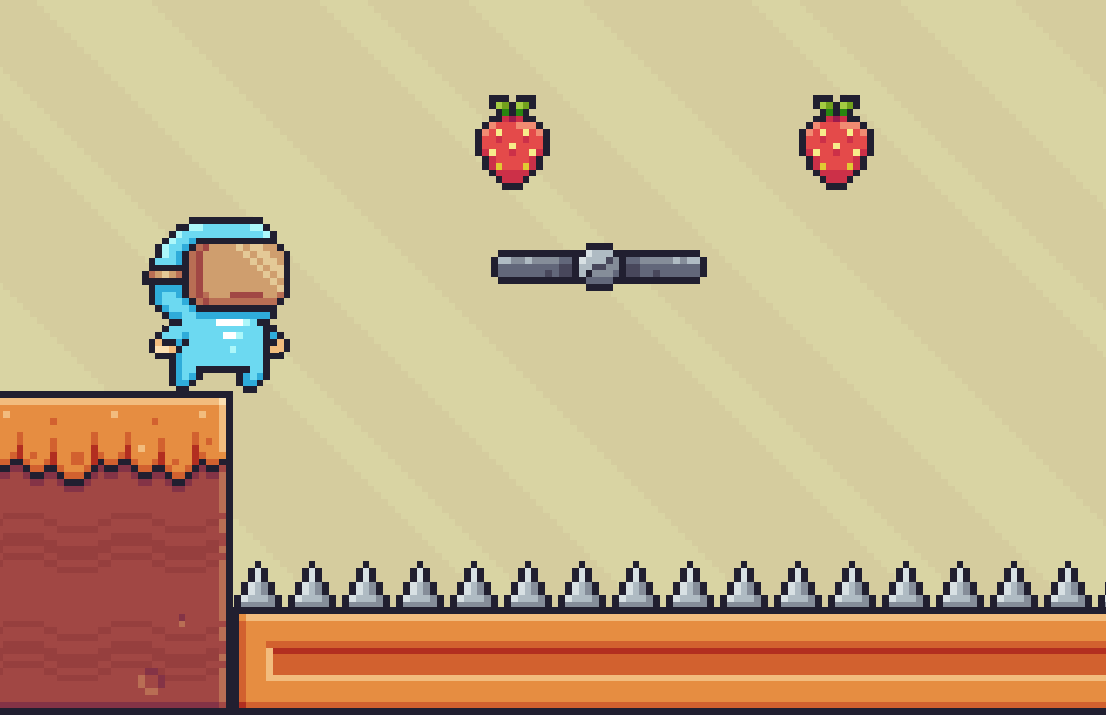
Prvi nivo je jednostavan i služi za upoznavanje korisnika sa igricom. Sadrži osnovnu vrstu prepreka (šiljke) i penjanje na platforme.



Slika 7. Početak nivoa 1



Slika 8. Prepreka sa šiljcima nivoa 1



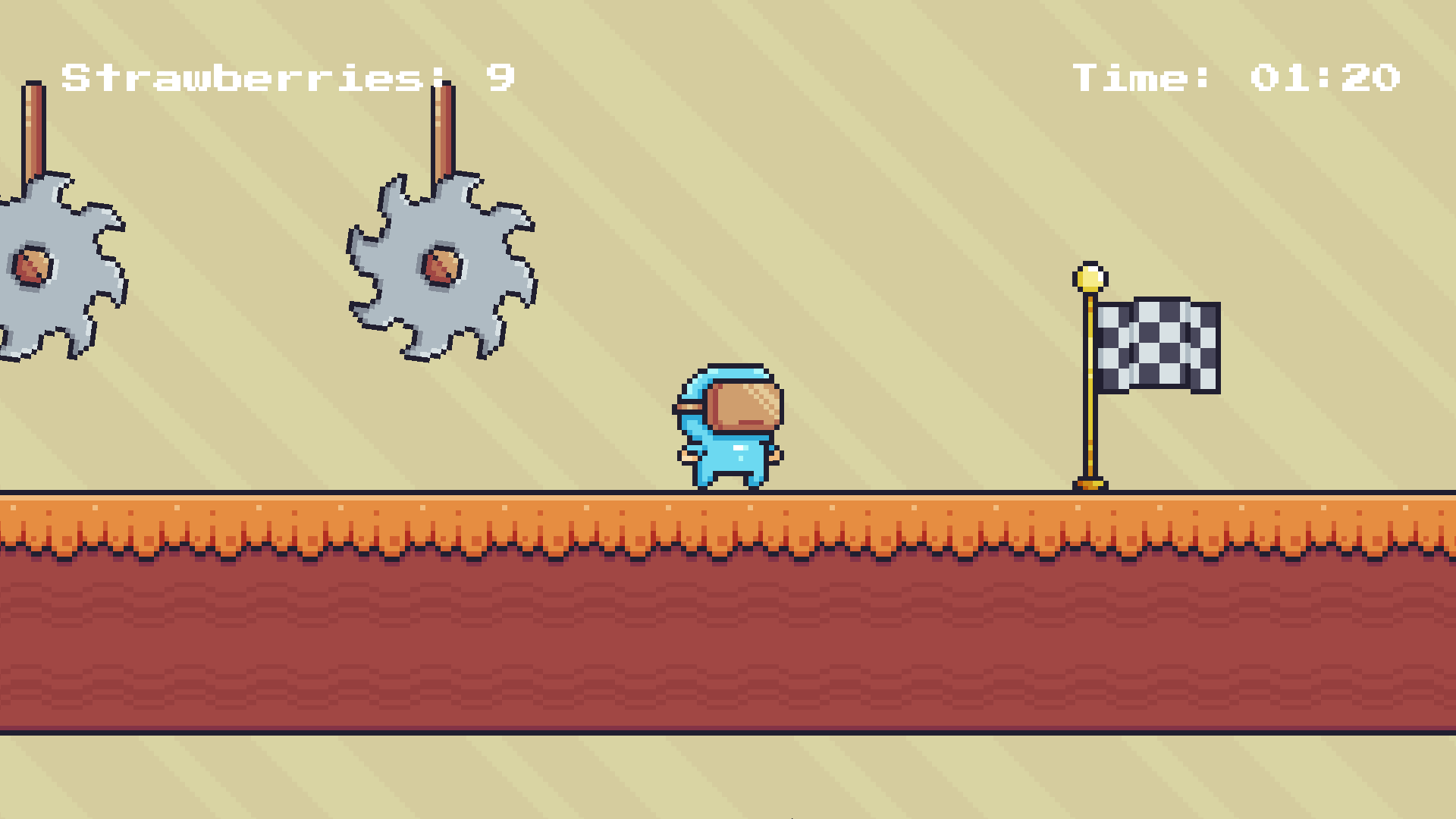
Slika 9. Pomerajuća platforma nivoa 1



Slika 10. Pomerajuća rotirajuća testera nivoa 1



Slika 11. Vertikalno pomerajuće testere nivoa 1



Slika 12. Kraj nivoa 1

## *Level 2*

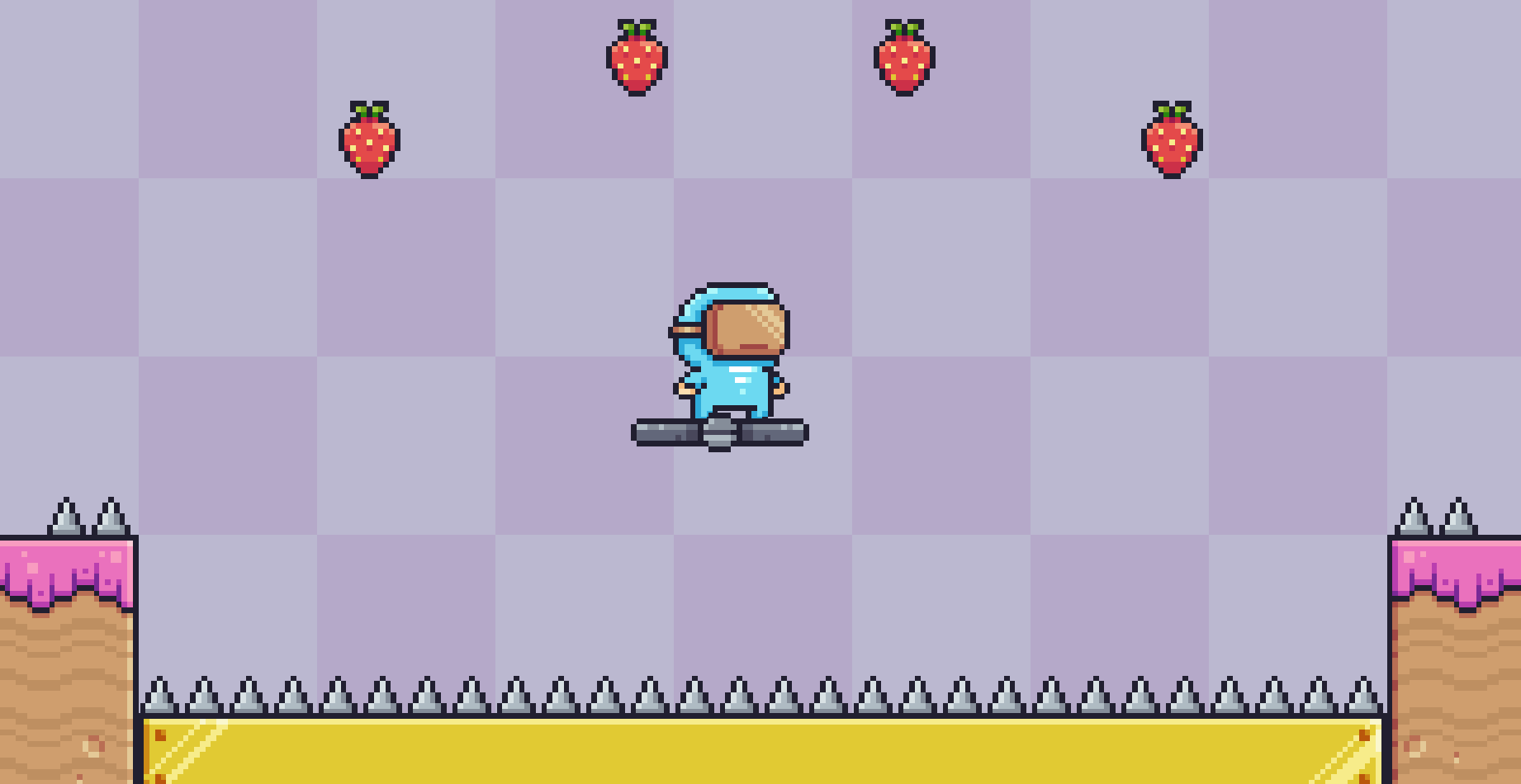
Drugi nivo je teži od prethodnog i sadrži veći broj prepreka.



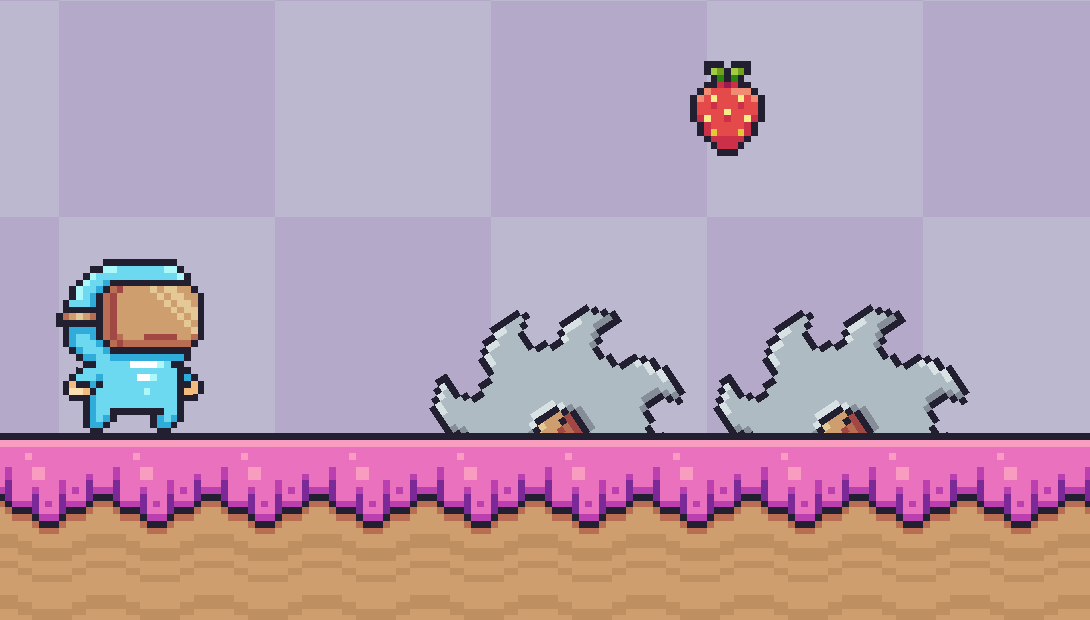
Slika 13. Početak nivao 2



Slika 14. Pomerajuća rotirajuća testera nivoa 2



Slika 15. Pomerajuća platforma nivoa 2



Slika 16. Pomerajuće rotirajuće testere u suprotnim smerovima nivoa 2



Slika 17. Vertikalno pomerajuće testere nivoa 2



Slika 18. Kraj nivoa 2

## *Završni meni*

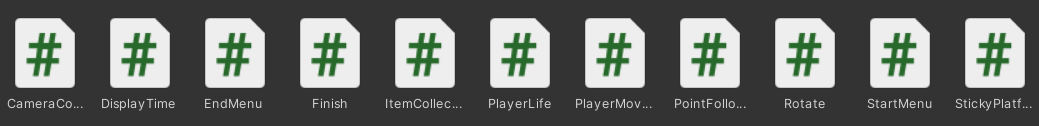
Poslednja scena koja se prikazuje je završni meni koji se prikazuje ukoliko igrač uspešno pređe sve nivoe. Prikazano je i vreme u minutima za koje je korisniku trebalo da pređe oba nivoa.



Slika 19. Završni meni

# Scripts

Skripte su pisane u C# jeziku u Visual Studio okruženju i sadrže mehaniku igre.



Slika 20. Scipts

## *StartMenu*

StartMenu skripta služi za prikazivanje početnog menija i menija sa opcijama.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.Audio;

using UnityEngine.SceneManagement;

public class StartMenu : MonoBehaviour

{

public AudioMixer mainMixer;

public void StartGame()

{

SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);

PlayerPrefs.SetFloat("SessionTime", Time.realtimeSinceStartup);

}

public void Quit()

{

Application.Quit();

}

public void SetFullscreen(bool isFullscreen)

{

Screen.fullScreen = isFullscreen;

}

public void SetVolume(float volume)

{

mainMixer.SetFloat("volume",volume);

}

public void VolumeApply()

{

PlayerPrefs.SetFloat("masterVolume", AudioListener.volume);

}

public void SetQuality(int qualityIndex)

{

QualitySettings.SetQualityLevel(qualityIndex);

}

}

## *EndMenu*

EndMenu skripta služi za izlazak iz igrice.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class EndMenu : MonoBehaviour

{

[SerializeField] private Text TimeText;

private void Start()

{

float sessionTime = Time.realtimeSinceStartup - PlayerPrefs.GetFloat("SessionTime");

int MM = Mathf.FloorToInt(sessionTime / 60.0f);

int SS = Mathf.FloorToInt(sessionTime % 60f);

TimeText.text = "Your total time - " + MM.ToString("00") + ":" + SS.ToString("00");

}

public void Quit()

{

Application.Quit();

}

}

## *PlayerLife*

Unutar PlayerLife skripte je implementirana čitava funkcionalnost “uništenja” heroja ukoliko naiđe na prepreku, uključujući i njegov respawn.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class PlayerLife : MonoBehaviour

{

private Rigidbody2D rb;

private Animator anim;

[SerializeField] private AudioSource deathSoundEffect;

private void Start()

{

rb = GetComponent<Rigidbody2D>();

anim = GetComponent<Animator>();

}

private void OnCollisionEnter2D(Collision2D collision)

{

if (collision.gameObject.CompareTag("Trap"))

{

Die();

}

}

private void Die()

{

deathSoundEffect.Play();

rb.bodyType = RigidbodyType2D.Static;

anim.SetTrigger("death");

}

private void RestartLevel()

{

SceneManager.LoadScene(SceneManager.GetActiveScene().name);

}

}

## *ItemCollector*

Ova skripta omogućava funkciju skupljanja poena.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class ItemCollector : MonoBehaviour

{

private int strawberries = 0;

[SerializeField] private Text strawberriesText;

[SerializeField] private AudioSource collectionSoundEffect;

private void OnTriggerEnter2D(Collider2D collision)

{

if (collision.gameObject.CompareTag("Strawberry"))

{

collectionSoundEffect.Play();

Destroy(collision.gameObject);

strawberries++;

strawberriesText.text = "Strawberries: " + strawberries;

}

}

}

## *PlayerMovement*

PlayerMovement skripta sadrži metode za funkcionalnost kretanja heroja.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PlayerMovement : MonoBehaviour

{

private Rigidbody2D rb;

private BoxCollider2D coll;

private SpriteRenderer sprite;

private Animator anim;

[SerializeField] private LayerMask jumpableGround;

private float dirX = 0f;

[SerializeField] private float moveSpeed = 7f;

[SerializeField] private float jumpForce = 14f;

private enum MovementState { idle, running, jumping, falling }

[SerializeField] private AudioSource jumpSoundEffect;

private void Start()

{

rb = GetComponent<Rigidbody2D>();

coll = GetComponent<BoxCollider2D>();

sprite = GetComponent<SpriteRenderer>();

anim = GetComponent<Animator>();

}

private void Update()

{

dirX = Input.GetAxisRaw("Horizontal");

rb.velocity = new Vector2(dirX \* moveSpeed, rb.velocity.y);

if (Input.GetButtonDown("Jump") && IsGrounded())

{

jumpSoundEffect.Play();

rb.velocity = new Vector2(rb.velocity.x, jumpForce);

}

UpdateAnimationState();

}

private void UpdateAnimationState()

{

MovementState state;

if (dirX > 0f)

{

state = MovementState.running;

sprite.flipX = false;

}

else if (dirX < 0f)

{

state = MovementState.running;

sprite.flipX = true;

}

else

{

state = MovementState.idle;

}

if (rb.velocity.y > .1f)

{

state = MovementState.jumping;

}

else if (rb.velocity.y < -.1f)

{

state = MovementState.falling;

}

anim.SetInteger("state", (int)state);

}

private bool IsGrounded()

{

return Physics2D.BoxCast(coll.bounds.center, coll.bounds.size, 0f, Vector2.down, .1f, jumpableGround);

}

}

## *PointFollower*

Ova skripta omogućava pomeranje platforme levo-desno.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PointFollower : MonoBehaviour

{

[SerializeField] private GameObject[] waypoints;

private int currentWaypointIndex = 0;

[SerializeField] private float speed = 2f;

private void Update()

{

if (Vector2.Distance(waypoints[currentWaypointIndex].transform.position, transform.position) < .1f)

{

currentWaypointIndex++;

if (currentWaypointIndex >= waypoints.Length)

{

currentWaypointIndex = 0;

}

}

transform.position = Vector2.MoveTowards(transform.position, waypoints[currentWaypointIndex].transform.position, Time.deltaTime \* speed);

}

}

## *StickyPlatform*

StickyPlatform je skripta koja služi za kretanje igrača zajedno sa pomerajućom platformom kada skoči na nju i vraćanje na normalno kretanje kada skoči sa nje.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class StickyPlatform : MonoBehaviour

{

private void OnTriggerEnter2D(Collider2D collision)

{

if (collision.gameObject.name == "Player")

{

collision.gameObject.transform.SetParent(transform);

}

}

private void OnTriggerExit2D(Collider2D collision)

{

if (collision.gameObject.name == "Player")

{

collision.gameObject.transform.SetParent(null);

}

}

}

## *Rotate*

Rotate skripta omogućava rotiranje testere (prepreka).

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Rotate : MonoBehaviour

{

[SerializeField] private float speed = 2f;

private void Update()

{

transform.Rotate(0, 0, 360 \* speed \* Time.deltaTime);

}

}

## *Finish*

Finish skripta omogućava prelazak na sledeći nivo kada igrač dotakne finish zastavicu.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

using UnityEngine.UI;

public class Finish : MonoBehaviour

{

private AudioSource finishSound;

private bool levelCompleted = false;

private void Start()

{

finishSound = GetComponent<AudioSource>();

}

private void OnTriggerEnter2D(Collider2D collision)

{

if (collision.gameObject.name == "Player" && !levelCompleted)

{

finishSound.Play();

levelCompleted = true;

Invoke("CompleteLevel", 2f);

}

}

private void CompleteLevel()

{

SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);

}

}

## *CameraController*

CameraController skripta služi za fiksiranje pozicije kamere na heroja.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class CameraController : MonoBehaviour

{

[SerializeField] private Transform player;

private void Update()

{

transform.position = new Vector3(player.position.x, player.position.y, transform.position.z);

}

}

## *DisplayTime*

DisplayTime skripta služi za prikaz tajmera koji se aktivira kada se nivo 1 pokrene.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class DisplayTime : MonoBehaviour

{

[SerializeField] private Text TimeText;

private void Update()

{

float sessionTime = Time.realtimeSinceStartup - PlayerPrefs.GetFloat("SessionTime");

int MM = Mathf.FloorToInt(sessionTime / 60.0f);

int SS = Mathf.FloorToInt(sessionTime % 60f);

TimeText.text = "Time: " + MM.ToString("00") + ":" + SS.ToString("00");

}

}

# Zaključak

U ovom radu su predstavljeni svi neophodni elementi za razvoj igre Pixel Runner, počev od audio elemanata, fontova do skripti u C# programskom jeziku. Dat je opis interfejsa i razvojnog okruženja Unity. Svaka igra pruža mogućnost dodavanja inovativnog sadržaja i prema tome logični nastavak razvoja igre Pixel Runner bio bi proširenje funkcionalnosti, rad na novim elementima igre, kreiranje novih nivoa, unapređenje grafike i vizuelnih efekata, zatim prebacivanje na Web platformu, igranje iz pretraživača, itd.

# Literatura

[1] Aleksić V. Razvoj digitalnih igara, FTN Čačak

# Reference

[1] <https://assetstore.unity.com/packages/2d/characters/pixel-adventure-1-155360#description>, heroj, zamke, platforme, reljef, muzika

# Spisak slika

[Slika 1. Audio 4](#_Toc102567769)

[Slika 2. Prikaz PressStart 2P retro fonta 5](#_Toc102567770)

[Slika 3. Prefabs elementi 5](#_Toc102567771)

[Slika 4. Scene Pixel Runner-a 5](#_Toc102567772)

[Slika 5. Početni meni 6](#_Toc102567773)

[Slika 6. Meni sa opcijama 6](#_Toc102567774)

[Slika 7. Početak nivoa 1 7](#_Toc102567775)

[Slika 8. Prepreka sa šiljcima nivoa 1 7](#_Toc102567776)

[Slika 9. Pomerajuća platforma nivoa 1 7](#_Toc102567777)

[Slika 10. Pomerajuća rotirajuća testera nivoa 1 8](#_Toc102567778)

[Slika 11. Vertikalno pomerajuće testere nivoa 1 8](#_Toc102567779)

[Slika 12. Kraj nivoa 1 8](#_Toc102567780)

[Slika 13. Početak nivao 2 9](#_Toc102567781)

[Slika 14. Pomerajuća rotirajuća testera nivoa 2 9](#_Toc102567782)

[Slika 15. Pomerajuća platforma nivoa 2 9](#_Toc102567783)

[Slika 16. Pomerajuće rotirajuće testere u suprotnim smerovima nivoa 2 10](#_Toc102567784)

[Slika 17. Vertikalno pomerajuće testere nivoa 2 10](#_Toc102567785)

[Slika 18. Kraj nivoa 2 10](#_Toc102567786)

[Slika 19. Završni meni 11](#_Toc102567787)

[Slika 20. Scipts 11](#_Toc102567788)