**HIGH SCHOOL STUDENT INSTITUTE OF MATHEMATICS AND INFORMATICS**

**SUMMER RESEARCH SCHOOL’17**

***Mobile application***

***“Blumber”***

**Author:**

Antoan Georgiev

Maths and Science High School – Montana

E-mail: [toto010@abv.bg](mailto:toto010@abv.bg)

**Mentor:**

Assoc. Prof. Zlatogor Minchev

Institute of ICT/ Institute of Mathematics & Informatics

Bulgarian Academy of Sciences

E-mail: [zlatogor@abv.bg](mailto:zlatogor@abv.bg)

**Science supervisor:**

Krasimir Asenov

Maths and Science High School – Montana

E-mail: [asenow@gmail.com](mailto:asenow@gmail.com)

# TABLE OF CONTENTS

Contents

[TABLE OF CONTENTS 2](#_Toc490524025)

[SHORT SUMMARY 3](#_Toc490524026)

[MAIN PURPOSES OF THE PROJECT 4](#_Toc490524027)

[WHAT ARE THE ADVANTAGES OF THE PROJECT? 5](#_Toc490524028)

[SOURCES 5](#_Toc490524029)

[HOW DID I START? 6](#_Toc490524030)

[DEVELOPMENT IN UNITY 9](#_Toc490524031)

[INTERFACE 10](#_Toc490524032)

[CONCLUSION 12](#_Toc490524033)

[FUTURE PLANS ABOUT THE PROJECT 12](#_Toc490524034)

[WHAT SOURCES DID I USE? 13](#_Toc490524035)

# SHORT SUMMARY

"Blumber" is a new mobile game, created with modern game design technological interactive tools of Unity. The game main idea is the following:

The player has to slice a flying banana, which spawns in a short interval. If the players fails to do that, he loses a life. The end of the game occurs when the player loses 3 lives (and unfortunately loses the game). Point or time limit is not set up.

The game is organized as SP (single-player), but leaderboard and achievement buttons are planned to be available soon, so that the players could compete each other.

You do not need to have internet connection in order to play the game.

The app doesn’t have specific terms of use, it is even open-source.

The age restriction is PEGI 3, which means that (almost) everyone can play Blumber.

And one of the most important things – it is free!

# 

# 

# 

# **MAIN PURPOSES OF THE PROJECT**

**The project should be (and I hope it is):**

* Entertaining
* Independent from Internet
* Available for everyone
* With easy and intuitive interface
* Interesting
* Innovative

# WHAT ARE THE ADVANTAGES OF THE PROJECT?

One of Blumber's goals is to be massively played. The key to this is the independence from the Internet, which ensures trouble-free play by users around the world.

And another major advantage - the totally free content and the lack of ads. Nobody loves annoying advertisements appearing across the screen.

# **SOURCES**

The application was built with Unity, validated to the latest standards.

For the code part of the game, C# was used. The Artwork was created by me, so Photoshop was used.

# HOW DID I START?

Before I started doing anything, I had to clarify some questions. And, to be honest, they were so many. Starting with the mathematics of the game, how to create the movement and even what artwork to use.

My main scripts are 2 – Food.cs (which probably sounds better as Banana.cs) and Manager.cs.

We will first take look at the Food.cs, starting with the declaration of the dimensions.

public bool Active{set;get;}

public SpriteRenderer sRenderer;

private const float gravity = 2.0f;

private float speed;

private float verticalVelocity;

private bool isSliced = false;

public Sprite[] sprites;

private int spriteIndex;

private float lastSpriteUpdate;

private float spriteUpdateDelta = 0.125f;

There is not anything confusing here. We have created some values which are being used later.

Another interesting moment in my code is the FoodLauncher void.

public void FoodLauncher(float verticalVelocity, float xSpeed, float xStart)

{

Active = true;

speed = xSpeed;

this.verticalVelocity = verticalVelocity;

transform.position = new Vector3 (xStart, 0, 0);

isSliced = false;

spriteIndex = 0;

sRenderer.sprite = sprites [spriteIndex];

}

A bit more complicated is my Update void.

private void Update()

{

if (!Active)

return;

verticalVelocity -= gravity \* Time.deltaTime;

transform.position += new Vector3 (speed, verticalVelocity, 0) \* Time.deltaTime;

if (isSliced) {

if (spriteIndex != sprites.Length-1 &&Time.time - lastSpriteUpdate > spriteUpdateDelta) {

lastSpriteUpdate = Time.time;

spriteIndex++;

sRenderer.sprite = sprites [spriteIndex];

}

}

if (transform.position.y < -1) {

Active = false;

if (!isSliced)

Manager.Instance.LoseLP ();

}

}

And the last thing in this code is my Slice void. When the banana is sliced, it bounces a bit, isActive becomes true, which means that another sprite is loaded. And, of course, we increment the score.

public void Slice()

{

if (isSliced)

return;

if (verticalVelocity < 0.5f)

verticalVelocity = 0.5f;

speed=speed\*0.5f;

isSliced = true;

Manager.Instance.IncrementScore (1);

}

# 

Now, we are going to take a look at my Manager script. Starting with the declaration. There are some more values here but they are pretty easy to understand.

private List<Food> food = new List<Food>();

public GameObject foodPrefab;

public GameObject pauseMenu;

private Collider2D[] foodCols;

private float deltaSpawn=1.0f;

private float lastSpawn;

private const float Sliceforce = 300.0f;

private int scoreCount;

private int highScore;

private Vector3 LastMousePosition;

public Transform trail;

private int lifePoint;

public Text scoreText;

public Text highscoreText;

public Image[] points;

public static Manager Instance;

private bool isPaused = false;

public GameObject deathMenu;

The Update void here is a bit more complicated as well.

private void Update()

{

if (isPaused)

return;

if (Time.time - lastSpawn > deltaSpawn) {

Food f = getFood ();

float randomX = Random.Range (-1.65f, 1.65f);

f.FoodLauncher (Random.Range (1.85f, 2.75f), randomX, -randomX);

lastSpawn = Time.time;

}

if (Input.GetMouseButton (0)) {

Vector3 pos = Camera.main.ScreenToWorldPoint (Input.mousePosition);

pos.z = -1;

trail.position = pos;

Collider2D[] thisFramesFood = Physics2D.OverlapPointAll (new Vector2 (pos.x, pos.y), LayerMask.GetMask ("Banana"));

if ((Input.mousePosition - LastMousePosition).sqrMagnitude > Sliceforce) {

foreach (Collider2D c2 in thisFramesFood) {

for (int i = 0; i < foodCols.Length; i++) {

if (c2 == foodCols [i]) {

c2.GetComponent<Food> ().Slice ();

}

}

}

}

LastMousePosition = Input.mousePosition;

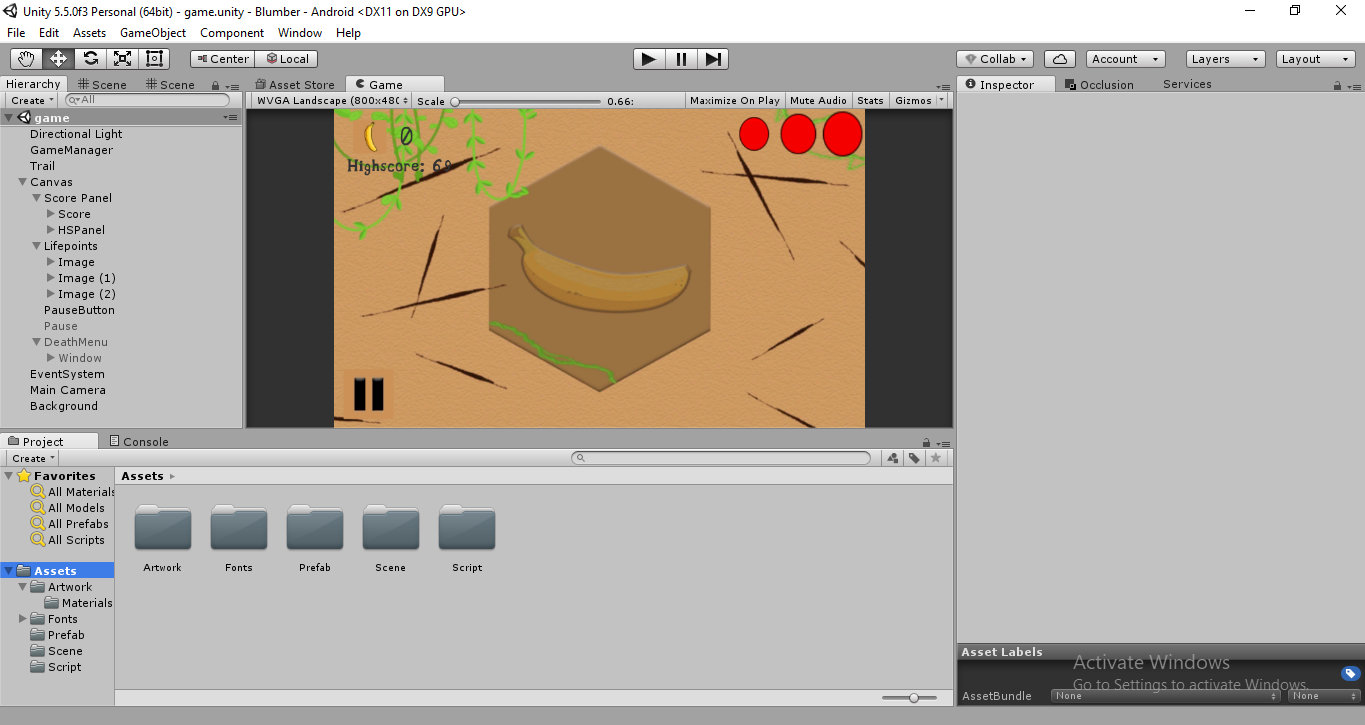
foodCols = thisFramesFood;

}

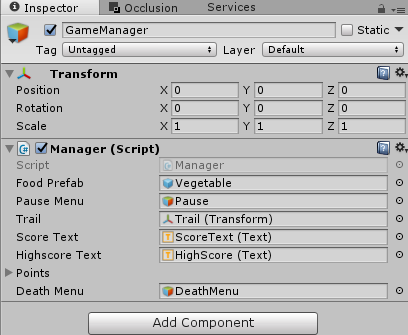
}

# DEVELOPMENT IN UNITY

So, that is how the game looks like in Unity.



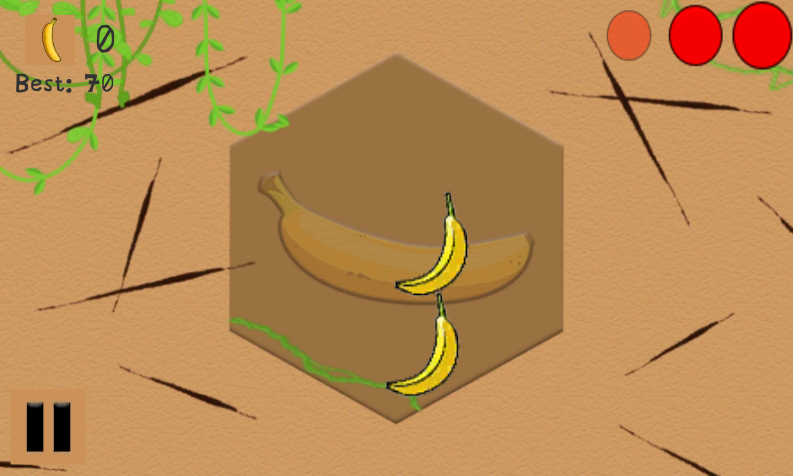
The GameObject called GameManager is really interesting since my FoodPrefab, Pause Menu, Trail, Score and Highscore text, plus the lifepoints and the DeathMenu are attached to it.



# INTERFACE







# CONCLUSION

Blumber is a mobile game in which you have to slice a flying banana and become a “banana-master”. Created with modern technology, the application has nice and innovative design, and intuitive interface for touch screen Android devices.

The game is independent from Internet and does not have age restriction (you just have to be more than 3 years old). This means that the game can be played by almost everyone.

# FUTURE PLANS ABOUT THE PROJECT

* Improved interface
* Reducing the size of the game
* Leaderboard
* Achievements
* Shop
* Adding more fruits
* Adding bombs!
* Multicriteria assessment of the game
* Improving the complexity of game menu
* Creating a version for 3D VR with intuitive control

# WHAT SOURCES DID I USE?

Because of the specificity of the project, I used different sources.

I consulted with:

* My mentor
* My science supervisor
* My counselor
* Family, friends
* Unity Help & Examples documentation
* Briar Lee Mitchell – Game Design Essentials