

KAUNAS UNIVERSITY OF TECHNOLOGY

FACULTY OF INFORMATICS

T120B166 Development of Computer Games and Interactive Applications

*Drowning Bullets*

|  |
| --- |
| *IFZm-1, Emilija Kravčenko:* |
|  |
| Date: 2024*.04.09* |

Kaunas, 2024

Tables of Contents

[Tables of Images 3](#_Toc167767391)

[Table of Tables/functions 4](#_Toc167767392)

[Description of Your Game 5](#_Toc167767393)

[Laboratory work #1 6](#_Toc167767394)

[List of tasks 6](#_Toc167767395)

[Solution 6](#_Toc167767396)

[Task #1. *Camera Movement* 6](#_Toc167767397)

[Task #2. *Guns* 7](#_Toc167767398)

[Task #3. *Ammo control* 9](#_Toc167767399)

[Task #4. Monsters walking, attacking and dying 10](#_Toc167767400)

[Task #5. Monsters dropping items after their death 12](#_Toc167767401)

[Task #6. Picking up items 13](#_Toc167767402)

[Task #7. Pause Menu 17](#_Toc167767403)

[Task #8. Weapon Upgrade 18](#_Toc167767404)

[Task #9. Monsters shooting at player 19](#_Toc167767405)

[Task #10. Pickaxe and crystals 20](#_Toc167767406)

[Task #11. Water and swimming 20](#_Toc167767407)

[Task #12. Adding grass, rocks 20](#_Toc167767408)

[Task #13. Music and sound effects 21](#_Toc167767409)

[Defense 21](#_Toc167767410)

[User's manual 24](#_Toc167767411)

[Literature list 25](#_Toc167767412)

[ANNEX 25](#_Toc167767413)

# **Tables of Images**

[Figure 1. Camera Movement Screenshot 6](#_Toc163956644)

[Figure 2. Gun and animation Screenshot 7](#_Toc163956645)

[Figure 3. Rifle gun’s ammo Screenshot 9](#_Toc163956646)

[Figure 4. Monster‘s punch animation 10](#_Toc163956647)

[Figure 5. Monster‘s death animation 10](#_Toc163956648)

[Figure 6. Spawned pickable items 12](#_Toc163956649)

[Figure 7. Health drink 12](#_Toc163956650)

[Figure 8. Weapon upgrade 12](#_Toc163956651)

[Figure 9. Ammo Box 13](#_Toc163956652)

[Figure 10. Player‘s scan area for pickable items 14](#_Toc163956653)

[Figure 11. UI label for picked up health drinks and weapon upgrades 14](#_Toc163956654)

[Figure 12. UI label for picked up health drinks and weapon upgrades (2) 15](#_Toc163956655)

[Figure 13. Paused game screen 16](#_Toc163956656)

[Figure 14. Arsenal scene 17](#_Toc163956657)

[Figure 15 Arsenal scene (2) 18](#_Toc163956658)

[Figure 16 Player‘s damage on collision screenshot 19](#_Toc163956659)

[Figure 17. Zoomed in sgame screen 20](#_Toc163956660)

# **Table of Tables/functions**

[Table 1. Camera Movement Code 7](#_Toc163956661)

[Table 2. Weapon Resource code 8](#_Toc163956662)

[Table 3. Weapon Collision detection code 8](#_Toc163956663)

[Table 4. Weapon Reload Code 9](#_Toc163956664)

[Table 5. Monster‘s movement code 11](#_Toc163956665)

[Table 6. Monster‘s distance to player calculation 11](#_Toc163956666)

[Table 7. Item spawn code 13](#_Toc163956667)

[Table 8. Player‘s pickable item scan code 15](#_Toc163956668)

[Table 9. health drink‘s code 16](#_Toc163956669)

[Table 10. Pause Menu code 17](#_Toc163956670)

[Table 11. Arsenal‘s weapon‘s data upgrade code 18](#_Toc163956671)

[Table 12 Player‘s damage on collision code 19](#_Toc163956672)

[Table 13. weapon zoom code 20](#_Toc163956673)

[Table 14. Crosshair texture change code 20](#_Toc163956674)

# **Description of Your Game**

Description of Your Game.

1. 3D or 2D? 3D
2. What type is your game? First Person Looter Shooter
3. What genre is your game?Action Roguelike
4. Platforms (mobile, PC or both?) PC
5. Scenario Description. Pirmo asmens 3D roguelike šaudyklė. Žaidėjas pradeda žaidimą vandenyje. Link jo bėga priešai (monstrai). Kai veikėjas šauna šiek tiek paneria gilyn. Ilgai panėrus paskęsta. Jei monstrai sužeidžia taip pat miršta. Turas pasibaigia kai žaidėjas nužudo visus priešus arba miršta. Tada žaidėjas gali pasirinkti ar kovoti toliau ar baigti žaidimą ir išlipanti iš vandens. Kiekvieną turą priešų vis daugėja ir jie stiprėja (turi daugiau gyvybių, suteikia daugiau žalos). Priešus nužudžius, jie palieka po savęs naujus ginklus, šaudmenis, gyvybių atstatymą arba, rečiau, pagerinimus žaidėjo veikėjui arba žaidėjo ginklams.  Nereikalingus ginklus galima išardyti į pagerinimus ginklams.Pagerinimų kiekis yra neribotas ginklams ir veikėjui, jų galima prisidėti kiek norisi. Kiekvienas žaidėjo veikėjas turi vieną pradinį pasyviai veikiantį gebėjimą. Atrakinus naują veikėją, veikėjo gebėjimas taip pat atsirakina ir jį galimą pridėti prie senų veikėjų gebėjimų. Perėjus penkis turus atsirakina gebėjimas pradėti žaidimą iškarto po penkių turų. Tai galioja kas 5 turus (10, 15, 20…). 101 turas yra paskutinis ir jame žaidėjas kovoja prieš vieną didelį monstrą. Nugalėjus jį žaidėjas laimi. Žaidimo metu žaidėjas taip pat gali atsirakinti naujas kovos vietas (pelkę, suledėjusį vandenį ir pan.). Kiekvienai kovos vietai saugomas atskiras pereitų turų sekimas.

**Laboratory work #1**

**List of tasks** (main functionality of your project)

1. Camera Movement
2. Guns
3. Ammo Control
4. Monsters walking, attacking, dying
5. Monsters dropping items after their death
6. Picking up items
7. Pause Menu
8. Weapon Upgrade
9. Monsters shooting at player
10. Pickaxe and crystals
11. Water and swimming
12. Adding grass, rocks
13. Music and sound effects

# **Solution**

## **Task #1. *Camera Movement***

Adding movement to character’s camera when player moves mouse. Moving the mouse changes what the player sees and the direction their character is looking at. Rotation to the side is unlimited, but looking up and down has fixed values.

A screenshot of a video game

Description automatically generated

Figure 1. Camera Movement Screenshot

After mouse motion camera and character rotates towards that direction based on defined mouse sensitivity. Pressing Escape key cancels mouse tracking and displays mouse cursor.

A screen shot of a computer program

Description automatically generated

Table 1. Camera Movement Code

## **Task #2. *Guns***

Adding gun models. Creating animations for shooting reloading, changing weapon. Creating weapon resources. Adding ray cast to check if shot hit target.

A screenshot of a computer program

Description automatically generated

Figure 2. Gun and animation Screenshot

Weapon Resource is used to define weapon class, weapon variables to use on all guns.

A screen shot of a computer program

Description automatically generated

Table 2. Weapon Resource code

HitScanCollision calculates ray’s direction, end point, any collision that intersects with ray. On collision it displays Hit indicator – Bullet hole.

HitScanDamage checks if Ray cast from a gun points at an object in group “Target”. And sends variables of weapon damage, bullet’s direction and position where it hit target.

A screen shot of a computer program

Description automatically generated

Table 3. Weapon Collision detection code

## **Task #3. *Ammo control***

Adding current ammo, magazine, reserve ammo and max ammo variables. These variables help to track and display each gun’s Ammo state and capacity. After reaching 0 ammo, weapon automatically reloads after “shooting” - pressing left mouse button. Player can also press “R” to reload at any time.

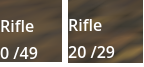


Figure 3. Rifle gun’s ammo Screenshot

When starting gun’s current ammo is gun’s full magazine ammo. On reloading it checks if the current ammo is full. If not, it calculates how much to take from ammo reserve and recalculates gun’s current and reserve ammo variables. At the end it sends signal to update gun’s variables.

A screen shot of a computer program

Description automatically generated

Table 4. Weapon Reload Code

## **Task #4. Monsters walking, attacking and dying**

Monsters walk/fly towards the player on predefined navigation mesh. After reaching a certain distance from the player they can punch the player, thus hurting them. If the player manages to kill them, they fall to the ground and disappear after a few seconds.

Monsters has three animations: walking, punching, and dying. Monsters interpolate between them based on their health and distance from their target – the player.

A screenshot of a video game

Description automatically generated

Figure 4. Monster‘s punch animation

A video game screen with cartoon characters

Description automatically generated

Figure 5. Monster‘s death animation

A screen shot of a computer screen

Description automatically generated

Table 5. Monster‘s movement code

A screen shot of a computer program

Description automatically generated

Table 6. Monster‘s distance to player calculation

## **Task #5. Monsters dropping items after their death**

## 

When the player kills any monster, after monster disappears three items are spawned on its death location. The dropped items are: health drink, random upgrade for a weapon and 10 bullets in a box.

Monster’s code initializes all spawnable items scenes and adds them to the scene before erasing killed monster’s scene from current scene.

A video game screen with a pixelated object

Description automatically generated

Figure 6. Spawned pickable items

A red can with black lid

Description automatically generated

Figure 7. Health drink

A green cube with a hole in it

Description automatically generated

Figure 8. Weapon upgrade

A purple and grey rectangular object

Description automatically generated

Figure 9. Ammo Box

A screen shot of a computer program

Description automatically generated

Table 7. Item spawn code

## **Task #6. Picking up items**

Player has sphere shaped area around them that scans for pickable objects that enter that area. This area does not interact with anything else and only calls a found item’s method, upon finding it. All pickable objects are placed in one layer and each one has a method “Collectable” that implements item’s functions and then removes it from a scene.

Item signals to player that it is collected and after receiving a signal player’s interface is updated, picked up item can be instantly used.

A grid with a circular pattern

Description automatically generated with medium confidence

Figure 10. Player‘s scan area for pickable items

A brown square with a black and white checkered pattern

Description automatically generated

Figure 11. UI label for picked up health drinks and weapon upgrades

A brown surface with white dots

Description automatically generated with medium confidence

Figure 12. UI label for picked up health drinks and weapon upgrades (2)

A screen shot of a computer code

Description automatically generated

Table 8. Player‘s pickable item scan code

A screen shot of a computer

Description automatically generated

Table 9. health drink‘s code

## **Task #7. Pause Menu**

Pressing Esc key pauses the game and shows a pause menu. By clicking “Pasiduoti” button player can leave the game and go to the Main menu screen. Clicking Esc key again, removes pause screen and resumes the game.

Pausing the game causes all scripts connected to the game to wait for the game to be unpaused. Pause menu and its elements process actions and input only when the game process is paused.

A video game screen with text and a blue sky

Description automatically generated

Figure 13. Paused game screen

A screen shot of a computer program

Description automatically generated

Table 10. Pause Menu code

## **Task #8. Weapon Upgrade**

If the player picks up weapon upgrades during the fight, all weapon upgrades appear in arsenal. Arsenal has two shelves. On the left one upgrades are displayed, on the right one, weapons and their data (damage, magazine size, ammo). After picking an upgrade and picking a weapon to upgrade, a player can press on the upgrade button in the middle to upgrade chosen weapon with a chosen upgrade.

All weapon data is stored in a weapon’s resource file. Picked up weapon upgrades are stored in a globally accessible file. After each upgrade files are updated and new values are used in fight scene.

A screenshot of a computer

Description automatically generated

Figure 14. Arsenal scene

A screenshot of a computer

Description automatically generated

Figure 15 Arsenal scene (2)

A screen shot of a computer program

Description automatically generated

Table 11. Arsenal‘s weapon‘s data upgrade code

## **Task #9. Monsters shooting at player**

A green monster can shoot beams at player from a bigger distance than melee monsters punch. Green monsters walk slower and take longer to recharge their attacks. Monster only finishes beam attack if it can hurt player.

Beam is created using raycast, particle and shader. Particle is used to make beam as long as the distance between monster and the player. Shader is used to create an effect of rotation and movement of the beam.

## **Task #10. Pickaxe and crystals**

By pressing “c” player can take out a pickaxe. Pickaxe mining is done with left mouse button. Pickaxe can only mine crystals, that are underwater Crystals are spawned in four predefined locations as one of three colors (yellow, green, red) and respawn again after each wave if they were mined. Each crystal can be mined three times and give three small crystals, that then can be used in armory to upgrade character stats (Hp and Time that player can be underwater before drowning).

Pickaxe hit is being animated and at the point it reaches the ground and crystals check if their area is colliding with pickaxe. After each successful hit with pickaxe on crystal it scales itself down 1/3 of its size before disappearing.

## **Task #11. Water and swimming**

Player is always floating in the water or swimming underwater. After pressing “Shift” key if player is in the water, player is submerged underwater. Then player can move like when floating in the water, but also surge deeper down with “Shift” or go towards the surface with “Space”. Player can only stay underwater for some time before drowning begins. After going underwater a timer appears in the top left corner of the screen. After reaching zero player starts to drown until dying or reaching the surface for air.

After going underwater, water’s collider for player is removed and player can shoot upwards. But once player’s raycast reaches water collider, it is no longer considered underwater and can now float. Pressing the same button (“Shift” or “Space”) increases the speed until reaching maximum possible speed. Pressing them once, one after the other, makes player float in one place.

Water is a simple plain mesh with a shader, that simulates three waves and creates foam around objects it touches. Material shader is taken from github and changed a little.

After going underwater environment becomes darker, fog appears ant caustic effect is displayed on the terrain underwater. Caustics effect is created from a texture and is moved over terrain material, but bellow water level.

## **Task #12. Adding grass, rocks**

A Godot plugin was used to create big area of grass and rock walls around lake. Grass and rock model used from plugin. In scatter node, after defining what amount of grass to generate, and its properties, the most optimal amount was 10 patches of 1000 grass models. A loading screen was created to wait for grass to be visualized, before starting the game. For rocks, only 4 patches of 20 large rocks were used, so it does not require intense processing.

## **Task #13. Music and sound effects**

Two different music’s are played across the game. One, when the player is in main menu or Arsenal, another, when the player is fighting. In main menu player can also check off sound button to mute all sounds or use slide bar to adjust sound level.

To play music continuously across scenes a single autoload scene was created with two sound players. One for music and one for sound effects. To music player, a low pass filter is applied when player is underwater to simulate effect. A sound effect is played whenever something happens. Player shoots, picks up items, is attacked, dies etc.

**Defense**

**Task: On player’s character’s collision with Monsters take damage.**

Solution:

When moving checks is collided object has method Hurt. If it does, the player takes one damage. And waits 1 second before checking if colliding with anything.

A screen shot of a computer screen

Description automatically generated

Table 12 Player‘s damage on collision code

A screenshot of a video game

Description automatically generated

Figure 16 Player‘s damage on collision screenshot

**Task: When zooming position weapon so it is visible to player and make bigger crosshair**

Solution:

Creating animations for weapons and playing them when weapons are in zoom mode, chainging crosshair texture when pressing zoom button.

A video game screen with an object and a cartoon character

Description automatically generated

Figure 17. Zoomed in sgame screen

A computer screen shot of text

Description automatically generated

Table 13. weapon zoom code

A screen shot of a computer program

Description automatically generated

Table 14. Crosshair texture change code

**User's manual**

**How to play?** Shoot monsters and gather resources dropped from dead monsters. Use gathered resources to upgrade weapons, replenish health and ammo during fight. Avoid dying by taking too much damage or drowning.

**Descriptions of the rules of the game**. Kill all monsters and advance to next wave with more and stronger enemies or come back home to improve stats keeping the loot. Killing Boss wins the game. But still lets you replay the waves as much as you want.

**Descriptions of the controls / keys.** Arrow keys moves character, mouse moves character’s camera. Spacebar makes character jump. Left mouse button shoots bullets from a gun. Mouse scroll wheel changes weapons. Right mouse button zooms in/out. R key reloads weapon. T uses picked up health drinks and replenishes health. In arsenal scene, clicking on the upgrade, clicking on the weapon and on the button between them upgrades that weapon by chosen upgrade. Esc pauses and resumes paused game.

# **Literature list**

1. Source #1. <https://www.youtube.com/watch?v=ZaEzjnoIy3M>
2. Source #2. <https://docs.godotengine.org/en/latest/>
3. Source #3. <https://www.youtube.com/watch?v=iV710Vm5qm0>
4. Source #4. <https://sketchfab.com/3d-models/skull-monster-6352e676c99946908d78f7dacd07ac0d>
5. Music by: Bensound, License code: IYPTKDXXOMWCTOTU
6. Music I Use: Bensound.com/free-music-for-videos, License code: EZMAGPBLZE61TZK4
7. <https://soundbible.com/1328-Pickaxe.html>
8. <https://github.com/gstark667/GodotTerrain/tree/master>
9. <https://godotengine.org/asset-library/asset/1866>
10. <https://www.cgtrader.com/free-3d-models/household/household-tools/pickaxe-4c4ca925-a44c-4359-8d58-ac9e489e500c>

# **ANNEX**

Github link:

https://github.com/EmiKrav/Drowning-Bullets.git