

Assignment 1

To Shoot or Not to Shoot

alex diker

Assignment 1 | COMP3064 Game Development - Przemyslaw Pawluk alex.diker@georgebrown.ca

# Table Of Contents

## Detailed game description

## Controls descrption

## enemy description

## Screenshot description

## Art and multimedia index

## Detailed game description

The game that I made for my assignment one, works the following way; I have outlined scripts and prefabs / sprites and sounds that are all utilized within the gameplay and game. The game is called Too Shoot or Not Too Shoot, and is aimed at children audiences.

**ScrollerScript.cs**

Description:

The following is what was achieved within the script:

First we retrieve background objects, we store a reference of each object. I then ordered those exact items in the order of the scrolling, so I know which item will be first to be deleted or recycled using the terminology used in class. Next we are going to compute the positions between each object part before they start interacting in the game. Next we will continue to check if the object is placed/located before or after the camera bounds in order to be able to specify what we are going to do with the object. Then we recycle the items that are not needed in simple terms as they leave the camera view.

**monsterController.cs**

Description:

The following is what was achieved within the script:

The enemy in this case the beetle is invisible until it reaches camera view, it is placed on a random location based on screen view (screen size) and moves the monster back and forth. Scoring of monsters and information is then passed onto the GUI (within GUI script). – You can refer this to understanding scoring subtitle.

Camera.main.GetComponent<GUI>().currentScore++;

To delete the monster I used the following command.

StartCoroutine(blinkUponCollisionAndDestroy(gameObject));

**enemyGenerator.cs**

The Beetle enemy, randomly generated on the map. Time is set by 2. The first enemy appears right away almost. Calls time between monsters, generated.

**GUI.cs**

Description:

The following is what was achieved within the script:

Using UnityEngine.UI for the text that is displayed in Camera View

Its pulling player information from the playerController regarding lives.

endGameText = GameObject.Find("GameOver").GetComponent<Text>();

is disabled by default in order not to display GameOver text on the screen at all times…only when the player is dead.

The GUI also displays the healthText, scoreText, endgameText – end gametext is hidden until gameover.

**Bullet.cs**

Allows you set the bullet speed as a prefab in the inspector. It also sets the direction of the prefab. It then destroys the object when it leaves camera view.

## How does scoring work

The GUI displays the statistics from the playerController, such as Lives, Score, etc

Player lives are specified in playerController.cs

if (playerLives > 0) {

// lose 1 life

playerLives = playerLives-1;

if (playerLives == 0)

{

gameOver();

}

}

}

Player Score are specified in monsterController.cs

//Increase score as needed

Camera.main.GetComponent<GUI>().currentScore++;

The GUI then displays this information.

## Controls descrption

WASD – For player movement

Space – to shoot.

Implemented the following way:

private void Move()

{

if (Input.GetKey(KeyCode.W))

{

transform.Translate(Vector3.up \* Speed \* Time.deltaTime);

}

if (Input.GetKey(KeyCode.S))

{

transform.Translate(Vector3.down \* Speed \* Time.deltaTime);

}

if (Input.GetKey(KeyCode.D))

{

transform.Translate(Vector3.right \* Speed \* Time.deltaTime);

}

if (Input.GetKey(KeyCode.A))

{

transform.Translate(Vector3.left \* Speed \* Time.deltaTime);

}

if (Input.GetKey(KeyCode.Space))

{

Shoot();

}

Shoot() is basically:

Instantiate(playerBullet, transform.position+new Vector3 (playerBulletXOffset,playerBulletYOffset,0), Quaternion.identity);

Quaternion because we are using the 3D Physics.

I added Time.deltatime to add ingame time for movement so that the player does not fly off the camera view…I had a problem with this and deltaTime fixed my problem.

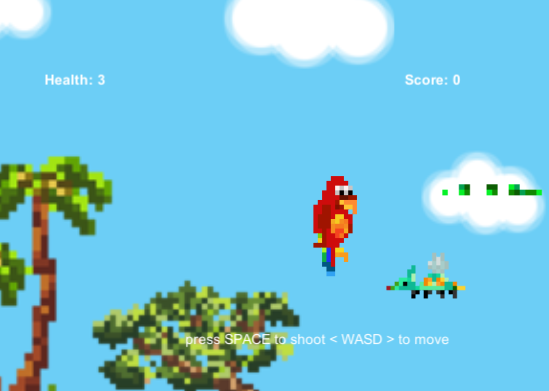
## enemy description



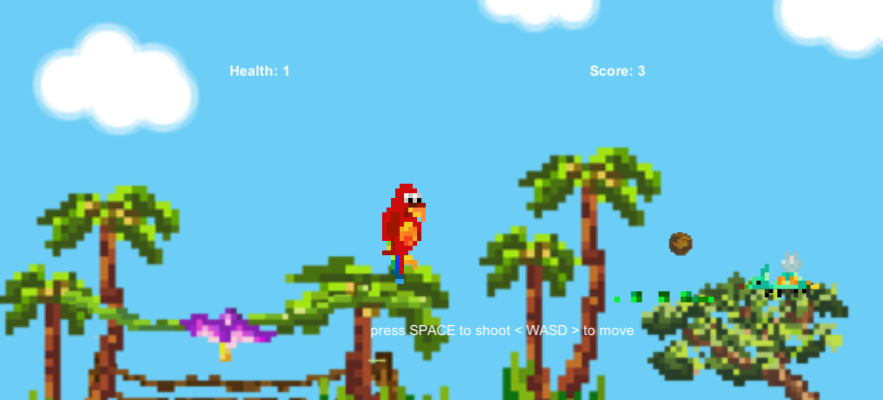
The Beetle enemy, randomly generated on the map

Enemy is invisible until it reaches camera view, it is placed on a random location based on screen view (screen size) and moves the monster back and forth.

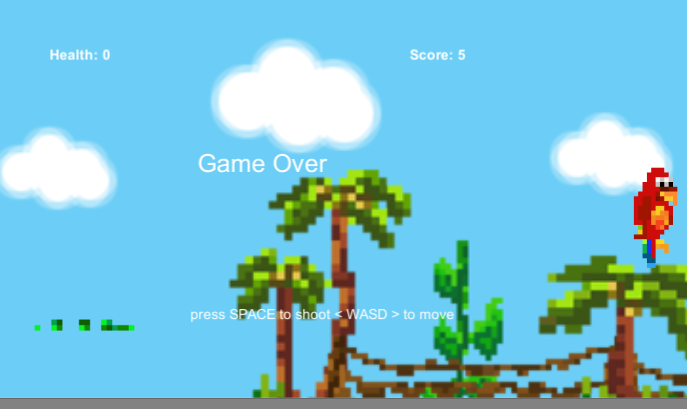
## Screenshot description



The start screen this is the default screen



Default gameplay screen



Game over screen

## Art and multimedia index

Pixel Art and Backgrounds by **Alice Kur** ([alisa.borisovna.k@hotmail.com](mailto:alisa.borisovna.k@hotmail.com))

 Monster Beetle  Player Bullet Enemy BulletMain Character  Trees and Vine

Background art (pixel art taken and placed in photoshop)

**Sounds :**

Bird.wav, bird.mp3, rainforest.mp3

Taken from: <http://www.grsites.com/archive/sounds/category/10/?offset=24>

**Files have been optimized and cut down using Audacity. (open source software)**

**Art has been rendered together as a Background Sprite using Adobe Photoshop CS**

**Pixel Art was created by Alice Kur using PyxelEdit**