City, University of London

BSc Computer Science with

Games Technology

Project Report for Advance Games Technology

December 2019

Hell Run

Cosmin-Ioan Capatina

Table of Contents

1. An overview, including asset listing and additional libraries:

1.1. Project overview:

The title of the project is Hell Run, it is inspired by the famous mobile game “Temple Run” that has more than 100,000,000 downloads.

The idea behind the game is to run as mush as possible without dying and score up as many points as possible to rank up the leaderboard. The genre of the game is Platform game and Endless running, at least that’s what the finished product should be. The game should start with the player running endlessly in a fixed direction/path where different obstacles and enemies will try to stop the player from reaching a high score.

Basketball - https://opengameart.org/content/basket-ball-texture - 12/01/2019 - CC-BY 3.0

Stone - https://www.pinterest.co.uk/pin/431853051754300790/?lp=true – 12/01/2019 – Personal Use License

Skybox- http://www.custommapmakers.org/skyboxes.php - 11/03/2019 - Personal Use License

Rusted Car - https://free3d.com/3d-model/rusted-car-250922.html - 12/01/2019 - Personal Use License

Wall - https://free3d.com/3d-model/old-wall-1357.html - 12/01/2019 - Personal Use License

Car - https://free3d.com/3d-model/-slot-car-blue-sg-v1--58942.html – 12/01/2019 – Personal Use License

Spooky Background Music - https://freesound.org/people/klavo1985/sounds/329789/ - 12/01/2019 – Royalty free Music CC-BY 3.0

Bounce Sound Effect - https://freesound.org/people/andre.nascimento/sounds/51461/ - 12/01/2019 – Royalty free Sound CC-BY 3.0

I did not add any additional libraries to the project.

2. For each of Parts 1,2 and 3 above:

2.1. Part 1:

I tried to add a game lever intro screen but all I could come up with as some simple instructions on the top left of the screen that show how to control the character moving on the screen. I did not manage to make the intro screen although I tried to implement it with Cross Fade.

I have added two primitive-based game objects, the sphere that has the Basketball texture bouncing on the right as the game starts and the brick wall at the far left. Both primitives were achieved by following the tutorials provided by the module and were texture mapped using appropriate texture coordinates. I have changed the skybox to a 'Hellish' red environment to match the game theme and the ground texture has been changed to a grey concrete tiles to give the feeling that the character runs on a road.

I have changed the background music to have a more intense and to give the player the thrill and fear that the character runs from something for his life and should be recognized as soon as the game starts. Also, the bouncing sound has been changed to match the sound of the ball bouncing on concrete.

I have implemented a simple scoring system that is based on how fast the game updates, it increases in increments of ten. I wanted to implement a lives system where the player takes damage from hitting a wall or an enemy, but I am afraid I did not have the time necessary to reach that state of the game. There is no time left implemented in the game because this is no necessary as the game level should be infinite and the game ends when the player runs out of lives and not when the game runs of time.

2.2. Part 2:

The camera technique implemented is behind the player in a 3rd person view, viewed at an angle right above the player. The game should feature a on rails camera technique that follows the game path and not the player but as many other features in this project I did not have the necessary time and knowledge to implement it because of personal problems.

The game in the current state features three mesh-based objects, and they are: Rusted Car, Fence and a race car. All these objects are to be used as obstacles in the game for the character to jump over them or be hit by them so that it stops the player from reaching a high score.

I am afraid I missed the lecture regarding the lighting and although I tried to implement it I did not manage it to get it to a playable state and it would just break my game, so I decided to leave it out since the game has to be at least playable by the end.

Although as seen in the game classes I have tried to use the explosions as a feature for the ball when it touches the ground but as with the problem above I was not competent enough to make it work, even though I wasted countless hours trying to. Same with the fog that would have been perfect for my game atmosphere, as it gives a mysterious vibe and would hide perfectly future obstacles that the player must avoid, making the game harder. Also, I wanted the fog to be a togglable option to make the game harder in the main menu in case the player wanted a harder experience and not be casual.

2.3. Part 3:

When it comes to Non-Player Characters and artificial intelligence I have not implemented anything because lack of time. Although it seems improbable that I did not have the necessary time to implement everything, I am working a full-time job and it is very hard to keep up with everything.

The physics implemented in the game are gravity and resistance. The gravity can be seen in the basketball bouncing off the ground. The resistance can be observed with the collision of the character and the fence. Also, the player moves forward constantly and can turn right, left and jump.

Regarding the gameplay elements. I wanted to implement a combo system that increases the increments of the score multiplier, but I was unable to find a way to make the program read when the player jumps over an obstacle as this is the principle behind it.

3. Final thoughts:

The game as stated in the introduction above build upon an already known game genre. It should be an endless running and a platformer, although the current state of the game doesn't show this, everything to build the game intro a partially finished game is there. With a little more time and dedication, the game should have a playable level with obstacles.

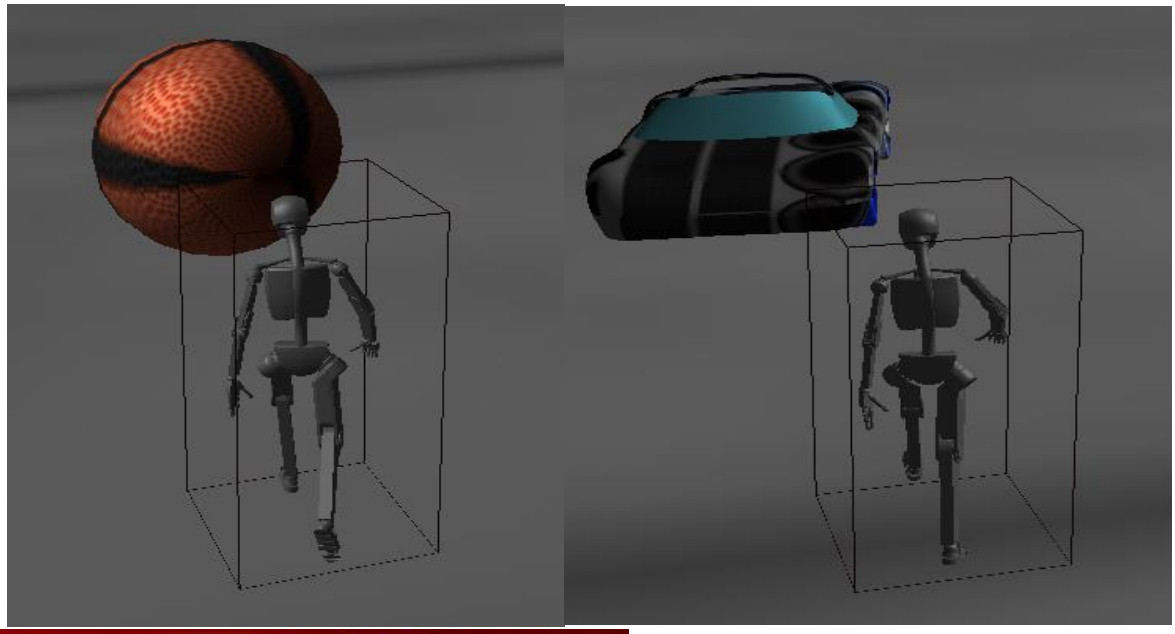
One of the main weakness that the game has is that it does not have the depth for a PC and its target audience is the person that doesn't have a lot of time to play video games and plays between ten to thirty minutes during commute or when possible. Also, the main strength that the game has is replayability, as this is one the main feature. On a final iteration of the game, it should also feature multiple Roguelike elements where everything resets but certain features and upgrades remain permanent, allowing the player to progress further and reaching higher score.

Although it might not seem much in terms of how much I have implemented, I have tried my best with the time I had on my hand and hopefully it will suffice for a pass or more.

Please see below some code snippets and screenshots of the game elements:

```
score = std::to_string(x);
x = y;
y = y + 10;


    const auto text_shader = engine::renderer::shaders_library()->get("text_2D");
    m_text_manager->render_text(text_shader, "Score=", 10.f, (float)engine::application::window().height() - 100.f, 0.5f, glm::vec4(0.f, 1.f, 0.f, 1.f));
    m_text_manager->render_text(text_shader, score, 100.f, (float)engine::application::window().height() - 100.f, 0.5f, glm::vec4(0.f, 0.f, 1.f, 1.f));
```




A - move left
D - move right
Space - jump
Score= 69851


A - move left
D - move right
Space - jump
Score= 63541

Above can be seen all the textures and models used at this stage of the game. When I was finishing this Report, I saw that the car doesn't render properly anymore, and I don't have time to retexture it again in Blender. Also, all the mesh textures have been aligned and textured using Blender, then exported as an Obj extension and used in the game.

I hope this Report explains everything required and I hope I have touched on every question that I had to answer. It had a blast working on this and I am looking forward to the next semester when I will be continuing with the Computer Graphics module.