IN3026 Advanced Games Technology - Milestone 2 Report

Name : Alain Siow

Overview:

The name of the game is currently called Iron Man as it is what I wanted to set out to achieve, a game that allows you to become Ironman. With multiple weapons at your disposal and the ability to fly around a small chunk of an urban city, the player can immerse themselves into being a superhero. The game in its current state is a wave based survival game, the player must utilise their arsenal and flight in order to protect the city from evil robots. The objective is to survive as many waves of increasingly tougher enemies while accumulating as high of a score as possible. The game has a strong mechanic focus as mentioned in Milestone 1, executing the right movement such as strafing before an enemy missile hits or flying away to escape enemies in order to pick up health is paramount to the player's survival. Different weapons allow the player to fire a missile and switch over to repulsor or bouncynade in rapid succession to inflict maximum damage. The player must also be careful of each enemy type as they have different strengths and weaknesses. The end result is an intuitive combat game with satisfying mechanics.

Asset Listing:

Audio :

Ambience :		
https://freesound.org/people/lazymonk/sounds/214319/	(zero)	(4/12/2020)
Explosion 1 :		
https://freesound.org/people/Piotr123/sounds/513973/	(zero)	(4/12/2020)
Explosion 2 :		
https://freesound.org/people/EFlexMusic/sounds/388528/	(attribution nc)	(4/12/2020)
Grenade fire :		
https://freesound.org/people/LeMudCrab/sounds/163458/	(zero)	(4/12/2020)
Glasscrack :		
https://freesound.org/people/jchiledred/sounds/445847/	(attribution)	(4/12/2020)
Heal :		
https://freesound.org/people/Rickplayer/sounds/530488/	(zero)	(4/12/2020)
Missile firing :		
https://freesound.org/people/wubitog/sounds/200459/	(zero)	(4/12/2020)
Shield :		
https://freesound.org/people/nekoninja/sounds/370203/	(zero)	(4/12/2020)
Weapon Switch :		
https://freesound.org/people/knova/sounds/170273/	(attribution nc)	(4/12/2020)

There are 2 sounds that I needed which could not be found elsewhere. It is Ironman's repulsor sound as there is nothing else quite like it. I will be able to use this as long as it falls under the category of fair dealing. And in the unlikely event of me distributing this game, the clips will be removed.

Fair dealing source :
https://www.gov.uk/guidance/exceptions-to-copyright (under fair dealing)
https://www.bl.uk/business-and-ip-centre/articles/fair-use-copyright-explained (under "fair use for the purpose of an examination")

Repulsor Charge :	https://www.youtube.com/watch?v=T2Eq_nH68NA	(5/12/2020)
Repulsor Blast :	https://www.youtube.com/watch?v=g-xIAYVTCCw	(5/12/2020)

All rights belong to their respective owner, Marvel Entertainment.

Most sound clips were cut and edited by me manually to sync duration and volumes with effects in the game.

Meshes:

BB8 :	https://free3d.com/3d-model/bb8-35865.html	(1/12/2020)
Droid :	https://free3d.com/3d-model/star-wars-1b03-93996.html	(1/12/2020)
Drone :	https://free3d.com/3d-model/sci-fi-drone--58689.html	(1/12/2020)
Heart :	https://free3d.com/3d-model/heart-v1--539992.html	(4/12/2020)
Ironman Helmet :	https://free3d.com/3d-model/iron-man-helmet-81900.html	(2/12/2020)
Jet :	https://free3d.com/3d-model/learjet-25-atlasjet-42833.html	(2/11/2020)
Mech :	https://free3d.com/3d-model/mech-robot-f-432-354558.html	(4/12/2020)
Missile :	https://free3d.com/3d-model/aim-120d-missile-air-to-air-20348.html	(1/12/2020)
Pizza :	https://free3d.com/3d-model/dominos-pizza-80809.html	(2/11/2020)
Police car :	https://free3d.com/3d-model/police-car-267529.html	(4/12/2020)
Shield :	https://free3d.com/3d-model/a-shield-with-a-raised-star-v1--27813.html	(1/12/2020)
Skyscrapers :	https://free3d.com/3d-model/skyscraper-91364.html	(6/11/2020)
	https://free3d.com/3d-model/skyscrapers--64845.html	(2/11/2020)
	https://free3d.com/3d-model/cartoon-skyscraper-43638.html	(5/12/2020)
	https://free3d.com/3d-model/evilgits-san-fran-tower-815041.html	(2/11/2020)
	https://free3d.com/3d-model/skyscraper-office-hotel-75255.html	(2/11/2020)

Some meshes were painted by me using blender when no texture was included with the download.

Texture:

Skyscraper :
https://www.filterforge.com/filters/5016-diffuse.html (29/10/2020)

Skybox :
https://www.cleanpng.com/png-skybox-texture-mapping-cube-mapping-desktop-wallpa-6020000/preview.html (2/11/2020)

Manhole :
https://www.sketchuptextureclub.com/textures/architecture/roads/street-elements/manhole-cover-texture-19691 (7/11/2020)

Plastic :
https://www.rgbstock.com/photo/nyVCRSo/Plastic+3 (29/10/2020)

Pavement :
https://www.outhaus.ie/products/silver-grey-granite-flag-paving-tile/ (28/11/2020)

Metal:
https://www.behance.net/gallery/60064327/20-Seamless-Brushed-Metal-Background-Textures-DOWNLOAD (1/12/2020)

Road Texture :
https://www.cgbookcase.com/textures/four-lane-road-clean-01 (28/11/2020)

The road and intersection texture was made using this road texture as a base and painted on by me using photoshop.

Images:

Particle Effect :
https://kenney.nl/assets/particle-pack (28/11/2020)

Screencrack :
https://www.hiclipart.com/free-transparent-background-png-clipart-zkjru (1/12/2020)

Hit Effect:
https://opengameart.org/content/hit-animation-frame-by-frame (1/12/2020)

All HUD elements were hand drawn from scratch using photoshop, drawing inspirations from Ironman’s HUD from the movies.
Start screen and end screen were also created by me using photoshop.

Part 1 - Basic Game Modelling:

• Final game level intro screen :

I’ve made an intro screen detailing the objectives and some tips as well as some of the basic controls of the game.

Full controls :	WASD	- Move	Left Mouse	- Fire Selected Weapon
	Shift	- Sprint	1	- Weapon Slot 1
	Space	- Jetpack	2	- Weapon Slot 2
	C	- Change Cam	3	- Weapon Slot 3
	F	- Toggle Hover	0	- Secret
	Right Mouse	- Rotate Cam (3rd person view)		
For dev purpose :	T	- FreeView Cam (C to switch back)		
	9	- Trigger Crackscreen crossfade		
	N	- Toggle Day/Night		

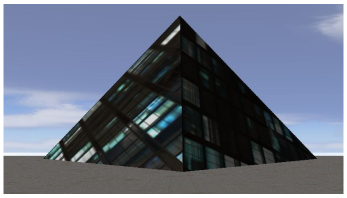
• Primitives :

There are currently 4 primitives in the game.

First one is a bench made entirely from scratch with 83 vertices, it is a whole object and not made out of different smaller primitive shapes. Fully texture mapped and lit with the correct normals. Rotated, translated, scaled and placed throughout the map. {bench.cpp}

Second one is a lamppost also made entirely from scratch with 75 vertices. It is dual textured using texture coordinates from the same image with two different materials (lamppost.png). Lit with correct normals including usage of cross product for the slopes at the bottom of the lamppost. Rotated, translated, scaled. {lamppost.cpp}

Third one is a hexagonal shaped manhole.
Textured and lit as well as scaled and translated in the world



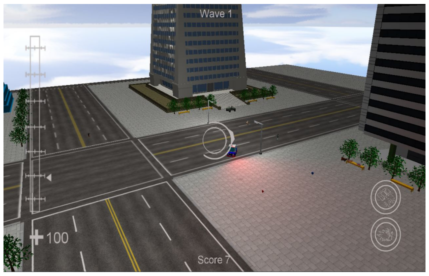
The last one is from the lab. A tetrahedron textured to look like the Louvre Pyramid in paris.

The roads are made from the “terrain” type. Not exactly a primitive but worth mentioning anyway.

• Audio :

The sound clips in the game were carefully chosen to fit the theme of the game. Different weapons have different hit and firing sounds. Picking up a powerup also makes a sound. Lastly, the ambient for the game is the busy sound of New York City. Slight modifications to event_sound.cpp were made to change sound dropoff, audios were cut and carefully synced by hand to deliver a polished experience. An attempt to add jetpack noise was made, experimenting with timer based triggers to looping millisecond clips, in the end it did not sound good and was removed. Initially there were more sounds such as AIs making noise, but it got noisy really quick along with the ambience and was therefore removed.

• Heads up display (HUD) :



All HUD assets were created by me hand drawn in photoshop, taking inspirations from Ironman's HUD. The HUD is also dynamic, meaning player height affects the rangefinder on the left showing how high up the player is as well as showing whether hover mode is on and which weapon is selected. The HUD including text renders track player health, jetpack hover, player y height, weapon selected, wave number and score. You will find that texts are scaled by the resolution as well.

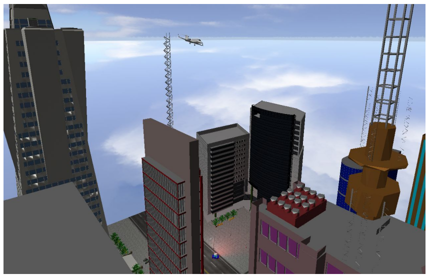
Part 2 - Camera, Meshes, Lighting, and FX:

• Final camera motion technique :

Nothing much has changed here since I went all out for this part in Milestone 1. It still provides a first person and third person view you would expect from the current game market. Though it is worth mentioning it took a while to figure out why the player model is toppling over when rotated to certain angles with Bullet. The y-axis needed to be set specifically before taking in rotation amount, it seemed others also struggled in this and I am happy I was able to help out with this. It might also be worth mentioning there's a freeview cam but it's code from the lab, it can be triggered with "T".

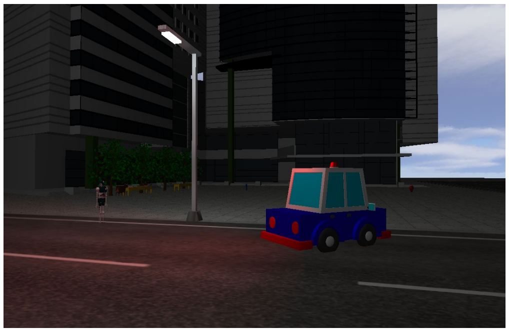
• Mesh-based objects :

There are a total of 18 mesh objects in the game. All rotated, translated and scaled correctly.



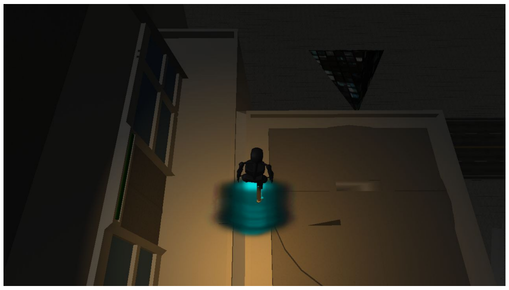
This is a screenshot featuring some of the many meshes in the game. Most of them being tall buildings as it fits with my theme of the game. Boxes were also manually created for each building that required collision (will talk about this in physics section). It is also worth noting that many meshes did not come with texture so I took it upon myself to colour them in Blender.

• Lighting :



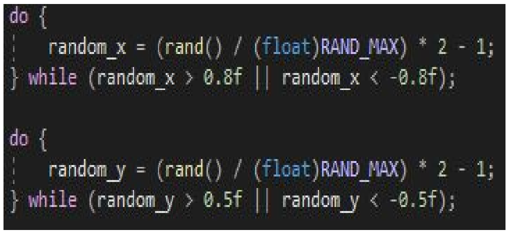
First one is a rotating red spotlight to mimic the siren of a police car. This was a tricky one to get as there is no easy method to rotate lights, I initially designed a timer that rotates the x axis and then flipping the z axis to allow rotation into the other 180 degree. This worked however it is not as smooth as I wanted it to be. With suggestions from Dr. Chris Child, I was able to achieve this by having an invisible object rotate and use the object's front vector as the spotlight's direction vector.

Second one is two white point lights I've attached to my primitively created lampposts. A glowy light material was used for the lamp part which acts as the pointlight's object. The above screenshot perfectly demonstrates how both lights beautifully integrate with the environment.



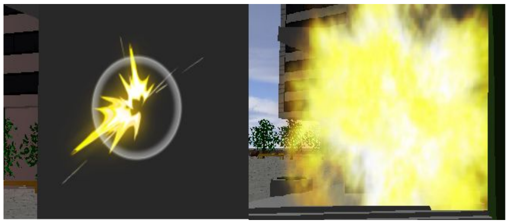
The last one is a yellowish orange spotlight which mimic jetpack's flame. The light turns on and off depending on whether the player is holding spacebar or whether hover mode is on.

• Special Effects:

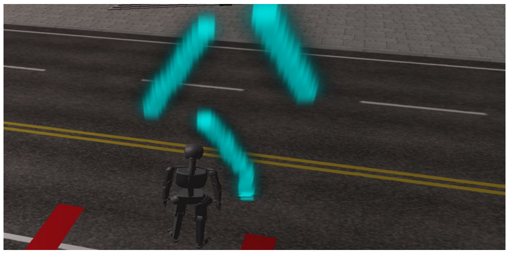


```
do {
    random_x = (rand() / (float)RAND_MAX) * 2 - 1;
} while (random_x > 0.8f || random_x < -0.8f);

do {
    random_y = (rand() / (float)RAND_MAX) * 2 - 1;
} while (random_y > 0.5f || random_y < -0.5f);
```

There are two crossfades that activate when the player takes damage, a red border and a cracked screen effect (mimicking iron man helmet). The special thing done for this is the randomised position of the cracked screen crossfade as well as guards to make sure the effect doesn't get played at the edge of the screen where it won't get half seen or missed by the player. You can spam "9" in the game to see this demonstrated.



Billboard was used for hit effects such as blast and explosions, fitting in with the theme of the weapon selection in the game. Repulsor having blast and explosive weapons having explosion.

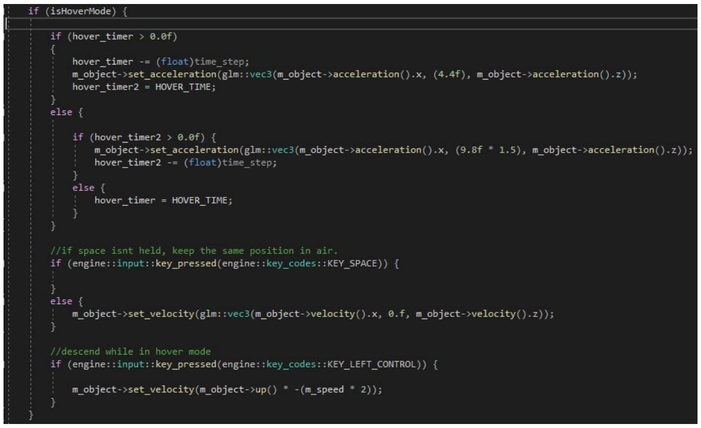


Lastly, particles were used for a trailing jetpack effect. The particles were specially designed to fall to the ground as time goes on mcmicking gravity. This allows some fun stuff such as drawing in the sky.

Part 3 - Physics, AI, and Gameplay:

• Game Physics :

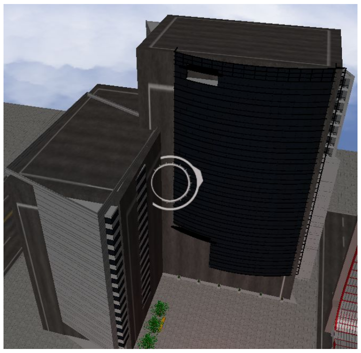
The game's physics predominantly uses Bullet with some exceptions. The implementation of flying uses Bullet's gravity, velocity and acceleration. A neat thing that I did which took a bit of tinkering with Bullet is the jetpack's hover mode.

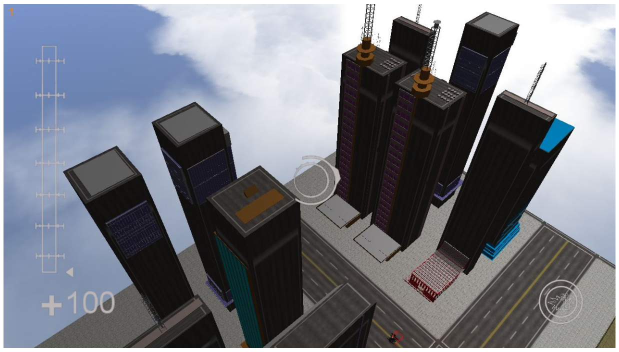


```
if (isHoverMode) {

    if (hover_timer > 0.0f)
    {
        hover_timer -= (float)time_step;
        m_object->set_acceleration(glm::vec3(m_object->acceleration().x, (4.4f), m_object->acceleration().z));
        hover_timer2 = HOVER_TIME;
    }
    else {

        if (hover_timer2 > 0.0f) {
            m_object->set_acceleration(glm::vec3(m_object->acceleration().x, (9.8f * 1.5), m_object->acceleration().z));
            hover_timer2 -= (float)time_step;
        }
        else {
            hover_timer = HOVER_TIME;
        }
    }

    //if space isnt held, keep the same position in air.
    if (engine::input::key_pressed(engine::key_codes::KEY_SPACE)) {

    }
    else {
        m_object->set_velocity(glm::vec3(m_object->velocity().x, 0.f, m_object->velocity().z));
    }

    //descend while in hover mode
    if (engine::input::key_pressed(engine::key_codes::KEY_LEFT_CONTROL)) {

        m_object->set_velocity(m_object->up() * -(m_speed * 2));
    }
}
```

By pressing "F", the player is suspended in mid air allowing greater control of shooting and other tasks while flying. The special thing I did for this is the simulation of hovering. While in hover you will notice the player going up and down slightly to mimic hovering, increasing immersion. All in all, a fun algorithm to implement.

Collisions in my game took the most development time of any other feature in the game. Bullet handles registering projectile hits on enemies while BYO was used for area-of-effect explosion damage such as missiles and bouncynade. During development for this, many students including myself have noticed the render of the AABB box is off from where the box actually is and the box doesn't actually cover the target. The box is usually ahead and to the right of the target, I spent a long time manually placing the box in the correct place in all of the AIs and explosion area box without visual aid of where the box is since the render is incorrect. I'd use missiles to constantly shoot at the edges of the box to determine where they are and manually adjust the values so they cover the target as intended. This was important as hit registration is what separates satisfaction and frustration especially for a game like mine. After countless hours, the hitboxes are near perfect, providing a polished and satisfying experience.





The second challenging part of collision for me was my environment. Due to the widely varying shapes of the buildings, bullet's default collision isn't usually accurate. In order to really sell the environment, the player has to be able to land on each building's roof seamlessly. This is why I've manually placed a terrain object box to every building in the game, customising the shape of the box to fit buildings the best I could. In the image above, you can see the boxes rendered with the road texture (the last thing submitted was road). These boxes are then submitted to bullet to let it handle the collisions and the render for these boxes are then turned off. The reason why BYO box wasn't used here is due to its inaccuracy of where they are and if we were to set the player's position back to where it was colliding with the box (like lab 7), the player won't be able to move while standing atop rooftops. In the end, I am happy with the way they turned out and definitely gives an extra sense of realism to players landing on these rooftops.

The last physics related thing is ball physics, most of it is handled by Bullet but things such as rolling friction is added. An example of these can be seen while using the bouncynade weapon (Slot 2).

• Artificial Intelligence :

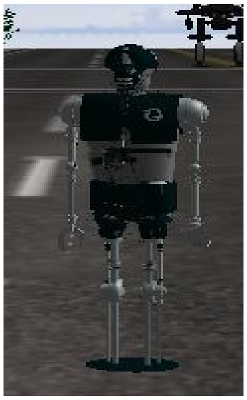
There are currently 4 distinct AI enemies the player will face.

BB8

Medium HP, medium damage, fast speed, short detection range, melee only. BB8 will run at the player and damage them, running away usually isn't an option, the best way is to fly away.

Drone

Low HP, low damage, fast speed, medium detection range, melee only, can fly. Once detected the drone will follow the player at a much faster speed, running or flying away usually isn't an option. The only way to break the chase is to perform a maneuver where you fly really high up and drop down, allowing your acceleration to "out speed" the drone.

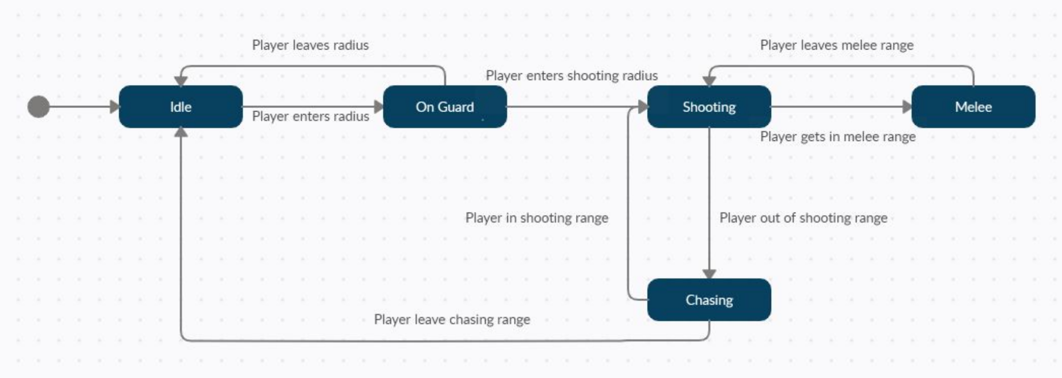


Droid

Medium HP, low damage, slow speed, fast firing speed, medium detection range. The droid will fire at the player once in range and detected. Trying to close the distance against it will result in the droid backing away from the player. However, during this the droid will fire at a much slower speed than it would have standing still. The player can then take advantage of it. Flying away is the best option against it as it has a high chase radius but it cannot fly.

Mech

High HP, high damage, slow speed, slow firing speed, high detection range. The mech is the "tank" of the bunch. If combat cannot be avoided, it is best to stay at a distance where you can time it's slow firing speed and strafe at the right time to dodge the rocket. Don't be fooled into thinking going under it will be safe, once close enough the mech will bulldoze anything near its legs for a substantially high amount of damage. Running away is not possible as it has a high detection and chase radius. Flying away is the best bet.


Player leaves radius
Player leaves melee range
Player enters shooting radius
Idle
Player enters radius
On Guard
Shooting
Melee
Player gets in melee range
Player in shooting range
Player out of shooting range
Chasing
Player leave chasing range

Here is an example of one of the state machine diagram created with app.creately.com, this is the Mech's state machine and shows the different transitions between each state.

Closing this section, the plan was to have the mech spawn two enemy drones when its hp falls below a threshold. With the way the code is setup, it can be added very easily if time permits, but I've decided to leave this feature for more important parts of the coursework.

• Gameplay Elements :

There are currently two power ups/pick ups that the player can collect. A health pickup and an immunity pickup. Health heals the player while immunity stops the player from taking damage for about 6 seconds.

There are numerous timers throughout the game such as ones that keeps track of respawn time of pickups, timer to loop tracks (m_audio_manager->loop doesn't seem to work) and timer to prevent too many jetpack particles from spawning due to framerate while holding down spacebar. And timer to give the player an invincible timeframe upon taking damage to prevent the player from taking damage from multiple instances at once.

The game currently has a wave system that respawns enemies tougher every wave allowing you to earn more points. Due to time constraint, holding a vector of predetermined positions to have the enemies randomly spawn from did not make it into the game as well as having more of the same units each wave. Similarly, the pickups could either spawn from predetermined locations randomly or have a chance of dropping from enemies. Pickups for increasing damage did not make it in time for the game as well. While in line with gameplay elements, the initial plan was to have different enemies type react differently to certain weapons (priority was handed to making those weapons first), some may be more vulnerable or resistant to certain weapons, promoting weapon switching. This is definitely something I will revisit later for my portfolio as it is a pretty important part of my plan, I have already coded a way to specifically set damage taken for different types of enemies in the collision code, so adding this will not be difficult at all.

The game also keeps track of player's score at the bottom for every damage dealt. Some weapons can deal more damage thus giving more points while others don't. On the topic of weapons, instead of introducing them in the physics section, it should be more appropriate here. There are currently 3 different weapons :

> Repulsor : Classic Ironman weapon, medium damage, fast projectile speed makes for easy target hitting.
> Missiles : Small AOE radius, low damage, medium projectile speed. Good for multiple enemies or when hitting targets directly is difficult e.g while flying (drop bombs on them instead)
> Bouncynade : Big AOE radius, high damage, slow projectile speed. Good for multiple enemies that are slow. The bouncynade arms on the first bounce, so time your shot and trajectory!

There were plans for a fourth weapon which is laser but due to time constraint has to be cut out from the game.

Discussion :

The strength of my game definitely lies with the mechanics of the game which is what I mentioned I'd want to perfect in my Milestone 1 report. The movement and flying combined with the weapons are fluid and based on my own experience of playing it, was actually pretty fun to shoot and quickly switch to other weapons for maximum damage output. Coming from a background of competitive Team Fortress 2, Apex Legends and Dota 2, mechanics of a game is what draws me to it. Comparing my game's movement to the first two FPS games I mentioned, it is definitely seamless and smooth enough to not hinder what the player wants to do. In that regard, I've done what I set out to do.

Aside from that, I've always prided myself in the level of polishness I add to all my projects, from my first year bootcamp game to the Java game and you should be able to see the same level of attention to detail for this game. Bounding boxes are manually adjusted to the right place, manual placement of boxes to the environmental buildings to provide as accurate of a collision of the render as possible to many more.

The weakness of the game at its current state is definitely repetitiveness. From all the things I've mentioned I wanted to include but had to cut out has made the game seem contentless. I am personally not too bothered by it as it is something that can be expanded further. The game was built with expansion in mind and getting the core mechanics of the player down and polishing it first before having extra features is important for me, hence the reason why you would get the feeling of playing a demo version of the game.

Overall, I am very happy with what I've accomplished in the time provided. I am especially happy with learning how to do basic things in Blender such as reducing polycount and colouring as well as learning how to read .mtl files and being able to use this knowledge to help fellow peers with missing texture. Additionally, my photoshop skills have improved by a large margin because I wanted to hand craft the HUD elements to my specific liking. Last but not least and perhaps the most important, being able to take my C++ knowledge to a whole new level. From my first time looking into an Unreal Engine and Unity code, vec3 and its associated transformation always confused me very much until now, definitely one of the more important things I learned from the course.

Lastly, I am confident the game can easily be expanded into something people will play for at least an hour. I think it is especially helpful to know everything in the code is very well polished and all I would have to do is to add new content and have them interact with the old. Expanding the map and having more enemies type and weapon will definitely diminish the game's repetitiveness. Once the map has been expanded, there can even be an objective mode where the player has to fly from point a to point b to fight enemies, unlock new weapons and skills. This has been a very intense past month for me spending all my time outside of university working on this game and a whole load more coursework in parallel (due to no exams), nevertheless this is a piece of work I'd proudly and happily display on my portfolio page.

P.S. I've noticed a strange behaviour where force applied to projectiles are lessened when you play the game in full screen, I've tested a different set of values if you wish to play it in 1080p full screen. The values can be found in example_layer.h and can be changed easily. The default values are for 720p.