CGP600 AE2

Space Game

Sean Khanna – Q11279516

Level 6

Contents

[Mechanics 2](#_Toc535414667)

[Testing 2](#_Toc535414668)

[Graphics, OO and 3D Development Techniques 2](#_Toc535414669)

[Design Patterns 2](#_Toc535414670)

[Logic 3](#_Toc535414671)

[Extra Features 4](#_Toc535414672)

[Future Improvements 4](#_Toc535414673)

[References 5](#_Toc535414674)

[Appendix 6](#_Toc535414675)

[Appendix A – GitHub 6](#_Toc535414676)

[Appendix B - Black Box Testing 7](#_Toc535414677)

[Appendix C - White Box Testing 9](#_Toc535414678)

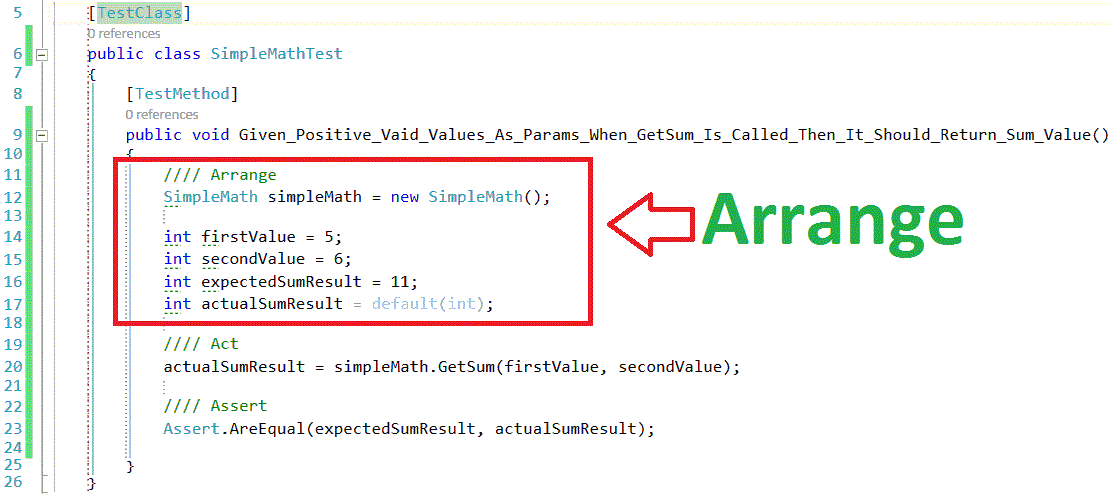
# Mechanics

My space game is a flying simulation game which allows you to free roam around a 3D environment dodging asteroids and enemies which try to chase you down and kill you. You as the player are capable of shooting lasers towards your enemies or asteroids to score more points, there is no win or lose state just an alive or dead state. AS I said in my previous report this game is very similar to way No Man Sky works in space when flying around.

No Man’s Sky, 2018. Ebunny3k.com.

# Testing

As part of my testing I did white box testing (See [Appendix C](#_Appendix_C_-)) and black box testing(See [Appendix B](#_Appendix_B_-)). For one of my black box tests, tested the collision and found an issue were if an object was pushed faster away from another object the original moving object would speed up exponentially. I found out later that this was caused because I was not normalising my direction vector. To normalise a direction vector, I needed to get my direction and then use Pythagoras to work out the magnitude of the vector and then use that to normalise it.

Due to the amount of time I had to make this program I was unable to implement any other forms of testing unit testing included. I’ve done unit testing in the past using the Triple A method, Arrange, Act and Assert.

Unit Test, 2017. HR Rony.

# Graphics, OO and 3D Development Techniques

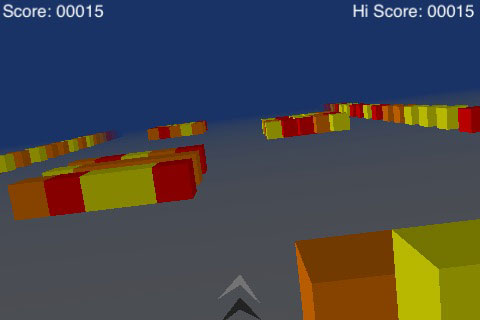
My game uses a multiple number of different shaders and lighting techniques. Additionally, to my model shader I use a reflect shader which allows my objects to use all the lights needed and then be able to reflect the skybox used using the applied lights. Unfortunately, you are not able to see the players reflection as that can get very complicated however, if I had more time on this project, I would have definitely implemented that.

# Design Patterns

I kept the design of my game to the original design as I felt that we had designed a very good game and that it needed very little adjustment. I did however have a very different setup of my classes than pre-designed, mainly because we were still learning parts of DirectX and how they can be structured.

We had previously designed a loss flow chart however, I did not use that, as I don’t have a loss or win condition within my game.

When designing the movement, I have managed to stick to the flowchart as best as I could furthermore, I was able to implement Xbox 360 controller functionality including, vibrations when colliding with objects like asteroids.

In the original design I had spoken about making the game similar to an old game I used to play called cube runner. However, I didn’t want my game to be based on just the Z-plane, I wanted my players to be able to fly around in a 3D environment.

Cube Runner, 2008. Arn.

# Logic

Within my game I use collision detection, using sphere colliders. When using sphere colliders, it makes it very easy to detect when a collision happens you begin by finding the distance between two objects by using Pythagoras.

Then we work out the radius of the sphere by finding the furthest vertex away from the centre of each object in question, add them together for a centre to centre distance. Finally, it just comes down to a simple if statement.

I did however want to additionally, implement collision detection using polygons on a the objects to give me better collision detection and be able to us that information to apply the correct forces to objects, in the correct direction. This meant I would have to use the dot product and the cross product to help me calculate the points of intersection.

Dot Product

Cross Product

# Extra Features

I was able to include many extra features within my game from very simple things like, keeping the aspect ratio when going into Fullscreen; using Xbox controller and its vibration motors; sound and music; FPS counter and delta time applied to all objects.

# Image result for no mans sky flyingFuture Improvements

No Man’s Sky, 2017. Boston Blake.

If I had a little more time, I wanted to apply 3D sounds using X3DAudio, using this I could be able to apply engine sounds so that you can hear where the enemies are located around you. Additionally, I wanted to create a mini-map of where we are and where enemies are, so you could have further information to help you with the game. I had attempted to use specular shaders within my game unfortunately, I was unsuccessful but, I am looking to use it in future projects.

 Another feature I wanted to implement was high quality explosions when the lasers hit the asteroids and other explosions when they got destroyed. As we very limited with the amount of time for this game I was unable to achieve this goal but, plan to use it in the future.

Fractured Space, 2016. Ben Barret.

# References

Ebunny3k.com, 2018. No Man’s Sky NEXT – Where to find Chromatic Metal, Pure Ferrite, Sodium Nitrate, Hermetic Seal [viewed 15/01/2019]. Available from: <http://ebunny3k.com/2018/07/104537/no-mans-sky-next-where-to-find-chromatic-metal-pure-ferrite-sodium-nitrate-hermetic-seal/>

Arn, 2008. Custom Level Packs for Cube Runner [viewed 15/01/2019]. Available from: <https://toucharcade.com/2008/07/22/custom-level-packs-for-cube-runner/>

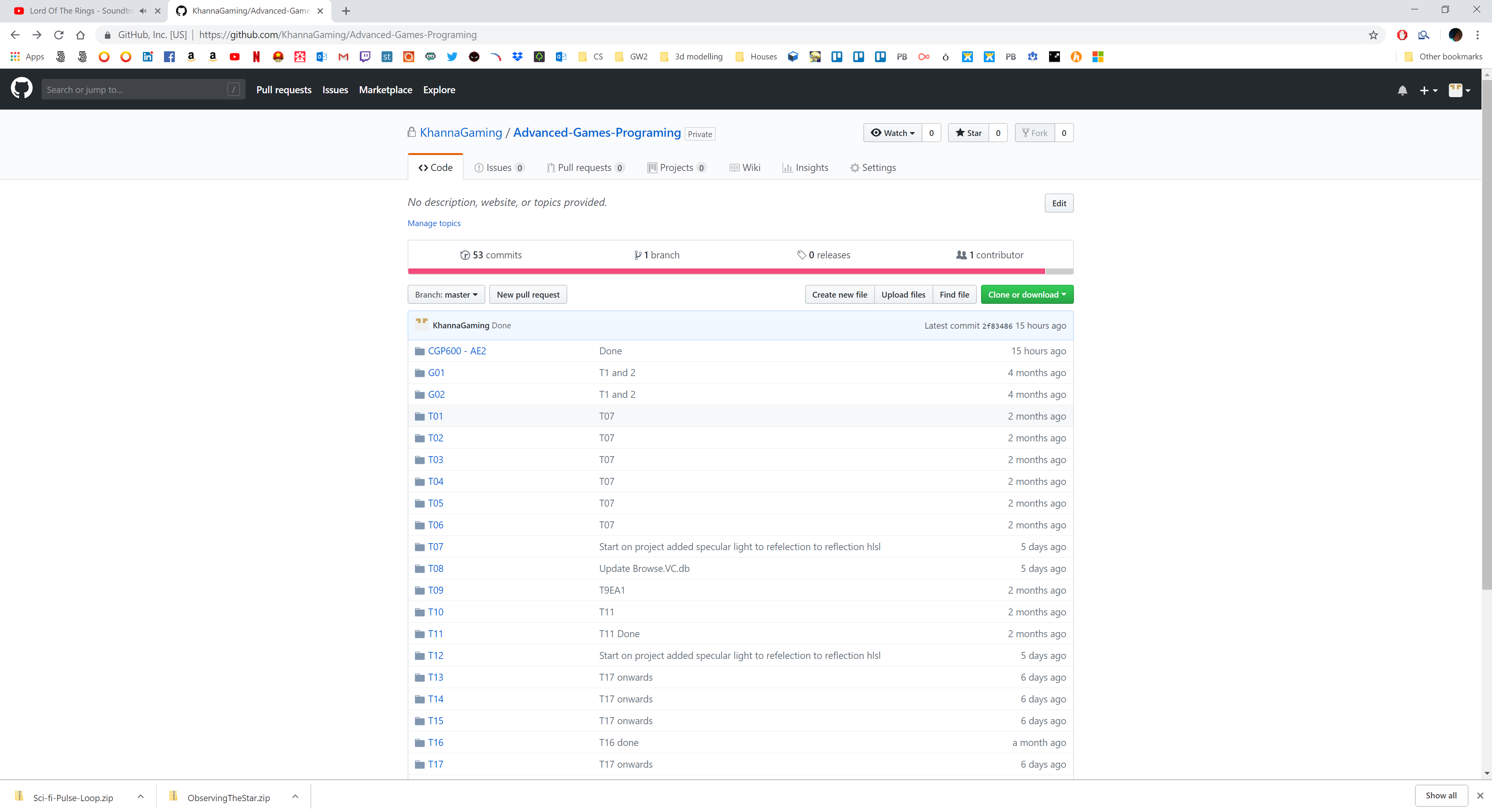
Ben Barret, 2016. Fractured Space introduces custom games aimed at scheduled clan games [viewed 15/01/2019]. Available from: <https://www.pcgamesn.com/fractured-space/fractured-space-introduces-custom-games-aimed-at-scheduled-clan-games>

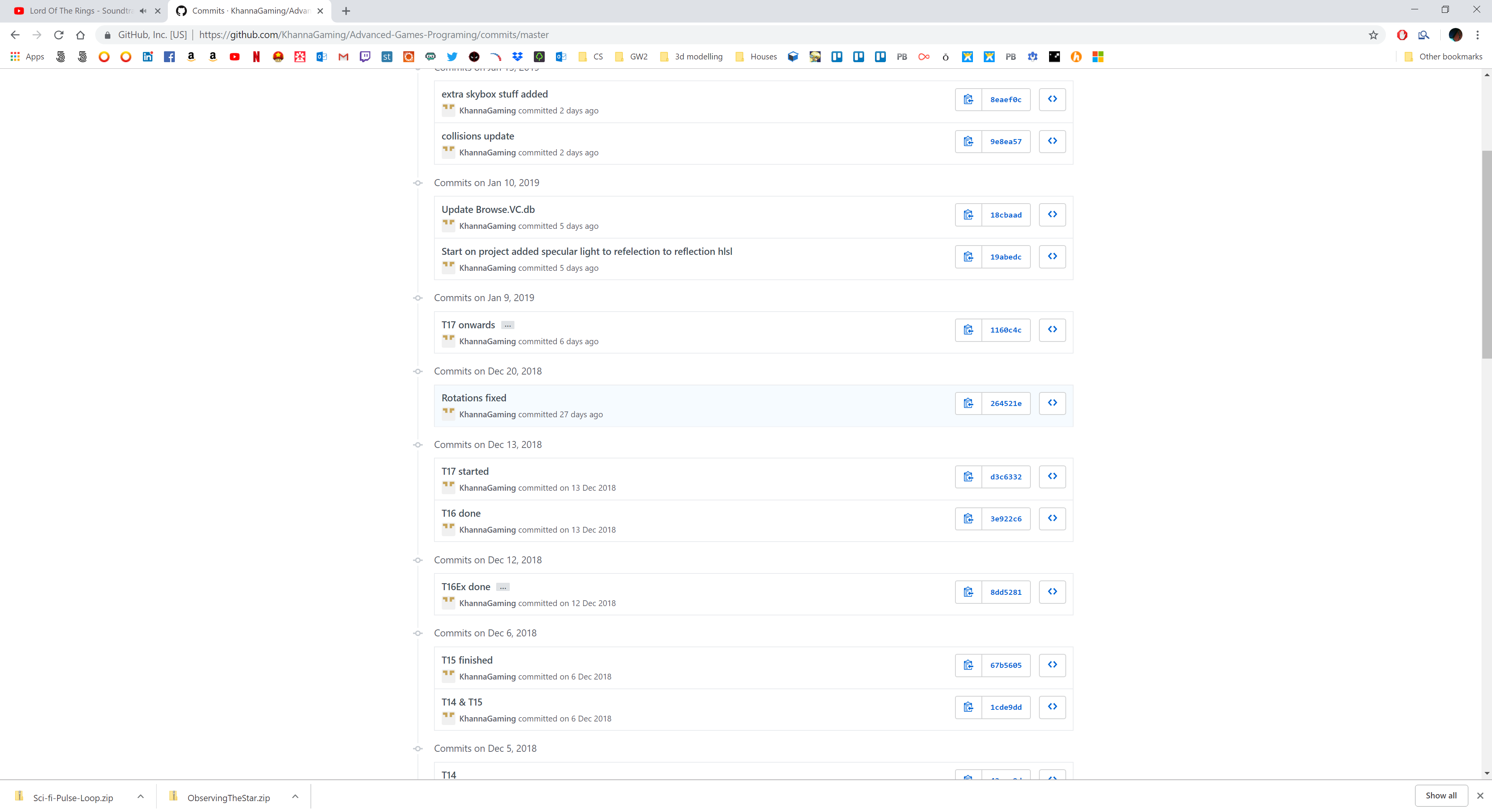
Boston Blake, 2017. No Man’s Sky Experiencing Frame Rate Drops When Flying Too Fast [viewed 15/01/2019]. Available from: <https://gamerant.com/no-mans-sky-fast-flying-frame-rate-drops/>

HR Rony, 2017. Unit Testing - BDD, AAA Structure And Object Mocking [viewed 15/01/2019]. Available from: <https://www.codeproject.com/Articles/1182279/%2FArticles%2F1182279%2FUnit-Testing-BDD-AAA-Structure-And-Object-Mocking>

# Appendix

## Appendix A – GitHub





## Appendix B - Black Box Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Feature | Test | Expected | Actual | Solution |
| 1 | Collision | Player hits object | Object moves away in opposite direction. | Player moved inside of object. | Stop adding the camera move vector when collided. |
| 1 – Retest 1 |  | Player hits object | Object moves away in opposite direction. | Object moved in same direction as player not the direction it hit | Work out the direction between the camera and the object and multiply it by the move speed. |
| 1 – Retest 2 |  | Player hits object | Object moves away in opposite direction. | Object moves away in opposite direction. |  |
| 2 | Collision | Object to object collision | Objects bump into other objects and then slow down themselves or stop depending on whether the other object is moveable. | When one object collides with another object, they just get stuck inside each other. | Direction vector being calculated wrong. |
| 2 – Retest 1 |  | Object to object collision | Objects bump into other objects and then slow down themselves or stop depending on whether the other object is moveable. | When one object collided with another, it would push it away but the further away the object would move the velocity of the first object would exponentially get higher. | Normalise direction vector. |
| 2 – Retest 2 |  | Object to object collision | Objects bump into other objects and then slow down themselves or stop depending on whether the other object is moveable. | Objects work in the positive axes but move positively in the negative axes. | Removed direction vector as it was a negative multiple as well, making it become a positive. |
| 2 – Retest 3 |  | Object to object collision | Objects bump into other objects and then slow down themselves or stop depending on whether the other object is moveable. | Objects collide and move right speed, but they only move in the same direction as the original object not the direction. | Unfixed, future games consider using elastic collision |
| 3 | Sounds | When shooting, a sound is played | Laser sound plays | Laser sound plays only once and then not again. | When it is needed to play again Stop playing the old sound, flush the buffers and then play from the beginning. |
| 3 – Retest 1 |  | When shooting, a sound is played | Laser sound plays | Laser sound plays |  |
| 4 | Sounds | When laser collides with any object, a sound is played | Explosion sound plays | Explosion sound plays |  |
| 5 | Sounds | When game starts music plays | Music track plays and loops | Music track plays but, not loops | Change loop count to infinite. |
| 5 – Retest 1 |  | When game starts music plays | Music track plays and loops | Music track plays and loops |  |
| 6 | Controller | Collide with objects | Controller vibrates | Controller vibrates |  |
| 7 | Enemy Chasing | Chase player when in range | Chases Player when in range | Chases Player when in range |  |
| 8 | Enemy Fleeing | Flees when low on health | Moves in opposite direction of player | Moves to random location not opposite direction | Calculating look at vector wrong. Calculate by taking its position and minus players position minus its position. |
| 8 – Retest 1 |  | Flees when low on health | Moves in opposite direction of player | Moves in opposite direction of player |  |

## Appendix C - White Box Testing

|  |  |  |
| --- | --- | --- |
| ID | Test | Evidence |
| 1 | Make sure Laser removes itself correctly from player |  |
| 1 - works | Make sure Laser removes itself correctly from player |  |
| 2 | Test to make sure health works currently and subtracts when bullet collides |  |
| 2 - works | Test to make sure health works currently and subtracts when bullet collides |  |
| 3 | Make sure Laser correctly adds itself back to the object pool. |  |
| 3 - works | Make sure Laser correctly adds itself back to the object pool. |  |