D00M3D Game Report

Members:

Luke Harper - 100886836

Nick Chiasson - 100891716

Skyler De Francesca - 100893298

D00M3D

D00M3D is an action/survival space game where the goal is to survive as long as possible. Enemies spawn and follow the player around and will attack the player if he/she is within a certain range. Asteroids also endlessly spawn as the player moves around in space. Destroying enemies and asteroids will spawn a retrievable item that adds to one of the player's many resources. The player must not travel too far from the large orange star as it is used to recharge the player's energy. The player must be cautious when retrieving energy from the star as if they get too close the star's heat and radiation starts to reduce the player's health.

Game Manual

Controls:

This game must be played with a computer mouse and keyboard.

W - Forward thrusters

A - Left thrusters

S - Backward thrusters

D - Right thrusters

Q - Upward thrusters

C - Downward thrusters

UP - Pitch up (also mouse up)

LEFT - Yaw left (also mouse left)

DOWN - Pitch down (also mouse down)

RIGHT - Yaw right (also mouse right)

V - Roll counter clockwise

B - Roll clockwise

Scroll Wheel - Change camera perspective (scroll in toward first person perspective; scroll out toward third person perspective)

SPACE - Fire weapon

1 - Switch to laser cannon

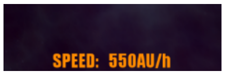
2 - Switch to plasma cannon

3 - Switch to black hole cannon

Esc - Exit game

How to play:

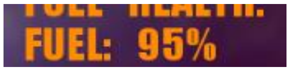
When you first spawn into the game you will see that you are surrounded by asteroids and will soon see enemy spacecraft starting to attack you. The first thing you should do is begin to move and gain substantial velocity because enemies tend to attack more while the player is travelling at a slow velocity. You can move your spacecraft by initiating the thruster(s) in the direction you want to go (controlled by the keys “W,A,S,D”). Your speed is displayed in the bottom right corner of the window:



SPEED: 550AU/h

<u>FUEL</u>

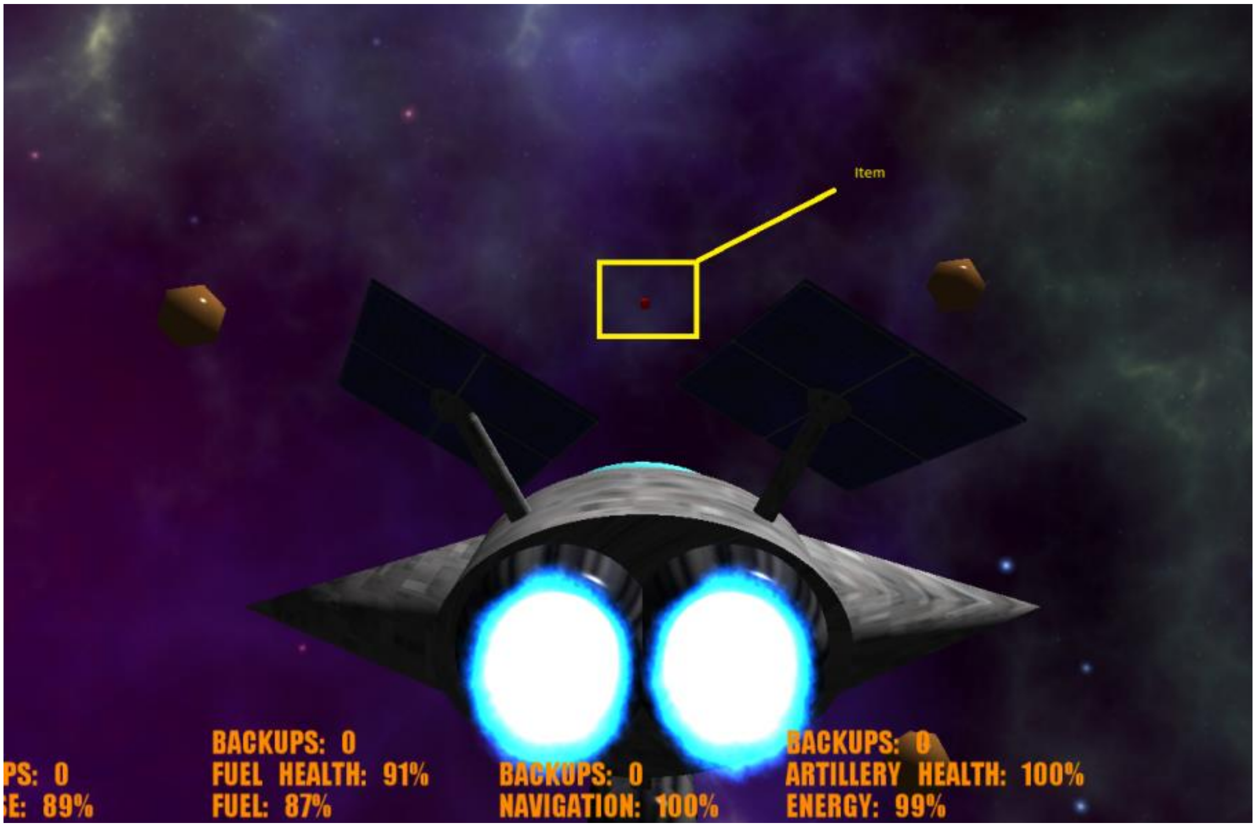
Be careful how often you use the thrusters! You have a limited amount of fuel and will run out if you are not cautious about the use of your thrusters. You can see how much fuel you have left at the bottom of the window:



FUEL: 95%

The amount of fuel you have is displayed as a percentage and will decrease whenever the thrusters are activated. When you run out you will not be able to use the thrusters anymore which means you will not be able to change the ships velocity!

A solution to this problem is to mine the asteroids and collect the fuel item that they drop. You can destroy asteroids by shooting them with any weapon. A small red cube will spawn in the center of the decimated asteroid which you can “pick up” by colliding with it. Be cautious when approaching the item! They are very small relative to the asteroid and can be hard to retrieve if the ship is going too fast. It is best to approach the item at a slow velocity to ensure efficient retrieval.



PS: 0
E: 89%

BACKUPS: 0
FUEL HEALTH: 91%
FUEL: 87%

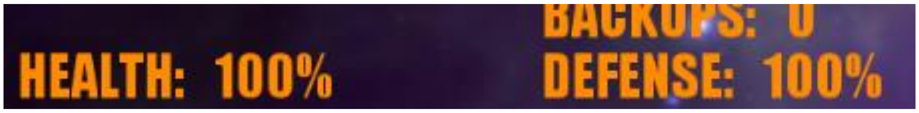
BACKUPS: 0
NAVIGATION: 100%

BACKUPS: 0
ARTILLERY HEALTH: 100%
ENERGY: 99%

Picking up these fuel items will increase your fuel percentage by a certain amount and will extend the longevity of your life in the game

Health and Defense

At the bottom left corner of the screen is the indicator for how much health the player has. To the right of it is the indicator for how much defense the player has:



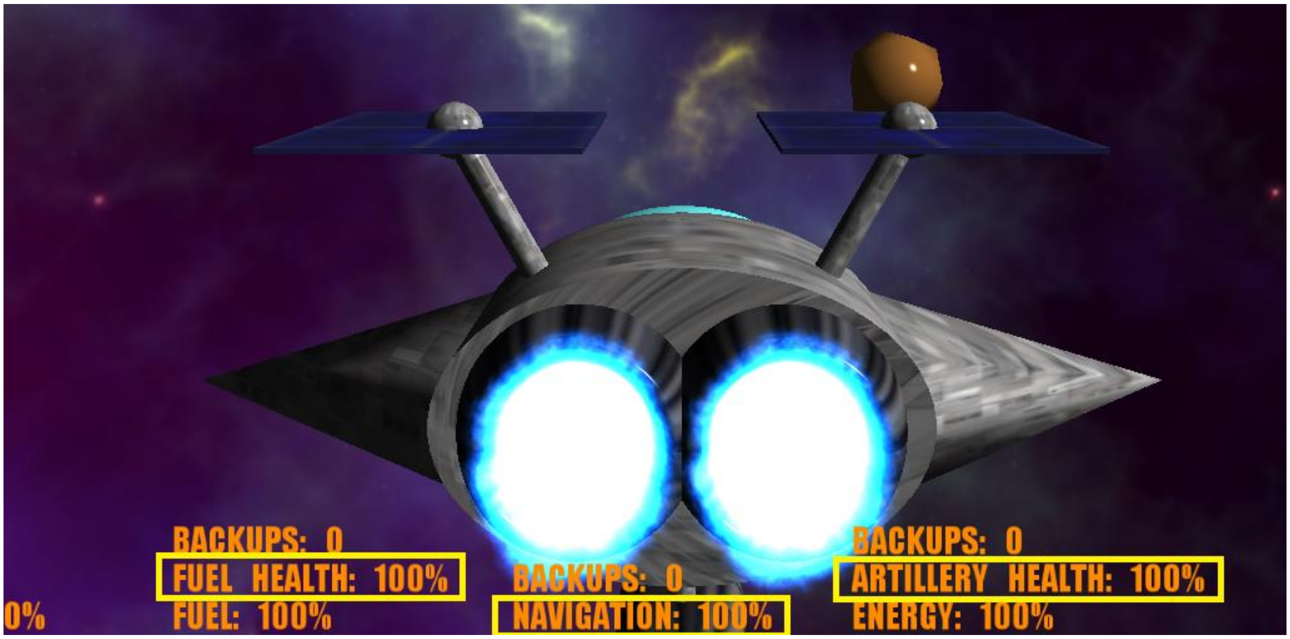
HEALTH: 100%

BACKUPS: 0
DEFENSE: 100%

Think of the defense as the “shield” of the ship. Once the ship has no more defense (i.e. the ship’s defense falls to 0%) the health of the ship (rather than the defense) will begin to diminish. One health hits 0% you are dead and the game is over.

There are multiple ways the ship can take damage to its health/defense. Firstly when an enemy shoots you with its weapon your ship will take damage. Secondly, your ship will take damage if it gets too close to the star.

Each of the ship’s systems (energy/artillery, fuel, and navigation) can also take damage. Once that system’s damage reaches 0% you will no longer be able to use that system! These systems are damaged if an enemy hits the part of the ship the system is associated with.



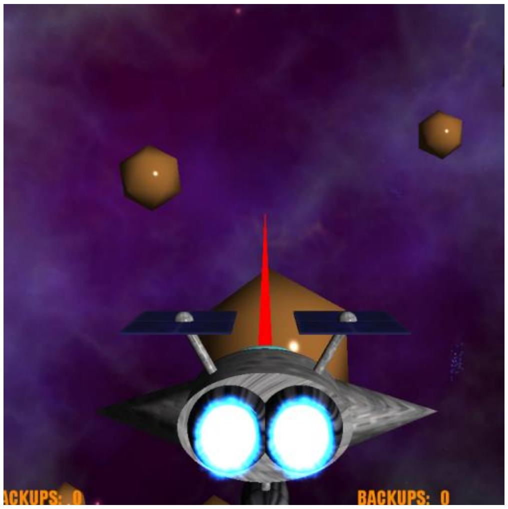
The values highlighted above are the values that represent each of the ship’s systems’ health.

One way to repair your ship’s defense is to kill enemies and pick up the item they drop. The item looks identical to the item the asteroids drop when destroyed. Picking up enemy items will fill up your defense, and therefore preserving the health of the ship.

Energy/Artillery

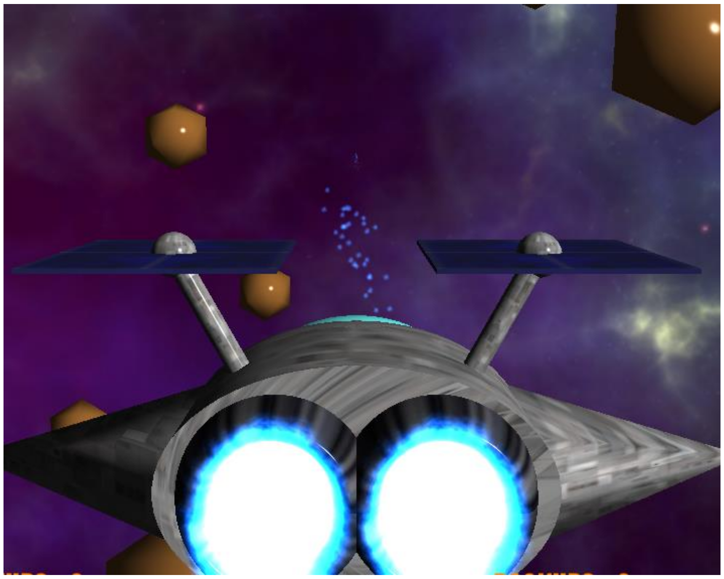
The energy system is extremely important for the ship’s survival. The ships energy is what powers the ships weaponry/artillery! Like the other ship systems you can run out of energy and need to replenish it!

The ship has three different weapons. The first weapon is a laser cannon:



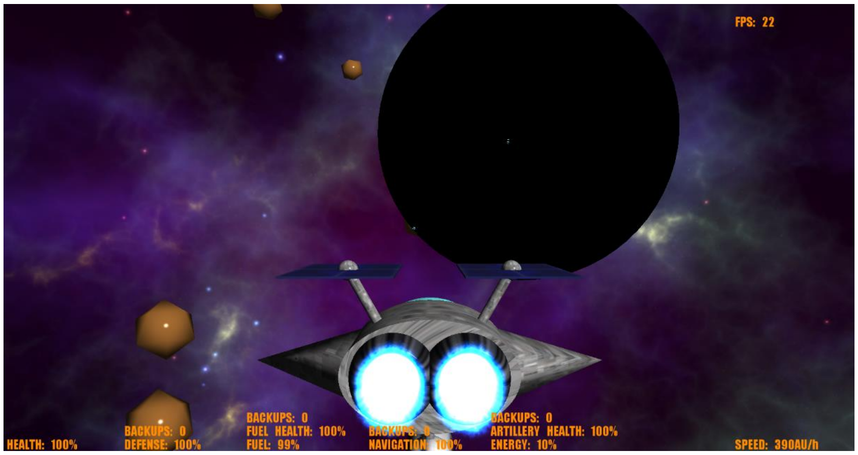
The laser cannon is fairly simple; it shoots a laser beam straight out from the ship's "gun" and will damage anything it comes into contact with. The laser cannon uses very little of the ship's energy but is not as powerful as the other weapons. You can switch to the laser cannon by pressing "1" on the keyboard.

The second weapon that the ship has is a plasma cannon:



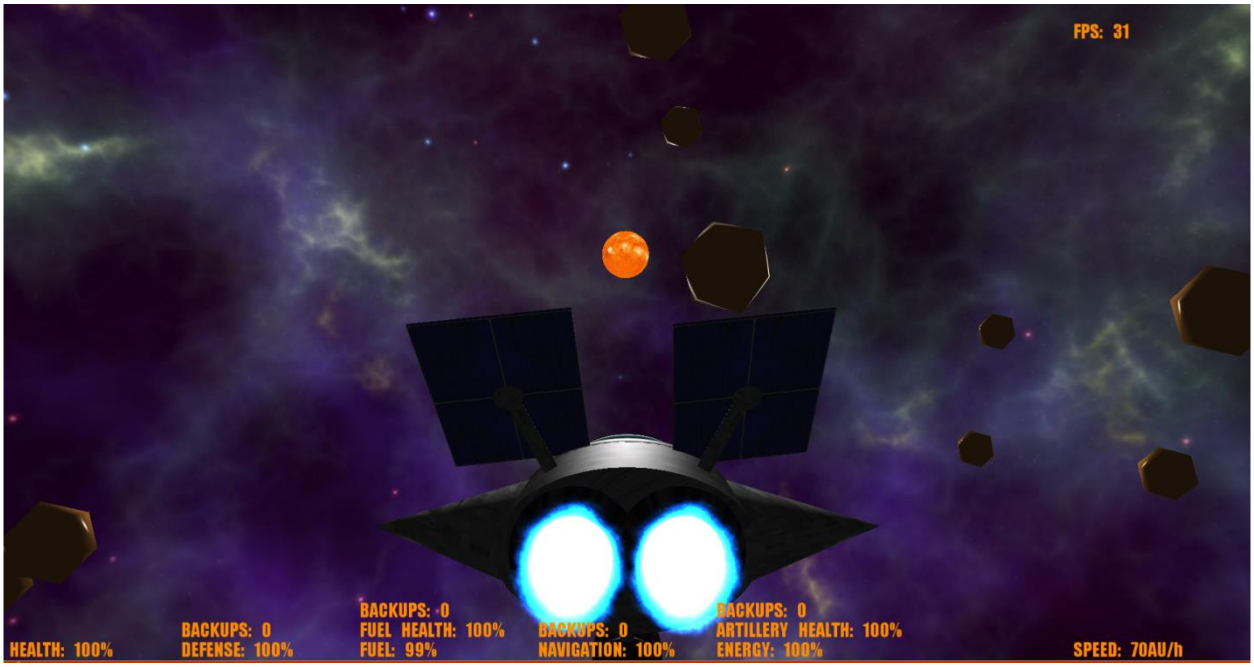
The plasma cannon shoots out plasma projectiles from the ships cannon. Using this weapon is more difficult than using the laser cannon as hitting enemies/objects farther away requires you to anticipate where the plasma projectiles and the object/enemy will meet. Though it is more difficult to use, the plasma cannon does twice the damage than the laser cannon. It is ideal to use in close combat situations with enemies. Be sure to watch the ship's energy levels while using the plasma cannon as it uses more energy than the laser cannon. To switch the ship's weapon to be a plasma cannon, press "2" on the keyboard.

The third and final weapon is the black hole cannon. This is the deadliest weapon, but also uses up the most energy. A single shot uses 90% of the ships energy so be weary when you use it! This weapon fires a projectile from the cannon which then expands into a black hole. The black hole will pull everything around it towards it and anything that gets sucked in will die!



To use the black hole cannon press “3” on the keyboard.

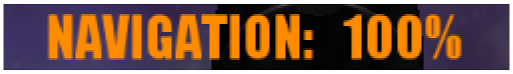
If you begin to run low on energy, there is a way to replenish it! Not too far from where the player spawns is a large, orange star:



If you travel close to the star you will begin to gain energy back! Be sure not to travel too close to the star as it will start to damage the ship’s health. If you ever have trouble locating the star, you can use the solar panels on the ship as a “compass”. The solar panels are constantly tilting so that they are facing the star!

Navigation

At the bottom of the window is the indicator for how much navigation health the ship has left:



NAVIGATION: 100%

If the wings of the ship are damaged by enemy spacecraft, the navigation health will decrease. Once it hits 0% you will not be able to change the ship's orientation!

Technical Requirements

Our project meets all technical requirements:

Both first-person and third-person camera (quaternion camera)

- The user is able to switch between first-person and third-person mode using the scroll button on the mouse.
- The camera is represented as a quaternion

Free flying with full ability to turn along 3 axes (use quaternion orientation)

- Using the mouse, the user is able to turn the ship freely along all 3 axes.
- The ships orientation is represented as a quaternion.

An armada of computer-controlled ships; at least 3 ship types

- There are three different types of enemy ships: speed, resources, and tank. The speed ship has more thruster power, the resource ship collects more resources, and the tank takes less damage. The player ship is the same as the resource ship.

Textured objects

- Each individual part of each ship as well as the sun is textured. We have also implemented a skybox that is textured to look like outer space.

Collision detection between player spaceship and other game objects

- All objects will collide with each other and react somewhat realistically to collisions.

Hierarchical spaceship design with independently moving parts: e.g., rotating turrets or orbiting shields

- The player spaceship has solar panels which move independently from the ship and always face the sun.

- The speed ships have a spinning antenna that moves freely from the ship, and the tank ships have an orbiting screen/shield.

Weapons fire and destroy objects; at least 3 different weapon types with distinct visual effects (different particle filter effects)

- As described in the game manual there are three types of weapons, each with a distinct visual effect.
- The laser cannon shoots out a scaled cylinder.
- The plasma cannon shoots out particles animated by a spline.
- The black hole cannon shoots out a black sphere which expands and sucks objects in.

Spline trajectories for some objects (e.g., exotic weapons or explosions)

- The plasma cannon uses a spline animation to animate the plasma particles when shot from the spaceship.

Additional Features

UI

An additional feature we added to the game is a user interface that displays information important for the user to know while playing the game. The user interface displays the levels of all the player ship's systems' values and health. It also displays the overall health of the ship, the speed of the ship, and the frame-rate the game is running at.

Skybox

We implemented a skybox as an additional feature to the game. We used ogre's built in skybox to apply our space texture.

Physics

Our game uses a custom built physics engine to perform physics operations on game objects to make the game somewhat realistic. Objects like the star apply a gravitational force to other objects in the game (player, enemies, etc.).

Game Architecture

ENTITIES

Every object in the game is an entity. The entity class has all the base information for each object and handles the object's interaction with ts own scene Node.

Physics Entity is a derived class of entity that contains all of an entity's physical properties like mass, velocity, collision information. Physics entities are handled by a physics engine for updating according to the world around it.

Both physics entity and entity are abstract. Everything in the world is a physics entity

PhysicsEngine

The Physics Engine updates all of the physics entity's physical states in the game. When a physics entity is created it attaches itself to the PhysicsEngine, and when the physicsEntity dies, it detaches. Our physicsEngine handles updating all the physicsEntity's states as well as managing all of the game's collision detection.

WORLD

The world class is our game runner; it contains the Camera, the input manager, the physics Engine and the particle engine. It handles the Ogre updates and facilitates updating all of the entities in the world, as well as the particle engine and physics engine. This class controls the spawning of all entities and also holds the containers in which the entities are stored. The world class also holds and calls the update for the UI. It also contains all the static functions used by our input manger to handle player input.

Particle Engine

The Particle Engine manages creating and updating all of the alive particles in the game. It contains different meshes that particles can be set to.

Spacecraft

Our spacecraft objects are physics objects and can be one of two derived classes, an enemy spacecraft which is AI controlled, or a player spacecraft which is player controlled. The spacecraft has systems that its' responsible to update and track in its own update function.

UI

The user interface class (textRenderrer.cpp) uses Ogre’s built-in overlay and text functions to display information onto the screen. It is passed a reference to a player object so that it can access the player’s data in order to display it. It also is passed in the viewport so that the UI will work on every computer screen no matter the size. The UI is controlled and updated in the world class.

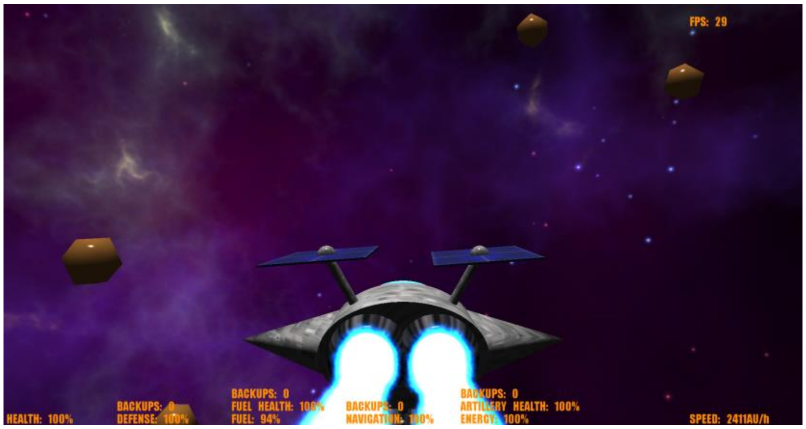
Player

The player class (player_spacecraft) inherits from the physics entity class since it is an entity affected by physics. All of its physics is handled through the physics engine.

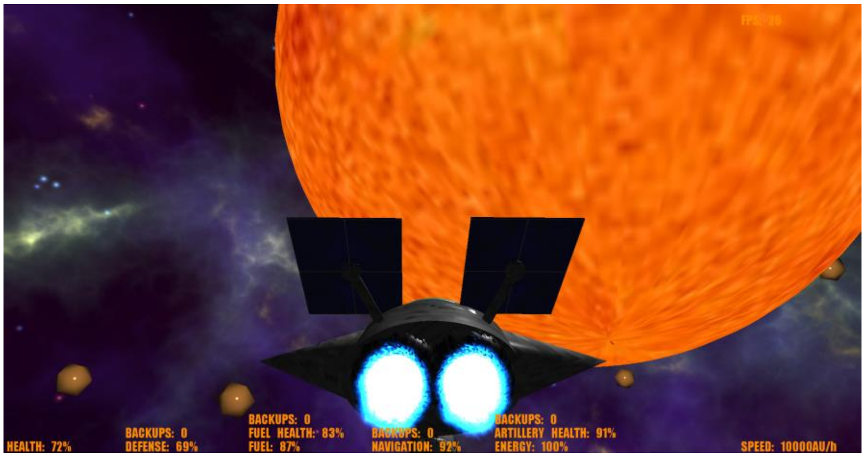
Mesh Factory

The mesh factory allows for geometrical meshes to be created at a more abstract level. The mesh factory class provides functions to create all the geometry and meshes needed for the game.

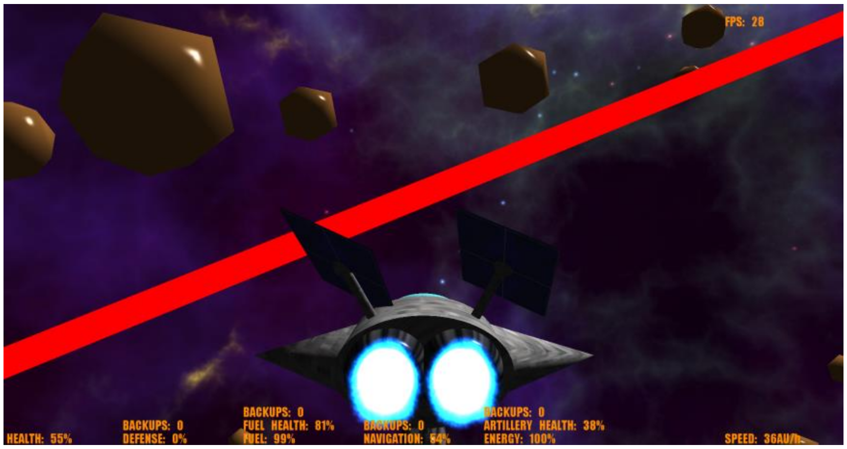
Appendix of supplementary material



Moving the ship forwards will cause the back thrusters to be initiated creating a cool particle effect.



This image shows the player getting a little too close to the star. They are losing health as well as gaining energy at the same time. You can also see that the solar panels are facing the star.



This image shows an enemy attacking the player with its laser cannon.