# Controls

|  |  |
| --- | --- |
| **Button** | **Function** |
| Q | Quit the Game |
| 🡨 | Turn Ship Left |
| 🡪 | Turn Ship Right |
| ˄ | Accelerate Ship |
| Right CTL | Fire Bullet |

# Program Flow

## Initialisation

At the start of the game’s running, the setup() method is called, which is responsible for initialising the necessary objects and variables. First, the ODE physics world, which is responsible for handling all simulations of motion and collision, is created and the necessary physics objects are initialised. Then, all necessary objects (which model components such as the player, asteroids and barriers) are initialised. These objects include

* ODE World created.
* Objects Initialised
  + OF Bounding box created
  + Collada Model loaded and scaled (for asteroids and player)
  + Physics Body Initialised
* OF Camera and light initialised

## Update

* Called before each draw loop.
* Objects updated.
* Physics increased in steps
* Collisions Processed
* Destroyed asteroids and bullets re-initialised (bullets set to unfired).

## Collision Resolution

* Players and asteroid collisions ended the game 🡪 score outputted to console.
* Asteroids or players colliding with barriers will mirror them across arena (open boundary).
* Bullet and Asteroid collision 🡪 asteroid destroyed and respawned, bullet destroyed too.
* Bullet and barrier collision 🡪 bullet destroyed.

## Draw

* Set background colour.
* Rotate and position camera to always be behind and facing the player’s direction.
* Start the light
* Draw the plane (black).
* Player drawn
* Asteroids and barrier drawn.
* Lights and camera ended
* Score string stream displayed

# Evaluation