

Assignment 2 – Asteroid Arena 3D

s3850825

You Chan Lee

Summary of what I've implemented

ARENA (HARD)

If the ship gets close to any wall of the arena, the colour of the wall will be changed from light grey to red. Only the wall which the ship is getting closer to will change the colour by calculating the distance between the wall and the ship. (Please check line 700 – 791 in gameobject.cpp)

If the ship collides with any wall of the arena, the ship's direction will be changed instead of dying. Reverse direction vector will be calculated depend on the wall. (Please check line 653 – 692 in gameobject.cpp)

I've drawn a SkyBox with the given PNG files. The SkyBox is fully textured, but each wall of the arena is drawn by GL_POLYGON with an alpha value by glColor4f for transparency. (Please check line 16 – 74 in main.cpp, 31 – 46, 278 – 290 in gameobject.h, and 793 – 887 in gameobject.cpp)

SPACESHIP & GUN MODEL (MEDIUM)

I've drawn a ship with the given ship.obj file. The ship is fully textured by ship.png file. (Please check line 16 – 74 in main.cpp, all lines in objectfiles.h, 31 – 46, 201 – 212 in gameobject.h, and 105 – 148 in gameobject.cpp)

I've tried to use free obj file of 3D bullet online, but it looks like a sphere because of its size, so I've decided to use just gluNewQuadric with some different slice and stack values. There is no point to use a lot of vertices which makes bit of buffer especially when we draw a lot of bullets. Therefore, I've used gluSphere and a free texture image from online for drawing bullets. (Please check line 14 – 74 in main.cpp, 31 – 46, 217 – 227 in gameobject.h, and 382– 438 in gameobject.cpp)

For the same reason, I've decided to use gluNewQuadric with some different slices and stacks for missiles. I've also used a free texture image from online for drawing missiles. (Please check line 14 – 74 in main.cpp, 31 – 46, 264 – 274 in gameobject.h, and 569 – 588 in gameobject.cpp)

For the HARD section, I'm not sure my hierarchical structure of game objects is appropriate to this, so I marked this part as MEDIUM. However, please have a look at my gameobject.h file because spaceship, asteroid, bullet, explosion, drone, missile, and skybox are all in the gameObject class. Each object classes are written by hierarchical inheritance. (Please check line 100 – 290 in gameobject.h)

ASTEROID MODEL (MEDIUM)

I've tried to generate random vertices which have random x, y, and z values with random angle from the centre, which is what I've done for assignment 1, but somehow it shows some empty faces, and it looks more like 2D because of empty spaces. Therefore, I've decided to use gluNewQuadric with random tessellation values for slices and stacks with the given 4 asteroid image files. Some look like a sphere, and some look like a diamond. (Please check line 16 – 74 in main.cpp, 31 – 46, 170 – 193 in gameobject.h, and 32 – 67 in gameobject.cpp)

The code I've tried to draw random vertices for asteroids

```
void asteroid::generateAsteroidOutline(float max, float min) {
    // add vertex every 30 degree
    float count = 12.0f;
    for (float i = 0; i < count; i++) {
        float apie = PI * 2.0f / count;
        const float x = cosf(apie * i);
        const float y = sinf(apie * i);
        const float z = -1.0f + GetRandom(2.0f);
        vec3f outLineVector(x, y, z);
        outLineVector = outLineVector.normalize();

        float randnum = GetRandom(max - min);
        outLineVector = ((outLineVector * min) + randnum);
        m_listOfOutlinePoints.push_back(outLineVector);
    }
    radius = (max + min) / 2.0f;
}
```

ASTEROID MOVEMENT (HARD)

All the asteroids have random positions which are outside the arena, rotation direction, speed, and radius size. They never collide outside the arena, but as soon as getting into the arena, they will collide each other, also they will bounce off any of arena walls. Lastly it has 4 wave levels, so as wave level goes higher, more asteroids will be launched. (Please check line 170 – 193, 314 – 323 in gameobject.h, and 32 – 85, 156 – 265, 590 - 652 in gameobject.cpp)

LIGHTING (HARD)

I've used one animated point light for the camera and ship. As camera keep following the ship, the light also needs to follow the camera position. This light has reddish colour for more realistic view. (Please check lines 125 – 185 in main.cpp)

I've used one fixed point light for the entire objects in the arena. Therefore, when asteroids come into the sight of the ship, they look more like reddish. Otherwise, they will get natural light from the ceiling of the arena. (Please check line 695 – 792 in gameobject.cpp)

BULLETS and SHOOTING (MEDIUM)

Bullets are fully textured by the free image file (bullets/bullet.jpg) from online. All the bullets and missiles will be launched from the head of the ship, and they are facing the camera because they are following the ship's position and direction when they are created. And if any asteroid's radius is bigger than 2.5f, then it has 2 hit points (called 'energy' here). Therefore, if a bullet hits them, the asteroid will get cracks. After that if another bullet hits them, then they will split into 2 small asteroids. (Please check line 16 – 74 in main.cpp, 170 – 193, 217 – 227, 314 – 323, 329 - 347 in gameobject.h, and 32 – 67, 274 – 340, 342 – 378, 382 – 438 in gameobject.cpp)

EXPLOSIONS (HARD)

Explosion particles are fully textured by the free image file (bullets/explosion.png) from online. Random number of explosion particles will be created from the same position of the asteroid just exploded, and their direction, speed, and radius will be randomised. (Please check line 31 – 46, 233 – 244, 349 – 385 in gameobject.h, and 442 – 458 in gameobject.cpp)

CAMERA and SHIP MOVEMENT (HARD)

The ship will be manipulated by 'w', 'W', 'a', 'A', 'd', and 'D' keys and left button of mouse. Missile will be toggled by 'z' or 'Z' key, and drone will be launched by 'x' or 'X' key. (Please check line 237 – 307 in main.cpp, 249 – 274, 390 – 459 in gameobject.h, 89 – 103, 462 – 535, 569 – 588 in gameobject.cpp)

I've set the camera position as 15.0f backwards and 7.5f upwards from the ship. Basically, camera keeps following the ship's position and set the same direction with the ship. If 'c' or 'C' key is clicked, then camera shows the rear side of the ship. If 'q' or 'Q' key is clicked, then camera shows the left side of the ship. If 'e' or 'E' key is clicked, then camera shows the right side of the ship. I've tested the game with not rendering the ship when the

camera position is changed, but because of the distance between the ship and asteroid, it is way too easy to collide with asteroid. Therefore, I had to draw the ship even though camera position is changed. It is simple not to render the ship, but this will distract the flow of the game. You will see what I mean once you play with 'q', 'e', and 'c' keys, so please understand this. (Please check line 13, 130 – 185, 238 – 307 in main.cpp)

Reference

1. asteroid1.jpg, asteroid2.jpg, asteroid3.jpg, asteroid4.jpg files are given by lecturer.
< https://rmit.instructure.com/courses/90696/files/23127517?module_item_id=3927483 >
2. back.png, down.png, front.png, left.png, right.png, up.png files are given by lecturer.
< https://rmit.instructure.com/courses/90696/files/23127517?module_item_id=3927483 >
3. ship.mtl, ship.obj, ship.png files are given by lecturer.
< https://rmit.instructure.com/courses/90696/files/23127517?module_item_id=3927483 >
4. bullet.jpg is free image from Freelmages website.
< <https://www.freeimages.com/photo/fire-1172366> >
5. missile.jpg is free image from Freelmages website.
< <https://www.freeimages.com/photo/danger-radioactive-1-1244879> >
6. explosion.png is free image from pngwing website
< <https://www.pngwing.com/en/free-png-dkhyj/download> >
7. propeller.png is free image from Freelmages website.
< <https://www.freeimages.com/photo/propeller-1420525> >
8. bullet.obj is free obj file from open3dmodel website.
< <https://open3dmodel.com/ko/download.php> >