Spacecraft Game

Tung Duong
Tung.Duong002@umb.edu
University of Massachusetts Boston

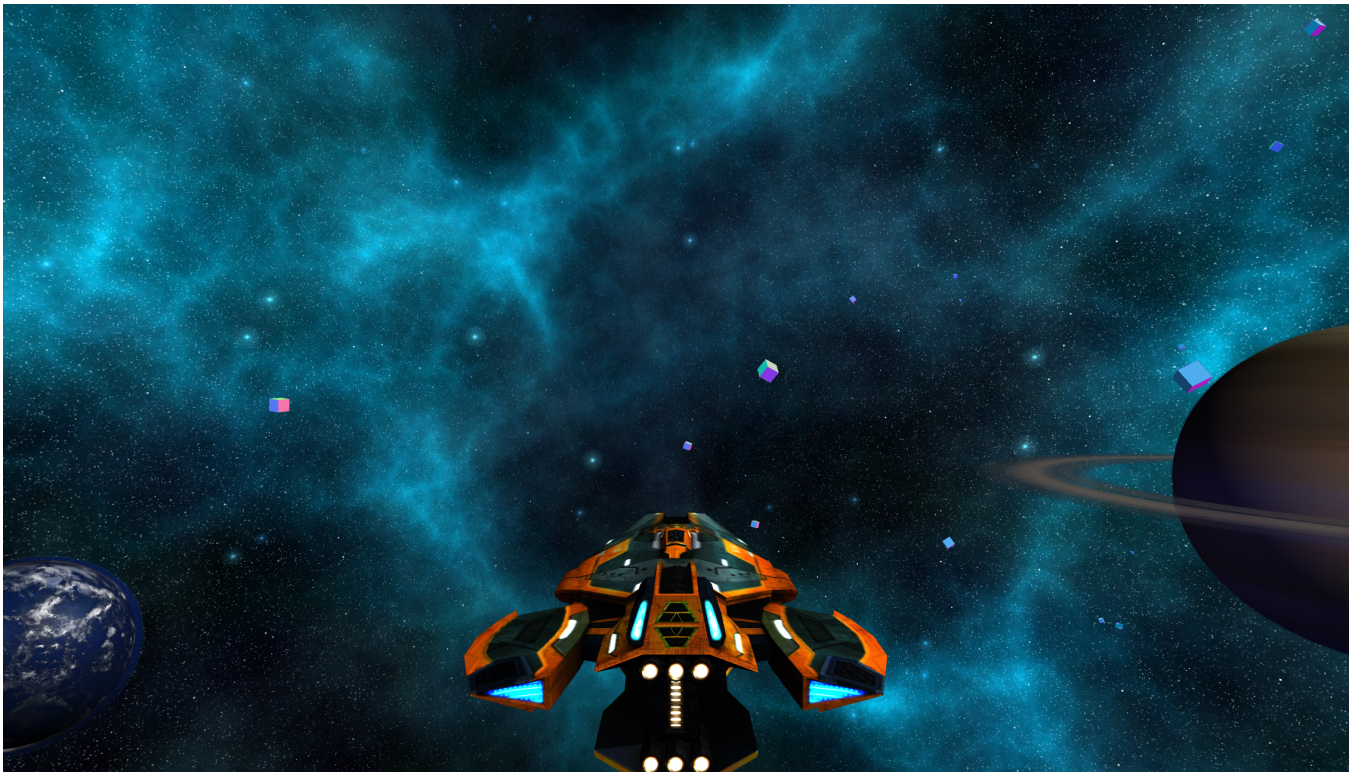


Figure 1: Spacecraft 3D Game Demo.

ABSTRACT

This should be a 1-paragraph summary of your project. Please replace this text and the image teaser and caption.

KEYWORDS

WebGL, Visualization

ACM Reference Format:
Tung Duong. 2021. Spacecraft Game. In *CS460: Computer Graphics at UMass Boston, Fall 2021.* Boston, MA, USA, 3 pages. https://CS460.org

CS460, Fall 2021, Boston, MA

ACM ISBN 1337.
https://CS460.org

1 INTRODUCTION

Spacecraft is a 3D game written by WebGL, Three.js, with many feature such as moving in space smoothly, fire, speed up, collision, and much more.

2 RELATED WORK

Here you can cite existing related work like XTK [2] or Three.js [1].

3 METHOD

- First, I created the three.js environment for showing any object which in my code.
- Second, I tested many style of using camera and control objects
- Next, I added lights, resize windows, loaded the glTF files and modified animation.
- Then I was using the slerp, quaternion, tween, etc. to modify object and camera moving.

3.1 Implementation

Some codes in my project:

```
//************************************************
// modify the laze bullets by using TWEEN
var bullet1 = new THREE.Mesh( geometry, material );
var position = new THREE.Vector3();
this._model.children[1].getWorldPosition(position);
bullet1.position.copy(position);
this._game._graphics.Scene.add(bullet1);
var distance = 1000000000000000000;
var targetPosition = new THREE.Vector3(
                    this._direction.x*distance,
                    this._direction.y*distance,
                    this._direction.z*distance );
var tween1 = new TWEEN.Tween( bullet1.position )
                .to( targetPosition, distance );
tween1.start();

//************************************************
// Check collisions by using Box3
var bbox_spacecraft = new THREE.Box3().setFromObject(
      player_spacecraft['player']._model.children[0]);
Collisions(bbox_spacecraft) {
  var centerBox = new THREE.Vector3();
  explosion1.position.copy(bbox_spacecraft.
                        getCenter(centerBox));
  explosion1.position.y -= 2000;
  // explosion1.position.z -= 400;
  player_spacecraft.player._game.
                _graphics.Scene.add(explosion1);
  explosion2.position.copy(bbox_spacecraft.
                           getCenter(centerBox));
  explosion2.position.y -= 600;
  // explosion2.position.z -= 400;
  player_spacecraft.player._game.
                    _graphics.Scene.add(explosion2);
 player_spacecraft.player._game._graphics.Scene.remove(
    player_spacecraft['player']._model);
  gameover_check = true;
}

//************************************************
// Control object by using quaternion
frameDecceleration.multiplyScalar(speedScale);
velocity.add(frameDecceleration);
const controlObject = this._params.target;
const quaternion = new THREE.Quaternion();
const angle = new THREE.Vector3();
const quaternionAction = controlObject.
                    _model.quaternion.clone();
```

3.2 Milestones

3.2.1 Milestone 1. My idea is to create a spaceship game where I will initially create a skybox, then I will process all the objects into it with a suitable scale.

3.2.2 Milestone 2. I divide the work into classes to handle them object-oriented. I ran into some problems dealing with how to properly adjust the camera as I moved the spacecraft.

3.2.3 Milestone 3. Dealing with rendering time properly is also quite confusing, I had to read many documents at Three.js homepage and some forums about them like Stackoverflow or Discover-threejs, etc. Besides, processing the mixers (animation) for the glTF files took me quite a while but it was worth it.

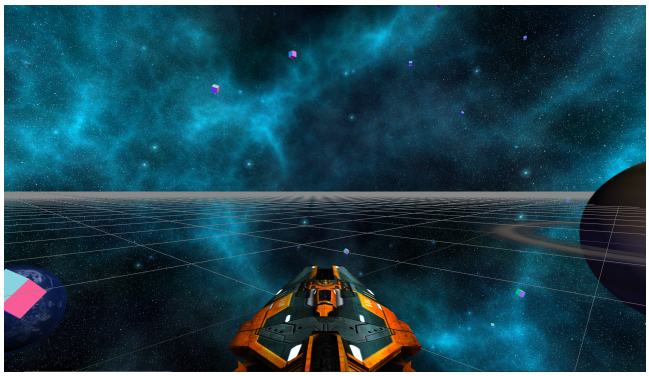
3.3 Challenges

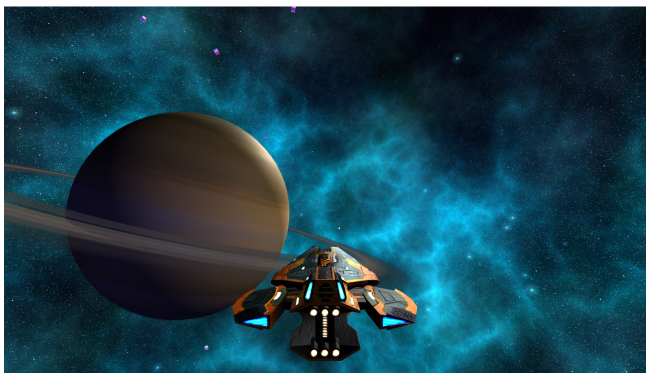
Describe the challenges you faced.

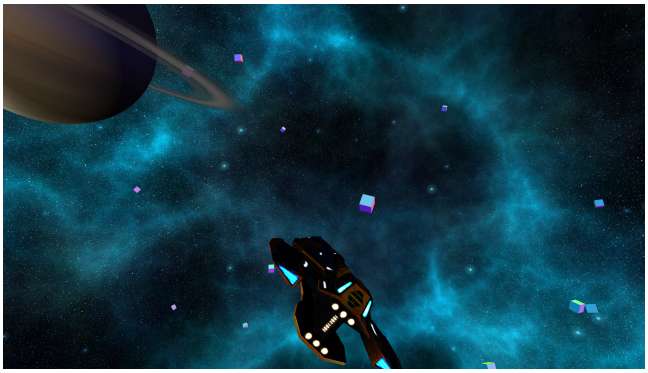
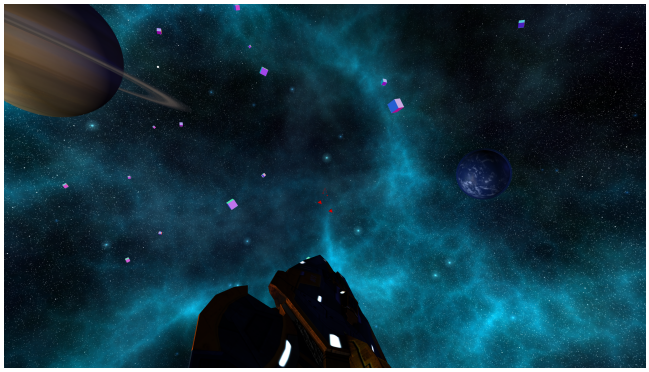
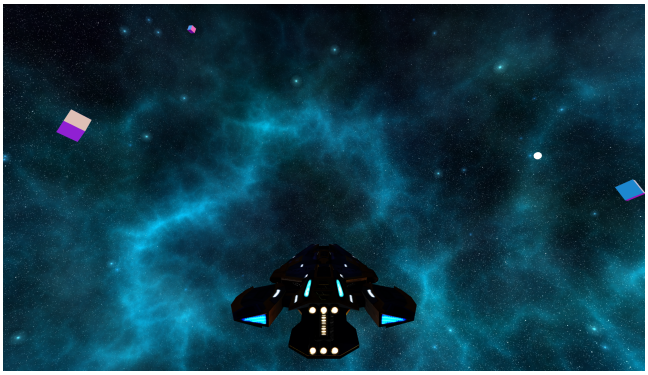
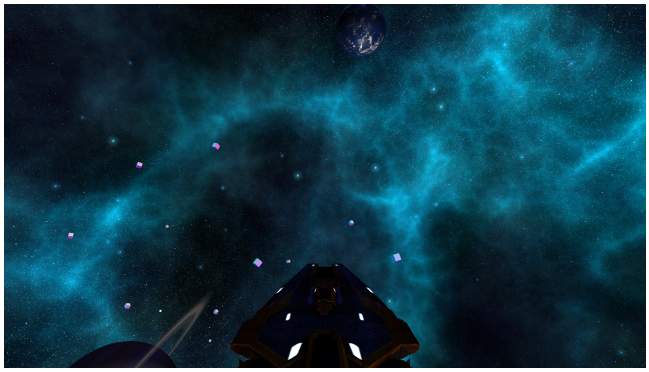
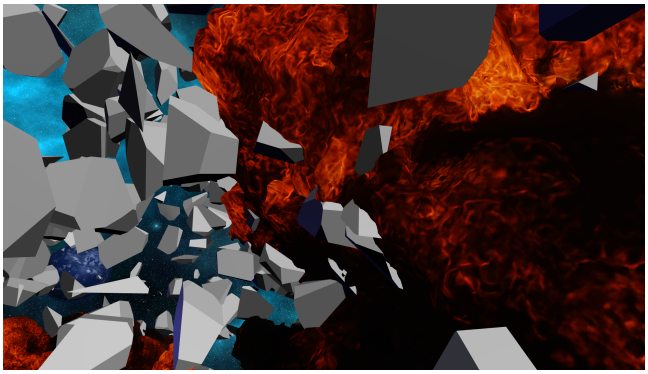
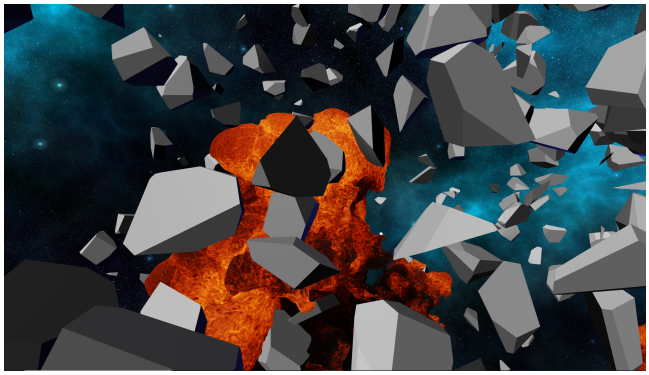
- Challenge 1: General
- Challenge 2: Camera control and Object controller
- Challenge 3: Interaction with other objects
- Challenge 4: Raycaster (or using TWEEN)

4 RESULTS

The demo of Spacecraft game with spacecraft: rocket, laze, moving, and meteors, planet, earth with explosion effect, etc.





5 CONCLUSIONS

Creating this game makes me more interested in Three.js in particular and Graphics in general. More than that, it helps me practice what I've learned during this semester from CS 460. Besides, doing a project helps me to discover the things I'm not good at, as well as improve my skills and raise my level with the advanced knowledges. And of course, I'm happy to make this game. In the future, I will develop this project more if given the opportunity and practice my knowledge of Graphics to help me better prepare for my future career.

Table 1: Performance table

Device	Performance
PC	60 FPS
Macbook	60 FPS

REFERENCES

[1] Ricardo Cabello et al. 2010. Three.js. *URL: https://github. com/mrdoob/three.js* (2010).

[2] Daniel Haehn, Nicolas Rannou, Banu Ahtam, P. Ellen Grant, and Rudolph Pienaar. 2012. Neuroimaging in the Browser using the X Toolkit. *Frontiers in Neuroinformatics* (2012).