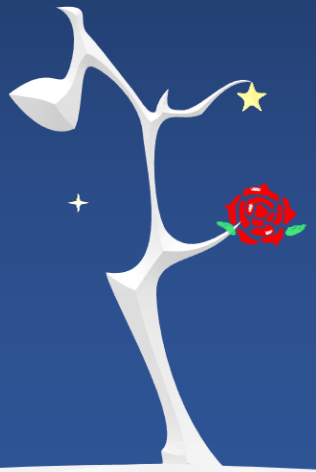
Team E

Little Princess



Index

Game BackGround

The rose boy and the little princess were happily living in B612.

But one day, the rose boy disappears leaving a diary.

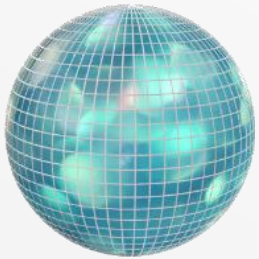
Why did the rose boy leave?

Let's cross the universe to meet the rose boy again!

Bitwise Planet
Binary arithmetic game

Gradation Planet
Find a different Color game

B612 Planet
Initial position



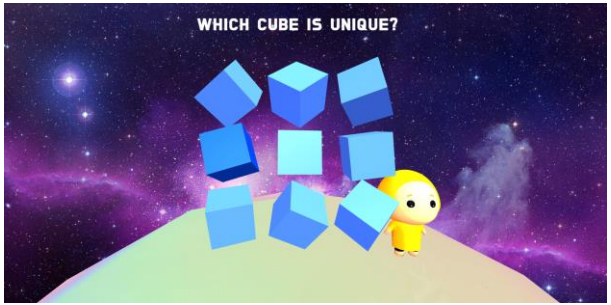
Mirrorball Planet
Keyboard game

Key Features

The 3D model was placed on a 2D plane to give it a unique look

Add stories throughout the map to create interest

Key Features



Make mini games that fit each planet

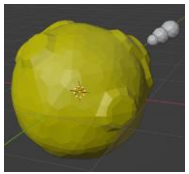
When the character approaches the planet, it moves to the mini-game page

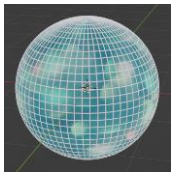
Implementation Detail

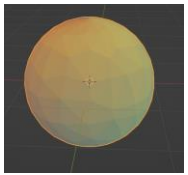
blender

Create own 3D models to match the mood of the game using

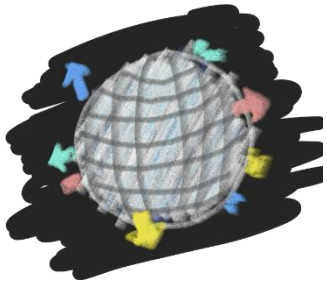
blender

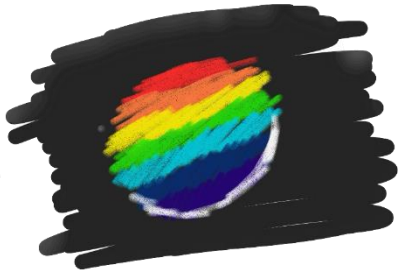






Implementation Detail

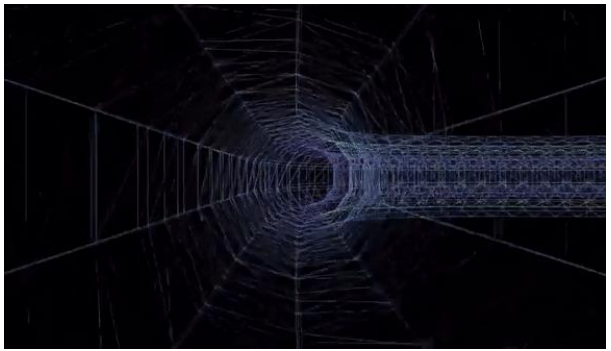


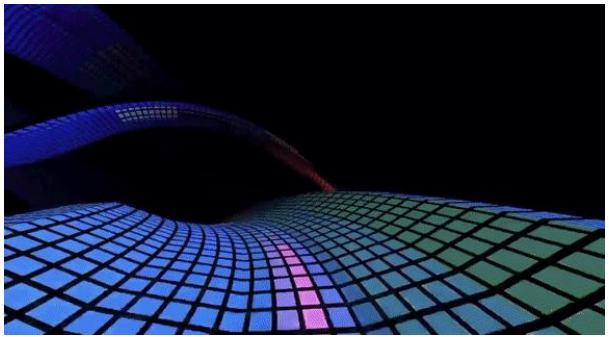


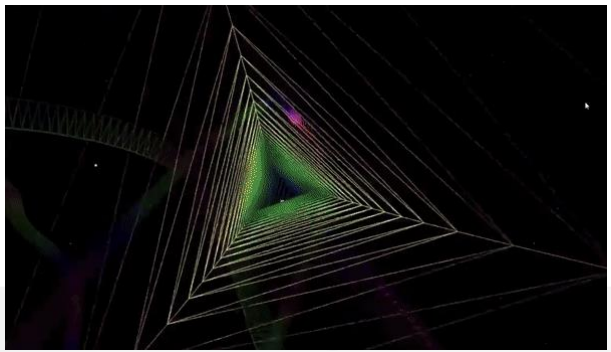
For improving the design, create and add elements to the game's initial map

Implementation Detail

Added a tunnel loading page to match the characteristics of the planet

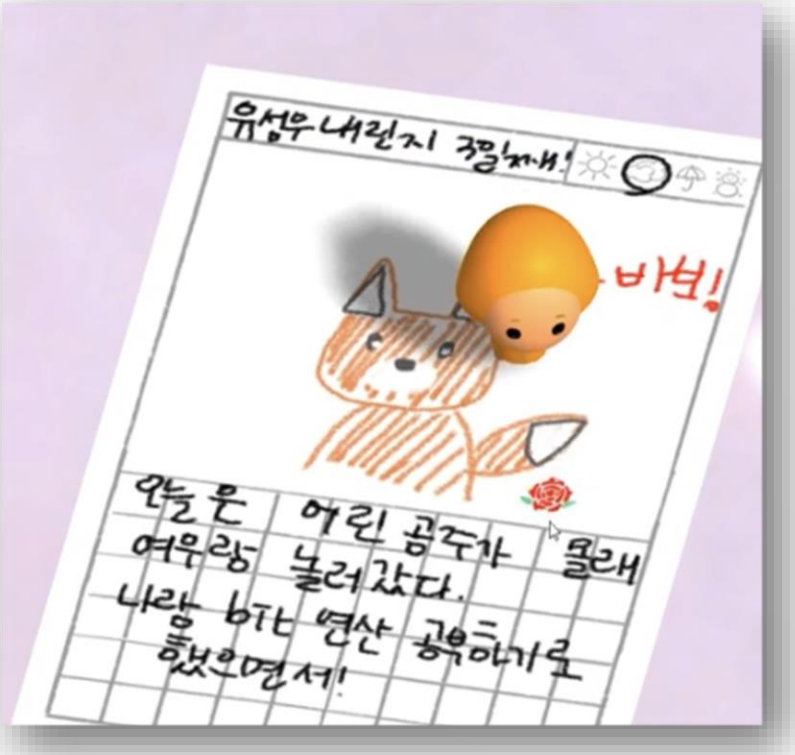






Implementation Detail

Main Map




```
//raycaster로 이동 구현
const raycaster = new THREE.Raycaster();
let mouse = new THREE.Vector2();
let destination = new THREE.Vector3();
let angle = 0;
let isPressed = false;
```

```
// Camera
const camera = new THREE.OrthographicCamera(
    -(window.innerWidth / window.innerHeight), // left
    window.innerWidth / window.innerHeight, // right,
    1, // top
    -1, // bottom
    -1000,
    1000
);
```

```
//gradation Diary Mesh
const graDiaryMesh = new THREE.Mesh(
    new THREE.PlaneGeometry(6, 10),
    new THREE.MeshStandardMaterial({
        map: graDiaryTexture
        ,transparent: true, opacity: 1.0, color: 'ffffff'
    })
);
graDiaryMesh.name = 'gradationDiary';
graDiaryMesh.position.set(30, 0.005, 10)
graDiaryMesh.rotation.x = -Math.PI/2;
graDiaryMesh.receiveShadow = true;
scene.add(graDiaryMesh);
```

Implementation Detail

Main Map


```
if (
    Math.abs(bitMesh.position.x - princess.modelMesh.position.x) < 1.5 &&
    Math.abs(bitMesh.position.z - princess.modelMesh.position.z) < 1.5
) {

    window.location.href = "http://127.0.0.1:5500/src/bitwise/tunnel_bit.html";
    if (!bit.visible) {
        bit.visible = true;
        gsap.to(
            bit.modelMesh.position,
            {
                duration: 1,
                y: 1,
                ease: 'Bounce.easeOut'
            }
        );
        gsap.to(
            camera.position,
            {
                duration: 1,
                y: 3
            }
        );
    }
```

Implementation Detail

In planet - bitwise game logic


```
function makeBinaryNum(){

  // make binary random number (4bit)
    var i = 4;
    while (i > 0){
        const num = Math.round(Math.random());
        num_2 = num_2 + num.toString();
        i--;
    }
    num_2_2 = num_2;
}

function changeBinaryToDecimal(){
    // change decimal number
    if(num_2 == "0000") {
        num_10= parseInt(0);
    }
    else {
        num_10 = (parseInt(num_2,2));
    }
}
```

Implementation Detail

In planet - keyboard game logic


```
//quiz 랜덤 출제
for(var i = 0; i < 6; i++ ){
  j = Math.floor(Math.random() * 4);
  console.log(j);
  quiz.push(keyArray[j]);
  quizArrow.push(arrow[j]);
  console.log(quiz);
}
```

```
//방향키 누르면 값 받아서 answer array에 저장
//6번 누르게 될경우 정답과 비교하는 함수 호출
$(document).keydown(function(e) {
  if (e.which==37) {
    $('.left').addClass('pressed');
    $('.left').css('transform', 'translate(0, 2px)');
    if(answer.length == 5) { answer.push(37); quizResult(); }
    else{ answer.push(37); console.log(answer);}
  } else if (e.which==38) {
    $('.up').addClass('pressed');
    $('.left').css('transform', 'translate(0, 2px)');
    $('.down').css('transform', 'translate(0, 2px)');
    $('.right').css('transform', 'translate(0, 2px)');
    if(answer.length == 5) { answer.push(38); quizResult();}
    else{ answer.push(38); console.log(answer); }
  } else if (e.which==39) {
    $('.right').addClass('pressed');
    $('.right').css('transform', 'translate(0, 2px)');
    if(answer.length == 5) { answer.push(39); quizResult(); }
    else{ answer.push(39); console.log(answer); }
  } else if (e.which==40) {
    $('.down').addClass('pressed');
    $('.down').css('transform', 'translate(0, 2px)');
    if(answer.length == 5) { answer.push(40); quizResult();} else{ answer.push(40); console.log(answer);}
  }
});
```

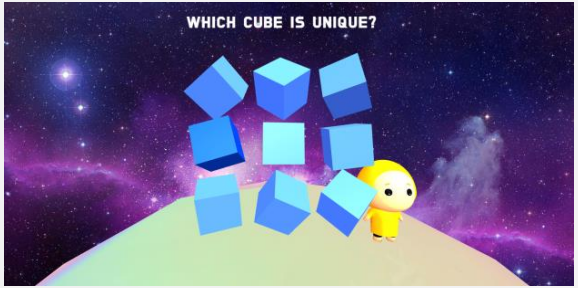
Implementation Detail

In planet - color game logic


```
randomColor();

const random_array = {rand_color, rand_color_1};
console.log(random_array);

answer_cube = Math.floor(Math.random()*8);
console.log(answer_cube);

const x = [-1.0, 0.0, 1.0, -1.0, 0.0, 1.0, -1.0, 0.0, 1.0];
const y = [0.9, 0.9, 0.9, 0.0, 0.0, 0.0, -0.9, -0.9, -0.9,];
const cubes = [];

for(var i=0; i<9; i++){
  if(i==answer_cube){
    cubes.push(makeInstance(geometry, rand_color_1, x[i], y[i]));
  }
  else{
    cubes.push(makeInstance(geometry, rand_color, x[i], y[i]));
  }
  console.log(cubes.length);

}

function randomColor() {
  var r = Math.random();
  var g = Math.random();
  var b = Math.random();
  r1 = r-0.2;
  g1 = g-0.2;
  b1 = b-0.2;
  rand_color = new THREE.Color(r, g, b);
  rand_color_1 = new THREE.Color(r1, g1, b1);

};
```

Implementation Detail

In planet



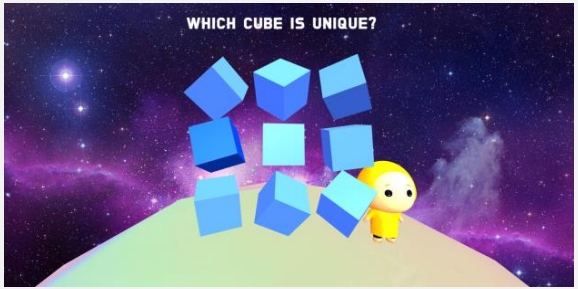

```
<audio autoplay loop>
  <source src="bitwise.mp3" type="audio/mp3">
</audio>
```

```
camera = new THREE.PerspectiveCamera(25, window.innerWidth / window.innerHeight, 1, 1000);
camera.position.z = 400;
camera.position.x = 0;
camera.position.y = 100;
scene.add(camera);
```

```
const loader = new THREE.GLTFLoader();
loader.load('../models/littlePrincess_2.glb', function(glb){
  princess = glb.scene.children[0];
  princess.scale.set(15,15,15);
 princess.position.x = -60;
 princess.position.y = 88;
 princess.position.z = 170;

 mixer = new THREE.AnimationMixer(glb.scene);

 var action = mixer.clipAction(glb.animations[0]);
 action.play();
```

Implementation Detail

In planet




```
function createMaterial(){
  var bitTexture = THREE.ImageUtils.loadTexture("green.jpg");
  var bitMaterial = new THREE.MeshBasicMaterial();
  bitMaterial.map = bitTexture;

  return bitMaterial;
}
```

```
function createMaterial(){
  var discoTexture = THREE.ImageUtils.loadTexture("disco.jpeg");
  var discoMaterial = new THREE.MeshBasicMaterial({
    // color: 0xBD9779,
    // shading: THREE.FlatShading
});
  discoMaterial.map = discoTexture;

  return discoMaterial;
}
```

```
function createMaterial(){
  var rainbowTexture = THREE.ImageUtils.loadTexture("rainbow1.jpg");
  var rainbowMaterial = new THREE.MeshBasicMaterial({
    // color: 0xBD9779,
    // shading: THREE.FlatShading
});
  rainbowMaterial.map = rainbowTexture;

  return rainbowMaterial;
}
```

Implementation Detail

In planet

```
function animateStars() {

  // loop through each star
  for(var i=0; i<stars.length; i++) {

  star = stars[i];

  // and move it forward dependent on the mouseY position.
  star.position.x -=  i/30;

  // if the particle is too close move it to the back
  if(star.position.x<-400) star.position.x+=800;
  }
}
```

Team Members

Seohyun Lee

4myf12u@gmail.com
201935097

- Keyboard Game
- Planet Design
- Planet Details

Sun Choi

sun4194@gachon.ac.kr
201935142

- Color Game
- Planet Design
- Loading page

Gaeun Kim

dkrk333@gachon.ac.kr
201935010

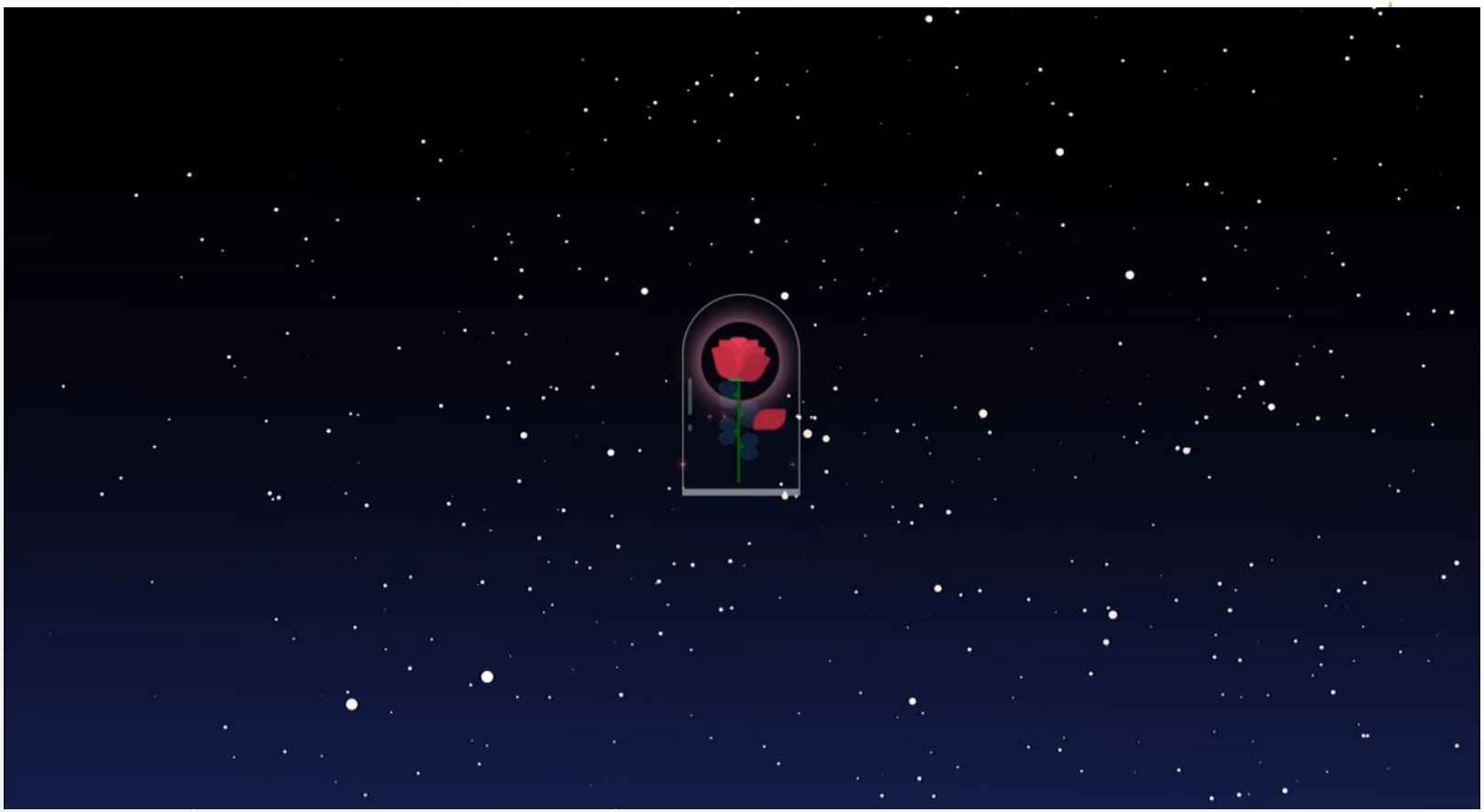
- Main Map Design
- Blender 3D modeling

Chaeyoung Oh

oco6029@naver.com
201935080

- Bitwise Game
- Planet Design
- Tunnel Page

Demo



THANK YOU

Little princess