const canvas = document.getElementById("gameCanvas");

const ctx = canvas.getContext("2d");

let bullets = []; //총알 저장

let meteo = []; // 메테오 저장

let curve\_bullets = [];

let isRotating = false; // 회전 상태 저장

let rotationAngle = 0; // 회전 각도

let selectedStone = null; // 선택된 스톤 저장

let stoneGlow = 0;

let glowing = true;

// 타노스 이미지

const ThanosImage = new Image();

ThanosImage.src = "images/Thanos.png";

// 건틀렛 이미지

const gauntletImage = new Image();

gauntletImage.src = "images/gauntlet.png";

//운석 이미지

const meteoImage = new Image();

meteoImage.src = "images/meteo.png";

// 인피니티 스톤 효과 이미지

const stoneImages = {

mind: new Image(),

reality: new Image(),

soul: new Image(),

time: new Image(),

power: new Image(),

};

stoneImages.mind.src = "images/mind.png";

stoneImages.reality.src = "images/reality.png";

stoneImages.soul.src = "images/soul.png";

stoneImages.time.src = "images/time.png";

stoneImages.power.src = "images/power.png";

// 인피니티 스톤 크기

const stoneSize = 100;

// 건틀렛 크기

const gauntletWidth = 595 \* 0.2;

const gauntletHeight = 1119 \* 0.2;

// 타노스 크기

const ThanosWidth = 571 \* 0.7;

const ThanosHeight = 389 \* 0.7;

let ThanosX = canvas.width / 2 - ThanosWidth / 2; // 중앙 정렬

//기본 총알 클래스

class Bullet{

constructor(x, y, dx, dy) {

this.x = x;

this.y = y;

//이동좌표

this.dx = dx;

this.dy = dy;

this.size = 8;

}

move() {

this.x += this.dx;

this.y += this.dy;

}

draw() {

ctx.fillStyle = "pink";

ctx.beginPath();

ctx.arc(this.x, this.y-10, this.size, 0, Math.PI \* 2);

ctx.arc(this.x-60, this.y-10, this.size, 0, Math.PI \* 2);

ctx.arc(this.x+60, this.y-10, this.size, 0, Math.PI \* 2);

ctx.fill();

}

soulDraw(){

ctx.fillStyle = "skyblue";

ctx.beginPath();

ctx.arc(this.x, this.y-10, this.size, 0, Math.PI \* 2);

ctx.arc(this.x-60, this.y-10, this.size, 0, Math.PI \* 2);

ctx.arc(this.x+60, this.y-10, this.size, 0, Math.PI \* 2);

ctx.fill();

}

}

class curveBullet {

constructor(startX, startY, controlX, controlY, endX, endY) {

this.startX = startX;

this.startY = startY;

this.controlX = controlX;

this.controlY = controlY;

this.endX = endX;

this.endY = endY;

this.t = 0; // 진행도 (0 ~ 1)

}

move() {

this.t += 0.02; // 이동 속도

if (this.t > 1) this.t = 1;

this.x = (1 - this.t) \* (1 - this.t) \* this.startX +

2 \* (1 - this.t) \* this.t \* this.controlX +

this.t \* this.t \* this.endX;

this.y = (1 - this.t) \* (1 - this.t) \* this.startY +

2 \* (1 - this.t) \* this.t \* this.controlY +

this.t \* this.t \* this.endY;

}

draw() {

ctx.fillStyle = "green";

ctx.beginPath();

ctx.arc(this.x, this.y, 5, 0, Math.PI \* 2);

ctx.fill();

}

}

//운석 클래스

class Meteorite {

constructor(x, y) {

this.x = x;

this.y = y;

this.size = Math.random() \* 30 + 30; // 크기 랜덤

this.dy = Math.random() \* 3 + 1; // 속도 랜덤

this.exploded = false; // 폭발 여부 체크

this.explosionTime = 0; // 폭발 후 유지 시간

}

move() {

if (!this.exploded) {

this.y += this.dy;

}

}

draw() {

if (!this.exploded) {

ctx.fillStyle = "brown";

ctx.beginPath();

ctx.arc(this.x, this.y, this.size, 0, Math.PI \* 2);

ctx.fill();

} else {

ctx.fillStyle = "orange";

ctx.beginPath();

ctx.arc(this.x, this.y, this.size \* 1.5, 0, Math.PI \* 2);

ctx.fill();

this.explosionTime++;

if (this.explosionTime > 20) {

this.exploded = false;

return true;

}

}

return false;

}

checkCollision() {

if (this.y + this.size >= canvas.height) {

this.exploded = true;

return true;

}

return false;

}

}

// 보스 클래스

class Boss {

constructor(x, y) {

this.x = x; // 보스 X 좌표

this.y = y; // 보스 Y 좌표

this.hp = 10000; // 보스 체력

this.phase = 1; // 보스 페이즈

this.attackPattern = ["mind", "soul", "time", "power", "reality"]; // 인피니티 스톤 패턴

this.width = ThanosWidth;

this.height = ThanosHeight;

this.movePattern = Math.floor(Math.random() \* 3) + 1;

this.shootCooldown = 0;

this.meteorCooldown = 0;

}

// 보스 이동

// 보스 이동 패턴 1 (좌우 이동)

move1() {

this.x += Math.sin(Date.now() / 1000) \* 1;

// console.log(this.x);

}

// 보스 이동 패턴 2 (위아래 이동)

move2() {

this.y += Math.sin(Date.now() / 500) \* 1;

// console.log(this.y);

}

// 보스 이동 패턴 3 (대각선 이동)

move3() {

this.x += Math.sin(Date.now() / 500) \* 1;

this.y += Math.cos(Date.now() / 500) \* 1;

// console.log(this.x);

// console.log(this.y);

}

// 기본 총알

shoot(){

if (this.shootCooldown <= 0) {

bullets.push(new Bullet(this.x + this.width / 2, this.y + this.height, 0, 1));

this.shootCooldown = 100;

} else {

this.shootCooldown--;

}

}

shootcurveBullets() {

if (this.shootCooldown <= 0) {

for (let i = 0; i < 3; i++) {

let controlX = this.x + this.width / 2

let controlY = this.y + this.height

let endX = this.x + (Math.random() \* 200 - 100);

let endY = this.y + 300;

curve\_bullets.push(new curveBullet(this.x, this.y, controlX, controlY, endX, endY));

}

this.shootCooldown = 100; // 일정 시간 후 다시 발사

} else {

this.shootCooldown--;

}

}

powerShot() {

if (this.meteorCooldown <= 0) {

for (let i = 0; i < 5; i++) {

let meteorX = Math.random() \* canvas.width;

meteo.push(new Meteorite(meteorX, -50));

}

this.meteorCooldown = 1000;

} else {

this.meteorCooldown--;

}

}

// 랜덤 스톤 선택

chooseRandomStone() {

const randomIndex = Math.floor(Math.random() \* this.attackPattern.length);

selectedStone = this.attackPattern[randomIndex]; // 스톤 고정

console.log(`선택된 스톤: ${selectedStone}`);

return selectedStone;

}

//보스 패턴들

bossPattern(){

if (selectedStone === "power"){

console.log(`나올 패턴: ${selectedStone}`)

}

else if (selectedStone === "soul"){

console.log(`나올 패턴: ${selectedStone}`)

ctx.fillStyle = "blue";

}

else if (selectedStone === "mind"){

console.log(`나올 패턴: ${selectedStone}`)

ctx.fillStyle = "yellow";

}

else if (selectedStone === "reality"){

console.log(`나올 패턴: ${selectedStone}`)

ctx.fillStyle = "red";

}

else if (selectedStone === "time"){

console.log(`나올 패턴: ${selectedStone}`)

ctx.fillStyle = "green";

}

}

//---------------------------------------

// 보스 & 건틀렛 그리기

draw() {

ctx.save(); // 현재 캔버스 상태 저장

// 건틀렛 회전

let gauntletX = this.x + 240;

let gauntletY = this.y - 20;

ctx.translate(gauntletX + gauntletWidth / 2, gauntletY + gauntletHeight / 2);

ctx.rotate(rotationAngle \* Math.PI / 180); // 각도를 라디안 값으로 변환

ctx.translate(-gauntletX - gauntletWidth / 2, -gauntletY - gauntletHeight / 2);

ctx.drawImage(gauntletImage, gauntletX, gauntletY, gauntletWidth, gauntletHeight);

ctx.restore(); // 캔버스 상태 복원

//타노스 그리기

ctx.drawImage(ThanosImage, this.x, this.y, this.width, this.height);

//선택된 스톤을 건틀렛 위에 표시

if (selectedStone) {1

ctx.globalAlpha = stoneGlow;

if (selectedStone === "mind"){

ctx.drawImage(stoneImages[selectedStone],gauntletX+1,gauntletY+30,stoneSize,stoneSize);

}

else if (selectedStone === "power"){

ctx.drawImage(stoneImages[selectedStone],gauntletX+30,gauntletY+5,stoneSize,stoneSize);

}

else if (selectedStone === "reality"){

ctx.drawImage(stoneImages[selectedStone],gauntletX-8,gauntletY+5,stoneSize,stoneSize);

}

else if (selectedStone === "soul"){

ctx.drawImage(stoneImages[selectedStone],gauntletX+12,gauntletY+5,stoneSize,stoneSize);

}

else if (selectedStone === "time"){

ctx.drawImage(stoneImages[selectedStone],gauntletX+60,gauntletY+55,stoneSize,stoneSize);

}

ctx.globalAlpha = 1;

}

}

// 보스 상태 업데이트

update() {

setTimeout(() => {

this.movePattern = 1;

console.log(this.movePattern);

}, 5000);

setTimeout(() => {

this.movePattern = 2;

console.log(this.movePattern);

}, 10000);

setTimeout(() => {

this.movePattern = 3;

console.log(this.movePattern);

}, 15000);

if (this.movePattern === 1) {

this.move1();

} else if (this.movePattern === 2) {

this.move2();

} else {

this.move3();

}

this.shoot();

this.shootcurveBullets();

if (selectedStone === "power") {

this.powerShot();

}

}

}

// 보스 객체 생성

let boss = new Boss(canvas.width / 2 - ThanosWidth / 2, 50);

let mete = new Meteorite(canvas.width / 2 - ThanosWidth / 2, 50);

// 임시 이벤트 (회전 및 스톤 선택)

window.addEventListener("keydown", (e) => {

if (e.code === "Space") {

isRotating = true;

selectedStone = null;

}

});

window.addEventListener("keyup", (e) => {

if (e.code === "Space") {

isRotating = false;

rotationAngle = 0;

boss.chooseRandomStone();

boss.bossPattern();

}

});

function glow(){

if (glowing){

stoneGlow += 0.05;

if (stoneGlow >= 3){

glowing = false;

}

}

else{

stoneGlow -= 0.05;

if (stoneGlow <=0.05){

glowing = true;

}

}

}

// 🎮 메인 게임 루프 (운석 효과 추가)

function gameLoop() {

ctx.clearRect(0, 0, canvas.width, canvas.height);

boss.update();

boss.draw();

if (selectedStone === "power") {

bullets.forEach((bullet) => {

bullet.move();

bullet.draw();

});

meteo.forEach((mete, index) => {

mete.move();

if (mete.checkCollision()) {

meteo.splice(index, 1);

} else {

if (mete.draw()) {

meteo.splice(index, 1);

}

}

});

}

else if (selectedStone === "soul"){

bullets.forEach((bullet) => {

bullet.move();

bullet.soulDraw();

});

}

else {

bullets.forEach((bullet) => {

bullet.move();

bullet.draw();

});

}

if (isRotating) {

rotationAngle += 10;

}

glow();

requestAnimationFrame(gameLoop);

}

// 게임 시작

gameLoop();