HTML

1. canvas : 기본 화면 틀 구성

```
<canvas id="gameCanvas" width="2000" height="900"></canvas>
```

2. audio : bgm 구현

```
<audio id="intro-bgm" src="sound/bgm/base.mp3" autoplay loop></audio>
```

3. div, button, img : 각종 Manual 구현

```
<div id="battleRuleDialog" class="infoDialog">
  <div><img src="picture/manual_battle.png" /></div>
  <button class="closeInfo">Close</button>
</div>

<div id="targetRuleDialog" class="infoDialog">
  <div><img src="picture/manual_target.png" /></div>
  <button class="closeInfo">Close</button>
</div>
```

=> HTML을 이용하여 전체적인 게임 틀을 구성하였다.

CSS

1. font-face : 기본 폰트 설정

```css
@font-face {
  font-family: "MyFont";
  src: url("./cruiser_fortress/cruiserfortress.ttf");
}
```

2. 각종 기능 : 게임 기본 틀 및 dialog의 위치, 색상 등을 설정

```css
body,
html {
  margin: 0;
  padding: 0;
  height: 100%;
  overflow: hidden;
  display: flex;
  justify-content: center;
  align-items: center;
  background-color: white;
}
#gameCanvas {
  border: 1px solid black;
  display: block;
}
```

```css
.infoDialog {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  background-color: rgba(0, 0, 0, 0.8);
  color: white;
  padding: 100px;
  border-radius: 10px;
  display: none;
  width: 60%;
  text-align: center;
}
.infoDialog button {
  margin-top: 20px;
  padding: 10px 20px;
  color: black;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
.infoDialog img {
  height: 600px;
  width: 1200px;
}
```

=> CSS를 이용하여 HTML 객체 각각을 세부 설정

JavaScript

1. 로드 : 폰트, 이미지, 효과음 등을 로드

```
// 폰트 로드 확인
document.fonts
  .load('40px "MyFont"')
  .then(function () {
    console.log("Font loaded");

    // 이미지 로드
    var background = new Image();
    background.src = "./picture/BackGround/intro.jpg";

    // 효과음 로드
    var clickSound = new Audio("./sound/click.mp3");
```

2. ctx : 각종 버튼을 디자인

```
// 버튼 스타일
function drawButton(x, y, width, height, radius, text, isPressed) {
  // 버튼 배경
  ctx.fillStyle = isPressed ? "#A0522D" : "#8B0000";
  drawRoundedRect(x, y, width, height, radius);
  ctx.fill();
  ctx.strokeStyle = "#000";
  ctx.lineWidth = 4;
  ctx.stroke();

  // 버튼 텍스트
  ctx.fillStyle = "#EEE8AA";
  ctx.font = "35px 'MyFont'";
  ctx.textAlign = "center";
  ctx.textBaseline = "middle";
  ctx.fillText(text, x + width / 2, y + height / 2);
}
```

3. addEventListener : 클릭, 키보드 사용 등 각종 이벤트 생성

```
// 클릭 이벤트
canvas.addEventListener("mousedown", function (event) {
  var x = event.pageX - canvas.offsetLeft;
  var y = event.pageY - canvas.offsetTop;
```

4. location : 페이지 이동

```
window.location.href = "select_game.html";
```

5. getElementById : 동적으로 HTML과 통신

```
document.getElementById("skills-console").style.display = "none";
```

6. 배열 : 상황에 따라 다르게 나타날 수 있도록

```
var tankHoverImages = [
  [
    "./picture/Skill/attack/BigBomb.jpg",
    "./picture/Skill/defense/IncreaseHP.jpg",
    "./picture/Skill/attack/StrongBomb.jpg",
  ],
  [
    "./picture/Skill/attack/IncreaseTank.jpg",
    "./picture/Skill/attack/TwiceChance.jpg",
    "./picture/Skill/defense/IncreaseMovement.jpg",
  ],
  [
    "./picture/Skill/attack/BigBomb.jpg",
    "./picture/Skill/attack/IncreaseTank.jpg",
    "./picture/Skill/attack/TwiceChance.jpg",
  ],
  [
    "./picture/Skill/defense/IncreaseHP.jpg",
    "./picture/Skill/defense/DecreaseTank.jpg",
    "./picture/Skill/defense/SmallBomb.jpg",
  ],
];
```

7. localStorage : 다음 html 파일에 데이터를 넘겨주는 역할

```
if (y > 500 && y < 580) {
  localStorage.setItem("user1", selectedTank1);
  localStorage.setItem("user2", selectedTank2);
  playClickSound();
  window.location.href = "game_battle.html";
```

7. Math : 포탄 각도, 포탄 파워게이지 등 구현

```
else if (key === " " && isCharging) {
isCharging = false;
isFired = true;
missilePower = gauge * 1.6;
missileDx = missilePower * Math.cos(cannon1Angle);
missileDy = missilePower * Math.sin(cannon1Angle);
gauge = Math.PI; // 게이지 초기화
Tank1Shoot.currentTime = 0;
Tank1Shoot.play(); // 플레이어 1 발사 소리 재생
```

8. 중력가속도 : 중력에 의해 y축 방향의 속도가 점점 감소

```
const GRAVITY_ACCELERATION = 0.098;
```

```
missileDy -= GRAVITY_ACCELERATION;
```

=> JavaScript를 이용하여 현재 HTML과 다음 HTML과의 통신, 사용자와의 상호작용에서 적절한 대응을 할 수 있도록 구축하였다. 그리고 수학적, 물리적 함수를 사용하여 포탄의 움직임, 파워 등을 구현