Space War

Space War is a arcade style browser game where player has to destroy all the falling enemies in order to survive and gain score.

Player has 3 lives at start of the game, it decreases each time a enemy's ship pass through the player's ship. The player's lives will become 0 if any of the enemy ship collides with it. When there are no lives left with the player, game is over.

Play game: https://gauravano.github.io/space-war/
Project repository: https://github.com/gauravano/space-war

Scoring:

Player will get the score for killing every enemy ship. Destroying a Enemy 1 ship will add 10 points and destroying Enemy 2 ship will add 20 points to the player's score.

Enemy

There are two types of enemy ships in the game:

Enemy 1

- Destroying it, will add 10 points to the player score.
- Initial velocity is 2 frames/sec

- With each new level, the enemy velocity will increase by 1

Enemy 2

- Destroying it, will add 20 points to the player score.
- Initial velocity is 3 frames/sec
- With each new level, the enemy velocity will increase by 1

Levels

There are infinite level in this game. Player will start with Level 1 and level count will increase, each time player score become greater than multiple of 300. With each new level, enemy's ship velocity will increase by 1, making it difficult for player to kill it and survive.

Implementation Approach

I started the project using the provided demo code. I gone through all the instructions and implemented all the steps. After that, I had a running game with me but that looked very basic.

I completed the addition implementations section after that, score count addition, gameover function implementation, new enemy created and levels created. So, at this point, I have all functionality ready but still game looks very basic(and bore).

So, I did some more work on styling, images, sounds, etc which can be seen here - https://github.com/gauravano/space-war/commits/master

The final game can be played at https://gauravano.github.io/space-war/ .

Below are the details of some important functions(from index.js) involved in the gameplay:

1. Start function

```
function start() {
    canvas.font = "800 80px Verdana";
    var gradient = canvas.createLinearGradient(0, 0,
canvasElement.width, 0);
    gradient.addColorStop("0", " red");
    gradient.addColorStop("0.5", "yellow");
    gradient.addColorStop("1.0", "red");
    canvas.fillStyle = gradient;
    canvas.fillText("SPACE WAR", 60, 100);

    var playButton = `<button id="playButton"
onclick="reset()"></button> `
    $("#maindiv").append(playButton);

    var image = new Image();
    image.src = "./images/space_bar.png";
    image.onload = function () {
        canvas.drawImage(image, 40, 378, 220, 48);
    }

    var image2 = new Image();
    image2.src = "./images/arrow_keys.png";
    image2.onload = function () {
        canvas.drawImage(image2, 320, 330, 200, 100);
    }

    canvas.font = "800 25px Verdana";
    canvas.fillStyle = "yellow";
    canvas.fillText("Press SPACE to shoot", 30, 458);
    canvas.fillText("Press LEFT & RIGHT", 300, 458);
    canvas.fillText("arrow keys to move", 320, 480);
```

```
}
```

It is called after the resource loading is done. This function shows the welcome screen of the game, showing name of the game, playing instructions with images, and Start game button using which player can start the game.

2. Reset Game

```
function reset() {
    Sound.reset();
    Sound.play("kick_shock");
    if (document.getElementById("restartButton") != undefined) {
        var elem = document.getElementById("restartButton");
        elem.parentNode.removeChild(elem);
    }

    if (document.getElementById("playButton") != undefined) {
        var elem = document.getElementById("playButton");
        elem.parentNode.removeChild(elem);
    }
    level = 1;
    playerBullets = [];
    enemies = [];
    explosions = [];
    enemy2counter = 5;
    player.lives = 3;
    player.score = 0;
    partition = 300;
    enemyVelocity = 2;
    player.x = 300;
    player.y = 440;

    interval = setInterval(function () {
        draw();
        update();
```

```
    }, 1000 / FPS);
}
```

Reset function re-initializes the value of the variables and start the game i.e., falling of enemies, shooting, etc will begin after this function is called. The background song is also played after entering in this function. It also contains the infinite loop which updates and renders the canvas everytime. This function is called when user presses start button on entry page or when user presses reset button after game over.

3. Game Over function

```
function gameOver() {
    var image = new Image();
    image.src = "./images/gameover.png";
    image.onload = function () {
        canvas.drawImage(image, 100, 100, 400, 140);
    }
    Sound.play("GameOver");

    canvas.font = "800 34px Verdana";
    canvas.fillStyle = "yellow";
    canvas.fillText("YOUR SCORE: " + player.score, 145, 300);

    var restartButton = `<button id="restartButton"
onclick="reset()"></button> `
    $("#maindiv").append(restartButton);
    clearInterval(interval);
}
```

This function is called when player lose all its lives. This function stops the game, shows the reset button and final score of the player. By pressing reset button, player can start a new game.

4. Render entities

```
function renderEntities(list) {
    for (var i = 0; i < list.length; i++) {
        renderEntity(list[i]);
    }
}

function renderEntity(entity) {
    canvas.save();
    canvas.translate(entity.pos[0], entity.pos[1]);
    entity.sprite.render(canvas);
    canvas.restore();
}
```

Render entities function receives explosions array as parameter and calls render entity on each object(explosion) present in the array. A explosion is shown by running a sprite image so render entity function is called for each explosion which then shows the explosion as a animation on the canvas using functions of sprites.js

Code Organization

```
 ├─ css
     └─ style.css
  ├─ images
     ├─ arrow_keys.png
     ├─ background.png
     ├─ bullet_enemy.png
     ├─ enemy1.png
     ├─ enemy2.png
     ├─ favicon-96x96.png
     ├─ gameover.png
     ├─ player.png
     ├─ restartButton.png
     ├─ space_bar.png
```

```
    ├─ sprites.png
    └─ startButton.png
  ├─ scripts
    ├─ index.js
    ├─ jquery.hotkeys.js
    ├─ jquery.min.js
    ├─ key_status.js
    ├─ resources.js
    ├─ sound.js
    ├─ sprite.js
    ├─ sprite2.js
    └─ util.js
  ├─ sounds
    └─ explosion.mp3
    ├─ GameOver.mp3
    ├─ kick_shock.mp3
    ├─ main.mp3
    └─ shoot.mp3
  ├─ index.html
  └─ README.md
```

CSS

css directory contains all the stylesheets used in the application

style.css is the stylesheet containing all the styling used in the application. It is imported in index.html file.

Images

images directory contains all the images used in the application at all the places.

arrow_keys.png is the image of arrow keys which appear at start of the game and direct user to use "left" and "right" arrow key for moving the player left and right

background.png is the image of the background which is used in application. Background remain same throughout the application.

bullet_enemy.png is the image of the bullet fired by the player's ship.

enemy1.png is the image of the Enemy 1 ship

enemy2.png is the image of the Enemy 2 ship

favicon-96x96.png is the image used as favicon for the HTML page.

gameover.png is the image that appear on canvas when game is over for player.

player.png is the image of the player's ship.

restartButton.png is the image of the reset button which appear on canvas after the game is over, placed there for restarting the new game.

space_bar.png is the image showing spacebar key. It appears at the start of the game, instructing player to press "space" for firing bullet.

sprites.png is the image sprite containing different images of the game. I am using collision images from this image sprite.

startButton.png is the image of Start button which appears at the start of the game, placed for starting the game.

Scripts

scripts directory contains all the JS scripts used in the application. All the scripts contained in this folder are imported in index.html

index.js is the main JS file which contains all the the main gameplay functions

jquery.hotkeys.js contains jQuery Hotkeys plugin which makes key handling in the game much easier

jquery.min.js is the minified version of the JQuery

key_status.js contains a 16-line JS wrapper that makes event querying available. Using it, one can query the status of a key at any time by checking keydown.left, keydown.right, etc.

resources.js contains various JS functions which perform image loading prior to game start. It is used for specifically loading the sprites.png

sound.js contains functions which are used for playing sound, pausing sound and for performing various functions with the sound in the game.

sprite.js is a simple utility class using which loading and drawing images on canvas is done. Except collision animation, all the other images use this file.

sprite2.js is a utility class which perform loading and drawing of the image sprite on the canvas. It is used for sprites.png which is used for showing explosion animation.

util.js contains clamp function which keeps the player in the bound of canvas.

Sounds

explosion.mp3 is the explosion sound played when player or enemy ship is destroyed.

GameOver.mp3 is the sound played when game is over

kick_shock.mp3 is the background sound played when player start playing game.

main.mp3 is the alternative song for the game

shoot.mp3 is the sound of the bullet

index.html

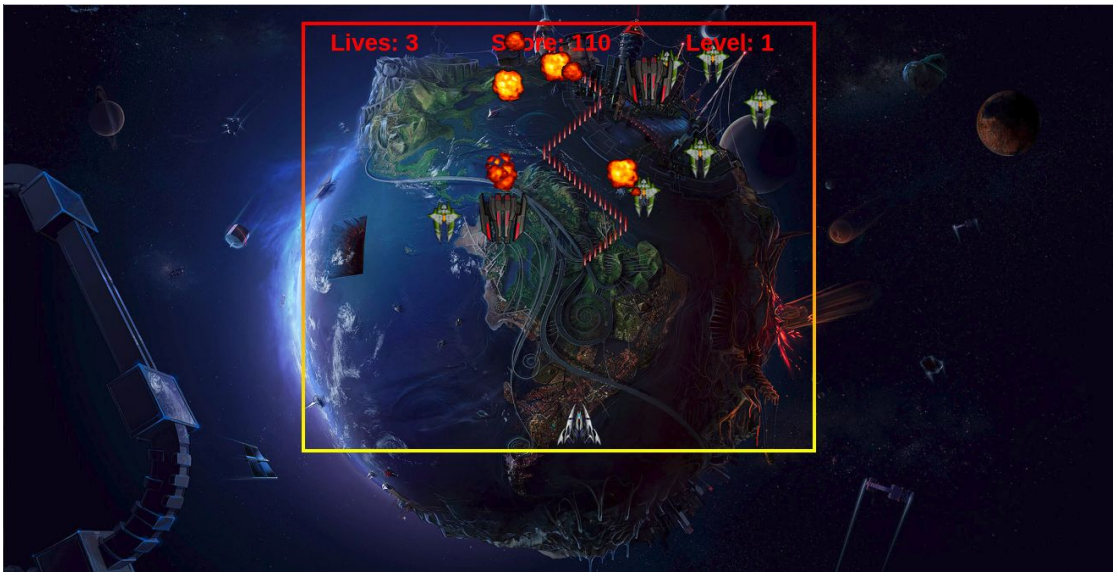
Contains the HTML of the game. It contains the canvas object.

README.md

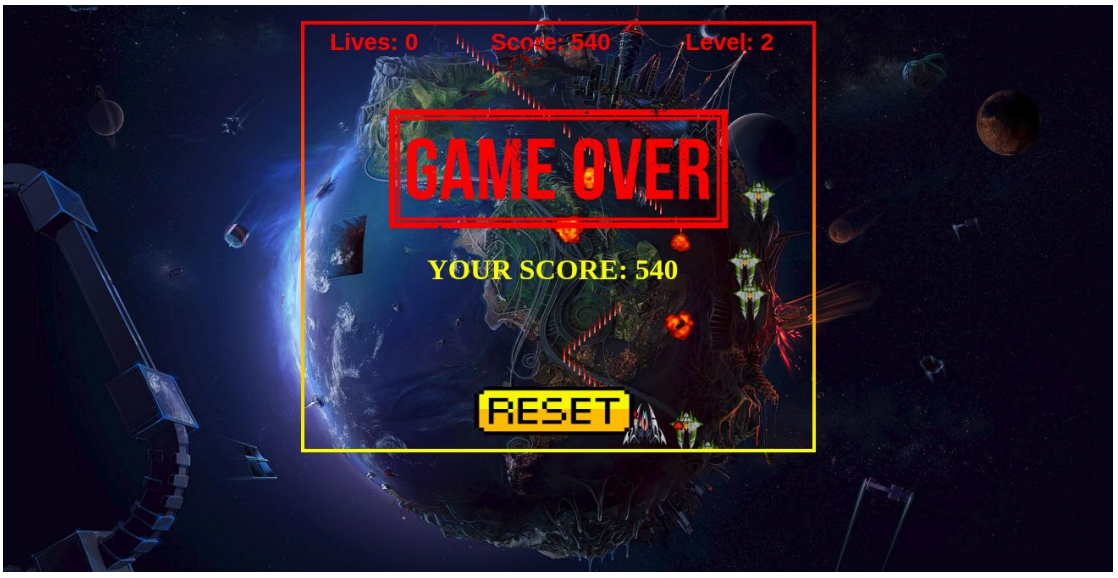
Contains documentation of the game in the markdown format

Screenshots

Game Start view



Game view



Game Over view