

**Fundamentals of WEB Technologies**

**Task 6 mid-term**

Student name: Shaltay Rakhymzhan

Group name: CSS\_2001

**Almaty, 2024**

**Documentation for "Space Game" Project**

**Overview**

"Space Game" is a dynamic arcade game where the player controls a spaceship, shoots enemies, avoids collisions, and aims to achieve the highest possible score. The game becomes progressively more challenging with each level, making it engaging for players. Scores are saved, allowing players to track their personal records.

**Project Files**

1. **index.html** — The base structure of the game, containing the canvas for gameplay and control elements.
2. **styles.css** — Defines the interface styling, including the menu, background, and visual effects.
3. **game.js** — Contains the main game logic written in JavaScript.
4. **space-background.mp4** — A background video that creates an atmospheric space theme.

**HTML Structure**

**Key Elements:**

* **<canvas id="gameCanvas">** — The canvas where the gameplay is rendered.
* **<div id="menu">** — The start menu with a button to begin the game.
* **<p id="highScore">** — A block to display the highest score.
* **<button id="startButton">** — Button to start the game.
* **<video id="gameVideo">** — Animated background video element.

**Connections:**

* The game logic is implemented in the game.js file and linked via the <script> tag.
* Styles are applied from styles.css, while the background is implemented with space-background.mp4.

**Styling (styles.css)**

1. **Background:**
   * The #gameVideo element is used as an animated background.
   * It is set to fill the entire screen with object-fit: cover.
2. **Central Menu:**
   * The #menuContent element is positioned at the center of the screen.
   * The menu includes a title, high score display, and a start button.
3. **Game Canvas:**
   * The canvas element is hidden until the game starts, ensuring users are not distracted during the menu phase.

**Game Logic (game.js)**

**Variables:**

* player — Object representing the player, including coordinates, dimensions, and speed.
* bullets — Array of bullets fired by the player.
* enemies — Array of enemies appearing on the screen.
* score — The player's current score.
* lives — The player's remaining lives.
* difficulty — Variable for increasing the game's difficulty.
* bestScore — The highest score, saved in LocalStorage.

**Core Functions:**

1. **updateHighScore()**
   * Compares the current score with the highest score.
   * Updates bestScore in LocalStorage if the current score is higher.
2. **createEnemy()**
   * Generates an enemy with random position and speed.
   * Adds the enemy to the enemies array.
3. **drawPlayer()**
   * Draws the player's spaceship on the canvas.
4. **drawBullets()**
   * Displays bullets on the screen and manages their movement.
5. **drawEnemies()**
   * Renders enemies and controls their movement.
6. **handleCollisions()**
   * Checks for collisions between bullets and enemies.
   * Decreases the player's lives if enemies reach the bottom of the screen.
7. **updatePlayer()**
   * Implements player controls for left and right movement using arrow keys.
8. **shoot()**
   * Adds a new bullet to the bullets array.
9. **gameLoop()**
   * The main game loop that updates and redraws all elements on the screen.

**Events:**

* **keydown** — Handles player movement and shooting.
* **keyup** — Stops player movement.
* **click on startButton** — Starts the game.

**Game Start:**

* Resets all counters (score, lives, difficulty).
* Switches from the menu to the game screen (canvas).
* Launches the main game loop gameLoop().

**Game Over:**

* Displays the message "Game Over" with the current score.
* Saves the highest score.

**User Guide**

**Starting the Game:**

1. Open the project in a web browser.
2. On the start screen, click the "Start Game" button.

**Controls:**

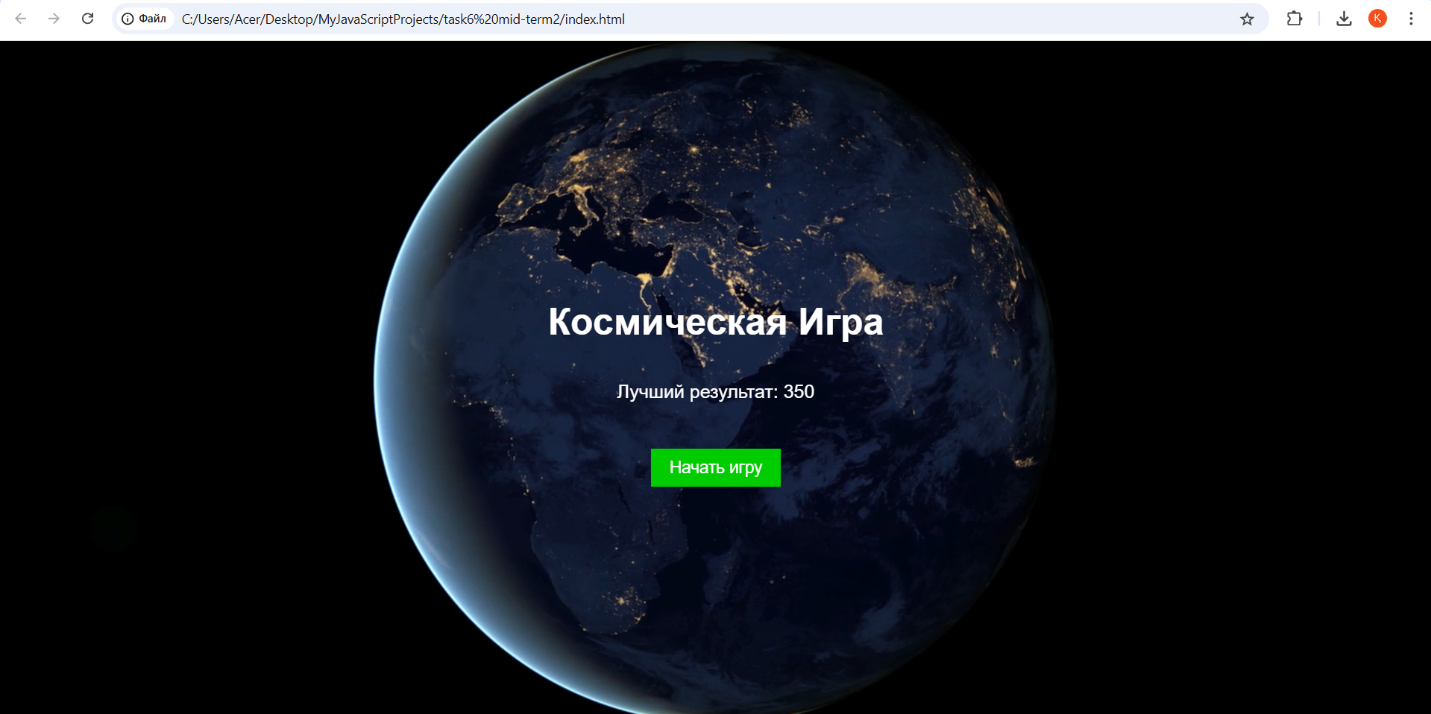
* Use **left and right arrow keys** to move the spaceship.
* Press the **spacebar** to shoot bullets.

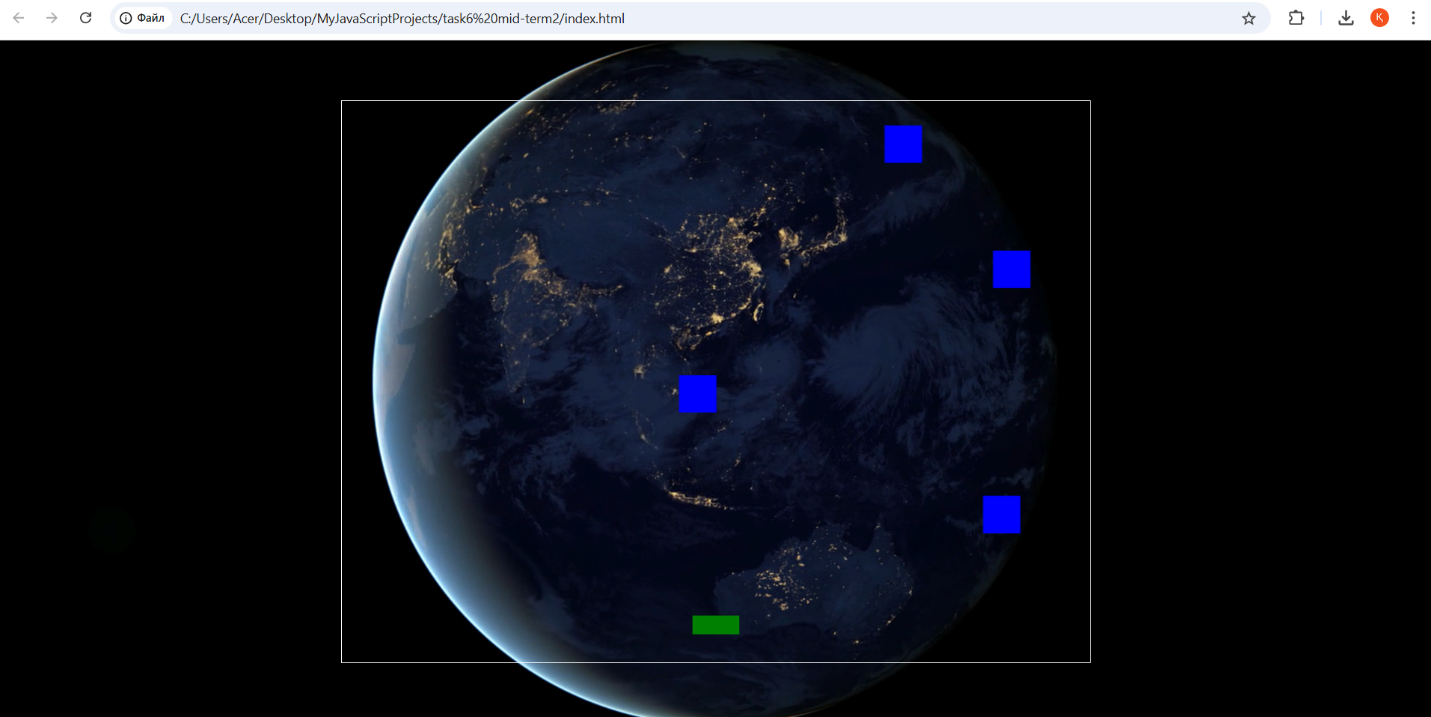
**Objective:**

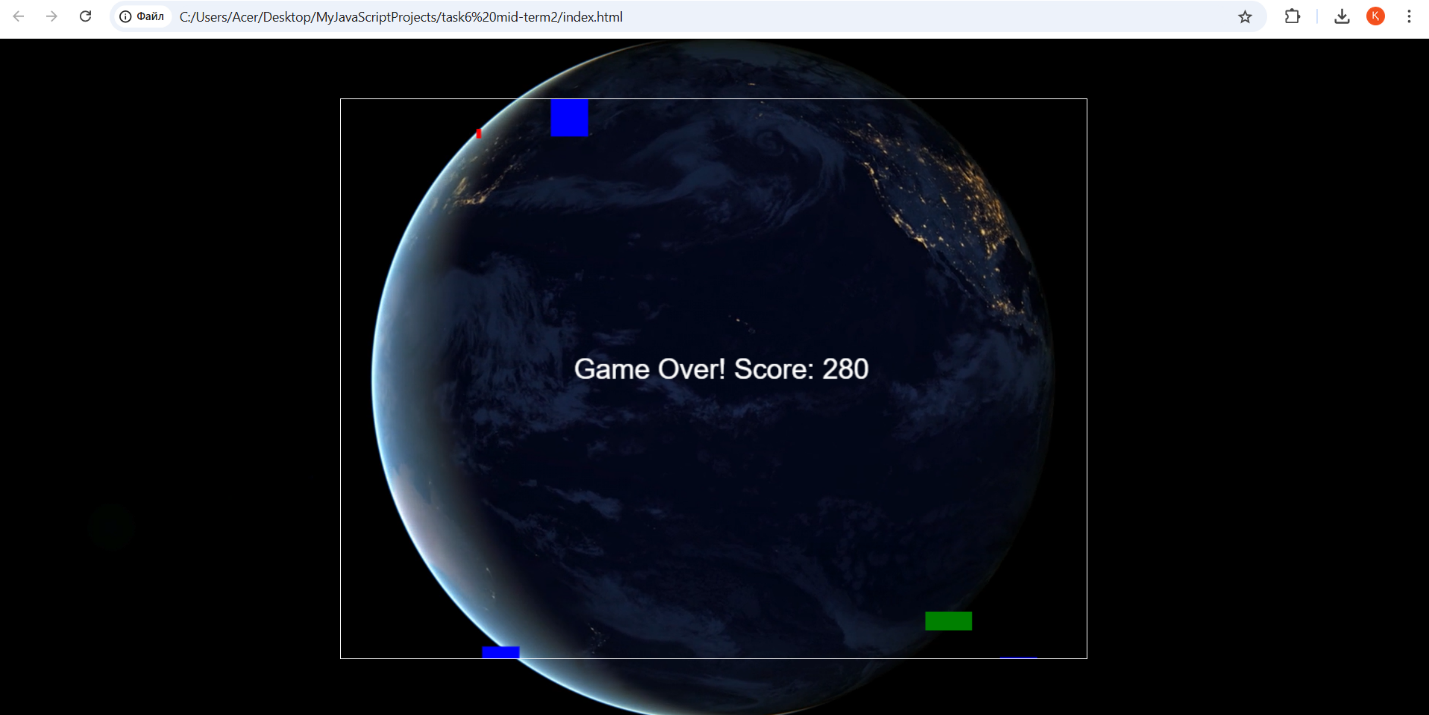
* Destroy enemies to earn points.
* Avoid collisions with enemies.
* Try to beat your previous highest score.

**Indicators:**

* **Score:** Displayed during gameplay.
* **Lives:** Decrease when enemies are missed.
* **High Score:** Automatically saved for future sessions.







In conclusion, the "Space Game" project is an engaging and dynamic arcade application that offers users a unique gaming experience. The game combines simple controls, increasing difficulty, and a score-saving system, making it both enjoyable and motivating to beat personal records. The project showcases fundamental web game development skills and has potential for expansion with additional features to enhance its interactivity and user engagement.

https://github.com/RakhymzhanShaltay