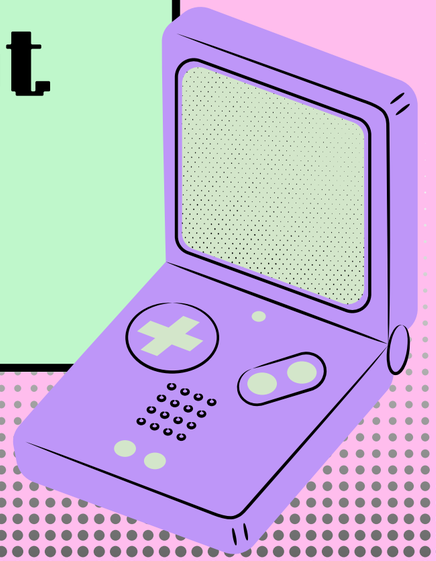

SPACE SHOOTER
GAME PROJECT
START

HELLO!

Let's get to know each other better by playing these games!

Title: Space Shooter Game

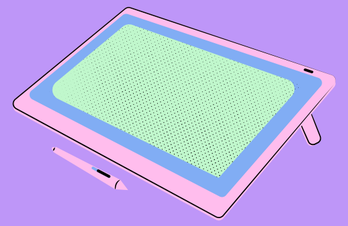
Subtitle: A Browser-Based Game Developed with HTML, CSS, and JavaScript



PROJECT OVERVIEW

- **Description:** Space Shooter is an interactive, browser-based game where players control a character to shoot at incoming asteroids.
- **Technologies Used:** HTML, CSS, JavaScript
- **Objective:** Shoot as many asteroids as possible to achieve a high score while avoiding collisions.

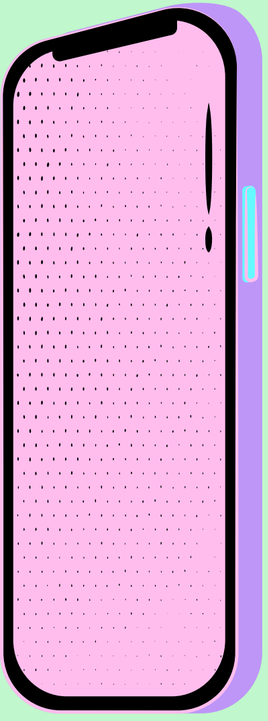
KEY FEATURES



- **Player Character:** Animated character controlled with arrow keys, shooting bullets to destroy asteroids.
- **Asteroid Mechanics:** Asteroids spawn at the top and move downward.
- **Score Tracking:** Points are awarded for each destroyed asteroid.
- **Lives and Game Over:** Player loses a life when an asteroid reaches the bottom.
- **Victory Condition:** The game ends in victory when a target score is achieved.
- **Responsive Design:** Game works on desktop and mobile browsers.

GAME MECHANICS

- **Movement:** Player uses arrow keys to move left/right.
- **Shooting:** Space bar to shoot bullets.
- **Collision Detection:** Detects when a bullet hits an asteroid.
- **Score System:** Live score counter that updates in real-time.
- **Game Over:** Triggered when the player runs out of lives.



PROJECT WORKFLOW

1. **Initial Setup:** Create index.html, styles.css, and game.js files.
2. **Character Creation:** Design character using CSS animations.
3. **Asteroid Logic:** Implement asteroid generation and movement.
4. **Game Loop:** Develop a loop to continuously update game states.
5. **Interaction Handling:** Implement controls for movement and shooting.
6. **Collision Detection and Score:** Code logic for collision detection and score update.
7. **Game End and Victory:** Define conditions for victory and loss.

CODE HIGHLIGHTS

Player Movement:

```
document.addEventListener('keydown', (e) => {
if (e.key === 'ArrowRight'){ /* Move right */ }
if (e.key === 'ArrowLeft'){ /* Move left */ }

});
```

Shooting Logic:

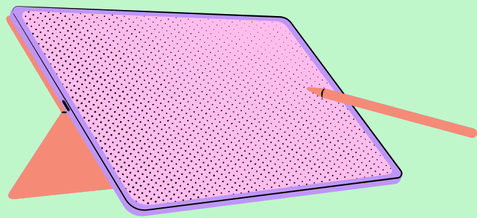
```
document.addEventListener('keydown', (e) => {
if (e.key === ' '){ /* Shoot bullet */ }
});
```

Collision Detection:

```
if (bulletHitsAsteroid){
score += 10;
destroyAsteroid();
}
```

CHALLENGES AND SOLUTIONS

- **Challenge:** Implementing smooth player movement.
  - **Solution:** Used event listeners for real-time updates.
- **Challenge:** Making the game responsive.
  - **Solution:** Used CSS media queries and flexible layouts.
- **Challenge:** Managing game state and collision logic.
  - **Solution:** Implemented a game loop with precise calculations.



CONCLUSION

SUMMARY: THE SPACE SHOOTER GAME COMBINES WEB TECHNOLOGIES TO CREATE AN ENGAGING AND INTERACTIVE EXPERIENCE.

LESSONS LEARNED: IMPROVED SKILLS IN GAME LOGIC, ANIMATION, AND RESPONSIVE DESIGN.

FUTURE GOALS: ADD MORE COMPLEX FEATURES LIKE MULTIPLAYER SUPPORT AND ADVANCED SCORING.


THANK YOU!
END