

## Feature Planning Document (Step 2)

Project: JavaScript Tetris

Forked from: <https://github.com/chvin/react-tetris>

### ➔ Feature 1: Pause and Resume Functionality

User Story: As a player, I want to pause and resume the game so I can take breaks without losing progress.

#### Technical Requirements

- Add a new piece of state (e.g., `isPaused`) to track whether the game is paused.
- Update the `useInterval` hook to skip movement when paused.
- Create a “Pause/Resume” button to toggle this state.
- Display a visual overlay or label indicating when the game is paused.

#### Files Likely to Be Modified

- `src/components/Tetris.js`: For button, pause state, and visual indication.
- `src/hooks/useInterval.js`: To halt automatic piece dropping when paused.

#### Potential Challenges

- Avoiding unintended resets when toggling pause.
- Ensuring timers and score updates are completely halted and resumed correctly.

### ➔ Feature 2: Dark Mode Toggle

User Story: As a user, I want to toggle a dark mode for more comfortable gameplay at night.

#### Technical Requirements

- Add a `darkMode` state to toggle styles.
- Apply conditional classes or CSS variables based on theme.
- Add a button to switch between light and dark modes.
- Create or update CSS styles to support both modes.

## Files Likely to Be Modified

- `src/styles/index.css`: Add dark mode styles.
- `src/components/Tetris.js`: Add `darkMode` toggle logic.
- `src/components/StartButton.js`: Theme-specific styling updates.

## Potential Challenges

- Managing global styles cleanly across components.
- Preventing visual glitches when toggling modes during gameplay.

## ➔ Feature 3: Local Score Leaderboard

User Story: As a player, I want to view and save my highest scores so I can track my progress over time.

## Technical Requirements

- On game over, compare score to local storage high scores.
- Store top 5 scores in `localStorage`.
- Create a new component to display the leaderboard.
- Display the leaderboard either on the game over screen or in a sidebar/modal.

## Files Likely to Be Modified

- `src/components/Tetris.js`: Save and load scores.
- Create a new file: `src/components/Leaderboard.js`
- `src/styles/index.css`: Style the leaderboard.

## Potential Challenges

- Managing `localStorage` read/write during rendering.
- Displaying scores cleanly in limited space.