**Space Ranger: Alien Invasion Game**

## Overview

In **"Space Ranger: Alien Invasion"**, players control a spaceship navigating a hazardous asteroid field and defending against alien invaders. The game is a console- based adventure where players aim to survive, destroy enemy ships, and collect resources to advance to higher levels.

## Features

#### Map Representation:

* + The game map is displayed as a **2D character array** (50x30).
  + Spaceship (<^>), asteroids (#), enemy ships (E), and stars (\*) are represented using ASCII characters.
  + The spaceship moves horizontally within the bottom 10 rows of the map.

#### Spaceship Abilities:

* + **Movement**: Controlled by arrow keys for left/right movement.
  + **Shooting**: Players press the spacebar to shoot lasers (|) upwards.
  + **Boost**: Pressing 'B' activates a short burst of speed, allowing the ship to dodge obstacles.

#### Obstacles:

* + **Asteroids**: Randomly generated obstacles that move downward. Colliding with an asteroid reduces the player's health.
  + **Enemy Ships**: Move horizontally, shooting bullets (-) towards the spaceship.
  + **Stars**: Collect stars to gain bonus points and power-ups.

#### Obstacle and Enemy Interaction:

##### Asteroid Collision:

* + - **Asteroids** will randomly spawn and move downward at a gradual speed. If the spaceship collides with an asteroid, the spaceship loses one life, and the game continues.
    - After each collision with an asteroid, the spaceship will momentarily freeze (e.g., for 1 second) to show the impact.

##### Enemy Ships:

* + - Enemy ships will be placed at random positions and start moving horizontally at different speeds.
    - When an enemy ship reaches the edge of the screen, it will reappear from the opposite edge.
    - If the player collides with an enemy ship, they lose a life.
    - **Enemy Ship Bullets**: Enemy ships shoot bullets downward every 5 seconds. The bullets disappear after traveling 10 rows.
    - **Bullet Collision**: If the player's spaceship collides with an enemy bullet, they lose one life.
    - **Player's Bullets**: Hitting an enemy ship with a bullet destroys the ship and grants a specific score (e.g., +50 points).

1. ***Game Speed and Progression:***
   * **Speed Road**:
     + In the center of the screen, there is a **Speed Road** (a row marked with =====) that temporarily increases the spaceship's speed. When the spaceship is on the speed road, its movement speed doubles.
     + Objects like asteroids and enemy ships can move over the speed road but are unaffected by it. Once the spaceship leaves the road, the speed returns to normal.

### Game Level Transition:

* The game has two levels:
  + **Level 1**: Normal speed and slower enemies.
  + **Level 2**: 1.5x speed for asteroids and enemies.
  + The game transitions to **Level 2** after either 10 enemy ships are destroyed or after covering a 5 km distance.

#### Crates and Collectibles:

##### Crate Destruction:

* + - **Crates** appear randomly and can be destroyed by the player's bullets. Each crate destroyed grants the player **1 life** after destroying five crates.
    - If the spaceship collides with a crate without destroying it, the player loses one life.

##### Collectibles:

* + - **Stars** appear randomly in the game map. The player earns **20 points** for every star collected.
    - Collecting stars will not just increase the score but can also provide temporary power-ups (like speed boost, extra shields, or invincibility for a short time).

#### Alien Bosses:

##### Boss Battles:

* + - After completing Level 2, a **boss alien ship** appears. The boss is larger, moves in erratic patterns, and shoots in all directions.
    - Defeating the boss will give the player **50 bonus points** and move them to the next level or end the game if the player chooses to exit.

#### High Score and Save System:

##### High Score Tracking:

* + - The top 10 high scores, along with the player's name, will be stored in a **text file**.
    - Whenever the game ends, it updates the leaderboard with the latest score if the player has made it into the top 10.
    - Players can view the leaderboard from the main menu.

##### Saving and Loading:

* + - Players can save their progress in the game (current level, score, and lives) and load it later to continue playing from where they left off.

1. ***Timers and Game Mechanics:***

### Countdown Timer:

* + - Each level has a **countdown timer** that starts when the level begins. The timer shows the remaining time to finish the level (e.g., 5 minutes for each level).
    - If time runs out, the game ends, and the player is shown their final score and rank.

### Distance Tracker:

* + - The spaceship moves upward in the game, simulating forward movement. Every step count as **2 meters**.
    - **5 km distance** (2500 steps) will trigger the **level-up** or unlock special events (like boss fights or power-ups).

### Bullet Movement:

* + - **Player's Bullets**: When the spacebar is pressed, the spaceship fires bullets upwards. Each bullet travels upward until it either collides with an obstacle, an enemy ship, or moves off-screen.
    - **Enemy Bullets**: Enemy ships fire bullets every 5 seconds, which move downward and disappear after 10 rows.

1. ***Extra Mechanics:***

### Jumping and Boosting:

* + - Pressing the 'J' key will allow the spaceship to jump over obstacles temporarily (if not colliding with other objects).
    - Boost ('B') temporarily doubles the spaceship's speed and allows it to dodge obstacles more effectively.

### Object Interaction:

* + - Objects like **enemy ships** and **asteroids** can collide and interact in complex ways, such as pushing each other or causing explosions when destroyed.

#### Menu System:

* + The game includes a menu with options:

1. **Play Game**: Start a new game and choose the difficulty level.
2. **Instructions**: Display controls and game play mechanics.
3. **High Scores**: Display top 10 scores and allow the player to view their ranking.
4. **Exit**: Quit the game.

### Programming Instructions for Space Ranger: Alien Invasion

***Main Guidelines***

### Modular Code Structure:

All logic must be encapsulated in functions; main() should only call these functions.

### Function Naming:

Use clear and descriptive names for functions, variables, and parameters (e.g., drawMap(), handleCollisions(), updateObjects()). Avoid overly short or vague names; prioritize readability.

### Code Reusability:

Encapsulate repetitive code into reusable functions. Example: Create a function like moveSpaceship() to handle all spaceship movements instead of duplicating logic.

### Function Responsibilities:

Assign a single responsibility to each function. Example: generateAsteroids() should handle only asteroid generation, while checkCollisions() should only verify and process collisions.

### Variable Names:

Use meaningful names to indicate purpose; reserve single-letter names for counters.

### Encapsulation:

Group related functionality within specific functions. Example: Encapsulate all high- score operations within saveHighScores() and loadHighScores()Also minimize the use of global variables; prefer passing data through parameters and returning results.

### Error Handling:

Create dedicated functions to handle errors with clear and user-friendly messages.

### Consistent Indentation and Formatting:

Follow a consistent style for indentation and code alignment.

### Commenting:

Add meaningful comments for complex logic, but prefer self-explanatory code.

### Peer Review:

Regularly review the code with peers to identify improvements and ensure quality.

### Code Documentation:

Use documentation practices to create structured explanations for functions and logic, detailing their purpose, parameters, and behavior.

### File handling

Data should be saved in the .txt file this the proper naming conventions