RQGames - Space Invaders Clone

Create a simplified *Space Invaders* game clone. The main idea here is to recreate basic game experience with use of advanced Unity features and top notch software design patterns.

Game reference:
https://www.youtube.com/watch?v=D1jZaIPeD5w

➔ Simplified game loop

- **Loading Screen**
  - Perform all data and asset initialization at this point
  - After everything is ready, proceed to Main Menu
- **Main Menu Screen** - should have two options to choose
  - Start Game - proceeds to Gameplay
  - High Scores - list of the top 10 scores
- **Gameplay Screen**
  - HUD - should consists of
    - Exit button
    - Current Wave
    - Current Score
    - Lives left
- **Results Screen**
  - Your score
  - Waves defeated
  - Button return to Main Menu
- **High Scores Screen**
  - List containing: score and date when it was achieved
  - Button return to Main Menu

➔ Gameplay requirements

- Player should have limited lives
- Enemies should come in waves indefinitely
- Player and enemies can shoot to each other
  - When player hits enemy it dies
  - When enemy hits player he loses one live

- The game ends when player loses all lives
- After player has been hit, he should become invulnerable for few seconds
- There should be a top down 3D camera
- We're providing ready 3d assets for the game
- We can simplify things and not do the obstacles in the game (player and enemies fight directly)

➔ Technical requirements

- Use latest Unity LTS version and choose Android platform
- Player progress should be stored locally on device
- Use Unity.Canvas for UI/HUD elements
- All gameplay related constants should be stored in form of a config file (Scriptable Object preferably)
- Use Unity.Addressables for asset management
    - (Optional) Organise your assets so that most of them could be downloaded separately in form of AssetBundle, leaving only what's entirely necessary in the build.
- Make use of a finite state machine (FSM) and dependency injection (DI). You can choose either any framework you like or write everything from scratch.
- (Optional) Make use of async/await syntax and UniTask (https://github.com/Cysharp/UniTask) when dealing with asynchronous stuff.

➔ Deliverable

- Project that we will be able to build ourselves or at least run in editor.