SideStep: Crazoonga Hours TDD

What is *SideStep: Crazoonga Hours*?

A 3D top-down action adventure game set on an unnamed, mystical beach. Play as “Crazoonga”, a crab with the intent of conquering the beach for the safety of his crustacean brethren. The game offers unique sidestepping movement and a simple combat system, where rotating and positioning yourself for combat is a must.

It would be aimed at fans of low-poly aesthetics, emulating the days of old such as the PS2-era, with a camera system reminiscent of the original three *Resident Evil* (Capcom) games.

Game Mechanics:

- Side-to-side movement (left and right arrow keys), to emulate that of a crab.
- Rotate Crazoonga’s body to affect where he moves (up and down arrow keys).
- Press Space to swing his claw and attack enemies.
- Collect shells for protection and locate matches to burn logs.

Game Visual Elements:

- 3D Models and animations for Crazoonga and the enemy types, as well as environmental details and structures.
- Cameras at unique, set angles to add to the retro game feel.
- Lighting and fog used with variety per stage to add a mystical ambience.

Level Design:

- Enemies in hidden locations that lie in wait for the player.
- Shells for the player to collect will be gathered in each stage.
- Mixture of regular enemies and occasional simple boss battles to teach about new enemy mechanics.

UI:

- Very minimal UI, in the sense that we only see basic text to inform the player of the controls, as well as UI to inform them of their success or their defeat.
- The end screen features text that informs you of how many enemies you have defeated, and how many times you have perished.

Tools:

- Game Engine: Godot
- Art and Graphics: Some textures in Photoshop, 3D models in Blender.

- Audio: Some sounds and tracks edited in Audacity. Music does not belong to us with the exception of the main menu music, most from Elden Ring and Bloodborne, and the popular Crab Rave. SFX from Minecraft, and free sources on Youtube.
- Version Control: Google Drive and Github.

System Design:

Each main interactable game object (crab, fish, lobster, match, shell) has its own single script. They use triggers set to differing layers and masks to communicate with each other (hazard layer, player body layer, player claw layer, collectible layer, environmental layer).

All cameras, on the other hand, are controlled by a script within each main stage. That script refers to several aspects:

- The player's position.
- The existing enemies.
- Which camera is currently active.
- The state of the player (in cutscene, alive, dead).

On the other hand, our singleton only needs to manage two things:

- Player deaths.
- Enemy defeats (reset by level script upon player death).