Beaver Shooting Game

Arcade game made by: Prestislav Dimitrov, Martina Aleksieva, Kristiyan Pavlov

This project is a Pygame-based "Beaver Shooter Game" where the player controls a beaver to shoot at incoming enemies, likely otters judging by the vidra.png filename and image.

1. Core Gameplay:

* Player Character (Beaver): The player controls a beaver that can move vertically (up and down) within a defined play area. The beaver is equipped with a gun (pushka.png) that can be aimed by adjusting its angle.
* Shooting Mechanics: The beaver can shoot bullets, with a limited ammo capacity (15 bullets) and a reload time (2 seconds). There's also a delay between shots.
* Enemies: Enemies, likely otters, spawn from the right side of the screen and move towards the beaver. Their speed increases with the player's score, introducing difficulty progression through "stages".
* Collision Detection:
  + Bullets destroy enemies upon impact, increasing the player's score.
  + Enemies colliding with the beaver reduce its health points (HP).
* Game Over: The game ends if the beaver's HP drops to zero, or if the score drops to zero due to enemies passing by. A "Game Over!" message is displayed with an image and a sound effect.

1. Game Boundaries and UI:

* Dynamic Play Area: The game dynamically calculates an upper vertical boundary for both the beaver and enemies to prevent them from overlapping with the Heads-Up Display (HUD) elements. This boundary is determined by the height of the ammo text and a vertical offset.
* Beaver Movement Limits: The beaver's vertical movement is strictly confined to the playable area, preventing it from going above the calculated upper limit (below the HUD) and below the bottom edge of the screen.
* Enemy Movement Limits: Similarly, enemies are constrained to spawn and move within the calculated vertical boundaries, ensuring they don't enter the HUD area or go off the bottom of the screen.
* HUD: The game displays essential information such as HP, current ammo, maximum ammo, current score, and the current stage. A "RELOADING..." message is shown when the beaver is reloading.

1. Technical Details:

* The project is built using pygame and asyncio for the game loop, making it potentially compatible with web environments like Emscripten.
* The game structure is modular, with separate Python files for GameManager, Beaver, Bullet, Enemy, and constants.

1. Restart Functionality:

* Players can restart the game by pressing the 'R' key after a "Game Over!" or manually initiate a reload.