Zap!

By: Daby Ibegbu and Chinonyerem Ukaegbu

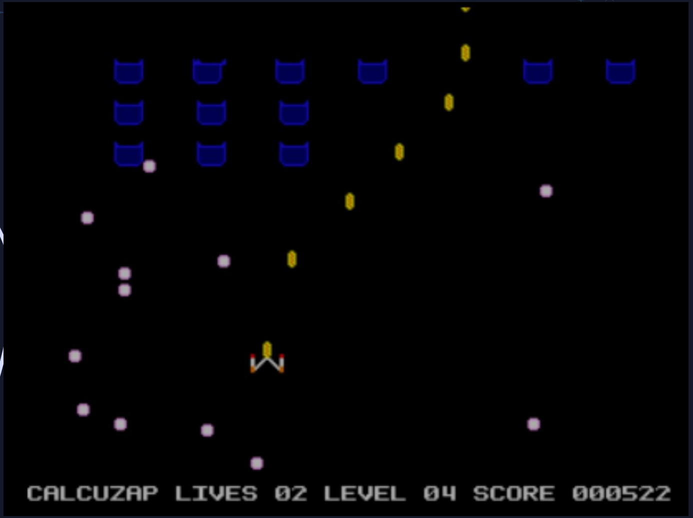
Description

Zap! is based on the popular arcade game Asteroids with a few, slight modifications.

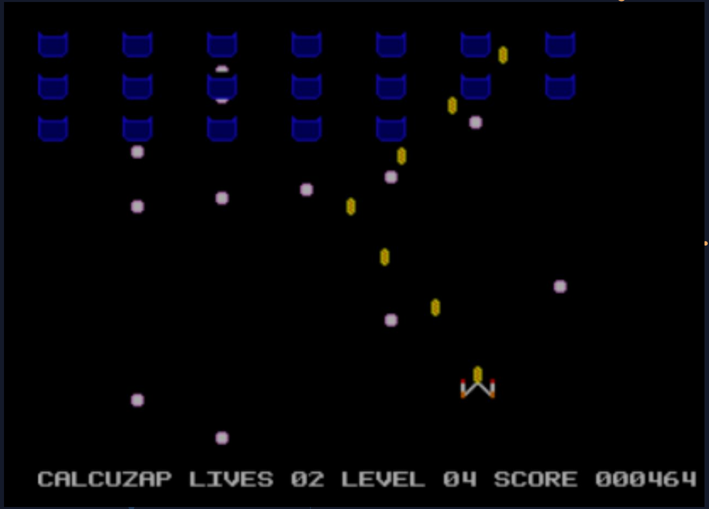
It is a space-themed one player game which features the single player controlling a spaceship that shoots laser beams.

The objective of the game is to defend the spaceship from attacks from the attacking spaceships that would periodically shoot laser beams at the spaceship.

Each hit inflicts damage to the spaceship, resulting in a deduction from the number of lives.



CALCUZAP LIVES 02 LEVEL 04 SCORE 000522



CALCUZAP LIVES 02 LEVEL 04 SCORE 000464

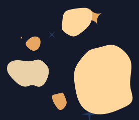
Description

The game begins with 3 lives and the objective is to eliminate all attack spaceships from the game without losing any or all the player’s lives.

If all the lives are exhausted before the attacking spaceships are eliminated, the player loses the game.

The player uses the arrow keys to move in the respective directions and uses the spacebar to shoot laser beams at the attacking spaceships.

The saucers in turn move in a clockwise shape and continuously shoot laser beams downwards, until they are eliminated.



Game Setup

For the initial setup of the game, an instruction page will be visible, and the player would be asked to click to start the game.

We will import images for the background and the objects and sound for the background and objects.

On the game page, the attack spaceships should be moving in a clockwise rotation. The attack laser beams should be able to shoot at random from any of the attack spaceships. They should shoot downwards.

The player must use the spacebar to shoot and move with the left and right keyboard keys.

The attack spaceships should not be affected from the attacking laser bullets. When a laser beam hits any of the spaceships, that object instance must be removed from the game.

Implementation

The implementation of the Zap! game should have the following Python classes:

1. Game class (Daby and Chinonyerem (split by number of methods))

   a. Method to destroy the attacking ship

   b. Method to restore lives if character loses a life.

   c. Method to end game

   d. Method to display

Implementation

2. Spaceship Superclass (Chinonyerem)

   a. Subclass for the player spaceship

      i. Methods to move spaceship

      ii. Methods to display

   b. Subclass for the attacking spaceships

      i. Method to move spaceships

      ii. Method to display

3. Laser beam Superclass (Daby)

   a. Laser beam subclass for player ship

      i. Method to allow the keyboard input (spacebar) to shoot a laser beam upwards.

   b. Laser beam subclass for attacking ship

      i. Method to allow bullets fall downwards

      ii. Method to shoot bullets at random

Score, Lives and Game Over

SCORE AND LIVES:

At the start of the game, the player has three lives and the score is initiated at zero.

When the player shoots one attack spaceship, the score is incremented by ten. If the player is hit and removed by the attack, their lives are reduced by one.

Levels could be implemented to the game by essentially restarting the game and increasing its speed after the player successfully defeats all the attack spaceships. There would be three levels.

GAME OVER:

The game ends when either the player shoots all the attack spaceships in the three levels, or the player loses all their lives.

The player has the option to click anywhere on the screen to restart the game.

THANK YOU

Any questions?