CPSC 4160 – 6160 / Crash Landing

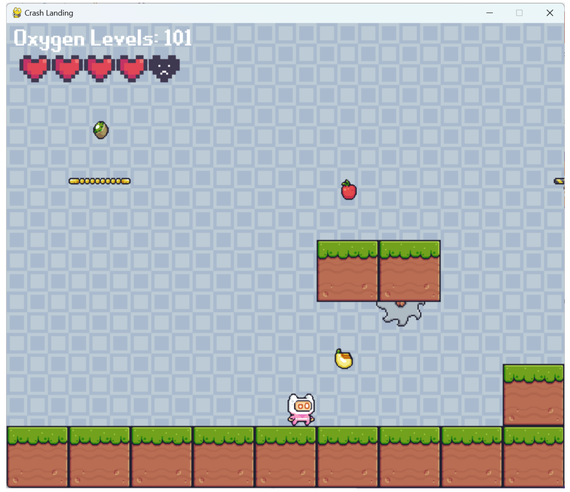
Game Description

Crash Landing involves an astronaut stranded on another planet. The astronaut must make their way through each zone to find their crewmates while navigating traps, enemies, magical fruits, and their depleting oxygen. The game is designed to play as a speed run using each fruit to your advantage or disadvantage.

Game objective

The objective of the Crash Landing is to find each crewmate in their landing zone. At the end of the game, the player is given their oxygen level left/time left to compare with themselves and others. The player controls the astronaut to complete the levels with the fastest time.

The player must learn what each fruit does through trial and error. Some fruits has their own timer before its effects wear off. Other fruits give permanent damage or buffs.



Game Mechanics

The gameplay involves jumping on moving platforms, avoiding traps like saws and spikes, and reaching the end goal of the other crewmate.

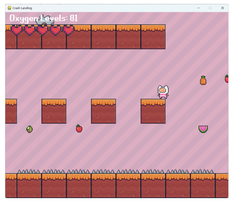
Key mechanics include collecting certain fruits while avoiding others to strategically activate each of their own powerups. The player must avoid taking too long or else their oxygen is gone, and the game is over.

Technical Description

Crash Landing for the average laptop and was programmed in python language, specially pygame for creating games. The game runs at a consistent 60 frames per second.

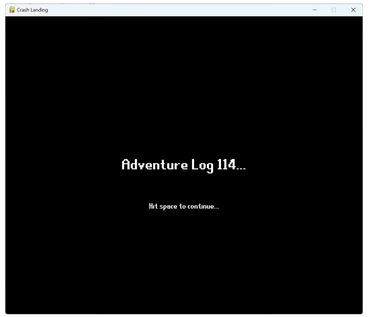
$$\text{fps} = \frac{\text{Number of Frames}}{\text{Time (in seconds)}}$$

The game uses delta time, the amount of time that has passed since the previous frame, to allow for more consistent movement and framerate across any system.



Controls

The controls are the keyboard arrow keys: up for jumping, up twice for double jump, and left and right for running. For a certain fruit, the game controls are flipped: down is jump, double down is double jump, right is left, and left is right.



Game Limitations

Pygame is beginner friendly free way with multiple free resources to learn how to code games. There are still limitations. Pygame freezes the game once the user has clicked off screen. Pygame also does not have basic physic, so complex physics must be implemented.

Future Work

Future iterations might incorporate more levels and more traps such as a swinging saw.

They made also include more fruits and/or more variations of each fruit. For example, half an apple may only do half of its powerup or half of the time length.