Jose Chavez Ramirez

EE104 Section 1

Professor Christopher Pham

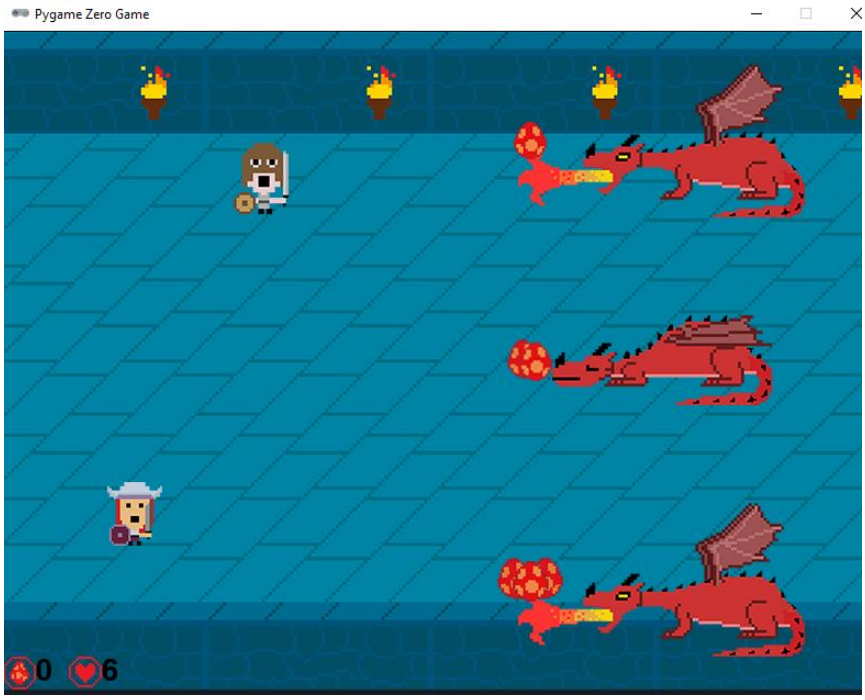
April 23 2023

EE 104 Lab 7

https://youtu.be/m4rdY-XgB8Y

Dragon Egg Hunt is a two-player game where players act as heroes, collecting dragon eggs from three different dragon lairs of varying difficulties. The game world is navigated using arrow keys and WASD keys. Each dragon is sleeping, and players must avoid waking them up. The objective is to collect all eggs in each lair, without waking a dragon or losing all lives.

To install, clone the repository using "git clone https://github.com/yourusername/dragon-egg-hunt.git" and install the required dependencies by running "pip install -r requirements.txt". To play, navigate to the project directory ("cd dragon-egg-hunt") and run the game ("python main.py"). Player 1 moves using arrow keys and player 2 uses WASD keys.As players collect more eggs, the randomness of the dragons' sleep schedule increases.



COCO Image Dataset Splitting Script

This script is written in Python and is designed to split the Microsoft Common Objects in Context (COCO) dataset into two subsets: one for training and one for validation. These subsets are intended for use with the YOLOv5 object detection algorithm.

To use this script, you must first download and extract the COCO dataset, and make sure you have Python 3.x and YOLOv5 installed. You'll also need to create directories where you'll store the training and validation subsets.

After setting up the dataset, open the script in a text editor and modify the following path variables to match your file system

coco_images: path to the directory containing the dataset's images

coco_labels: path to the directory containing the dataset's annotations

ee104TrainImagePath: path to the directory where the training subset's images will be stored

ee104TrainLabelPath: path to the directory where the training subset's annotations will be stored

ee104ValImagePath: path to the directory where the validation subset's images will be stored

ee104ValLabelPath: path to the directory where the validation subset's annotations will be stored

You can adjust the train_ratio and val_ratio variables to control the size of the training and validation subsets. By default, an 80-20 split is used.

Once you've made the necessary changes to the script, save the file and run it in a Python environment to generate the training and validation subsets.

```
Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/ultralytics/yolov5_ee104_split_train_val_files.py', wdir='C:/ultralytics')
Total number of COCO images are :  180
Number of training images are   :  144
Number of validation images are :  36
Traceback (most recent call last):

  File "C:\ultralytics\yolov5_ee104_split_train_val_files.py", line 99, in <module>
    shutil.move(coco_images, ee104ValImagePath)

  File "C:\Users\Jose\Documents\Anaconda\Anaconda_files\lib\shutil.py", line 823, in move
    raise Error("Destination path '%s' already exists" % real_dst)

Error: Destination path 'C:/ultralytics/ultralytics/datasets/datasets/coco128/images/ee104_val
\train2017' already exists


In [2]:
```