cmpg313 practical 3

Nkosi MB 30233739

Name: MB Nkosi Student Number: 30233739

Table of Contents

[Introduction 1](#_Toc42613875)

[Literature study 1](#_Toc42613876)

[Application Design 1](#_Toc42613877)

[Code Snippets 3](#_Toc42613878)

[Reference: 9](#_Toc42613879)

# Introduction

The project that shall be created will be a 3x3 sliding puzzle game with images as tile using python for the programming language. The reason behind the programming choice is to increase improve my proficiency in python and the reason for the project is to develop skills for a game development career and to submit for marks towards my degree.

# Literature study

Multiple resources were access and piled together in order to develop the project and were used to luminate the steps to take towards the completion of this project. First objective to be was to play a slide puzzle game to ensure I had a concrete understanding if what I wanted my end goal to look like. Following this was revising my knowledge in Python using an application *Sololearn* then looking at tutorial videos that focused on: how to use *pygame ­­*, a module of python, and retrieve values from a file into the project. The most focused tutorials were ones on creating a sliding puzzle using python in order to ensure that I was aware of every aspect. During the course of creating the slide puzzle sites pertaining to more detailed methods to use *pygame* were attained and used thus improving overall performance.

# Application Design

A class named *SlidePuzzle* was used hold the functions that would be used towards the design and application of the project. When initialized it must receive values for grid size of the puzzle whether it will be a 3x3 or 4x4 puzzle, then the length of each tile and finally margin, space between tiles.

Variables that are used constantly and across multiple functions are initialized in order to be accessed when require. Arrays were used in order in hold values for the dimensions of each tile using *tiles* array, the position of each tile using *tilepos ­*and the current position of every tile in the slide puzzle using *currentpos* that would be used later to check if the user has solved the slide puzzle.

The first function that was implemented was the *draw* function which was used to make the slide puzzle visual. The process for this is creating a background for the slide puzzle window, currently color is used, then add in each tile one at a time while simultaneously retrieving the image to be used for the puzzle and setting each tile to the corresponds image location.

The second set of functions that were implemented *getblank* that retrieves the location of the empty tile, *setblank* that designates the empty tile, *switch­* to switch the empty tile with an adjacent tile and *randomize* that would place every tile in a random position which is to be used when the slide puzzle starts.

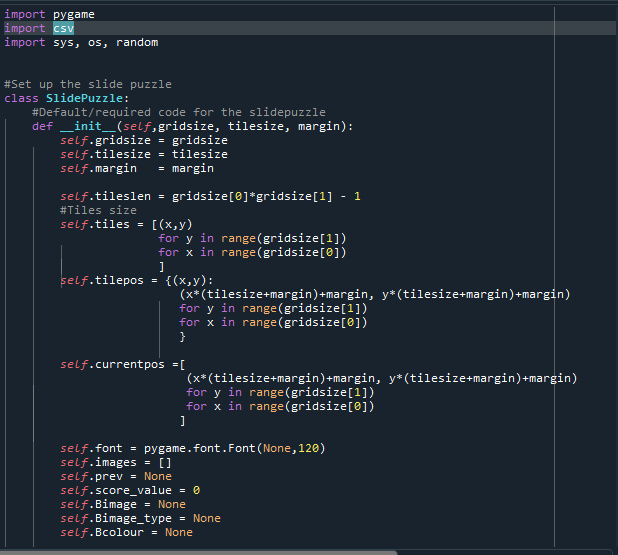
The functions *adjacent* ensure a tile is adjacent to another tile and *in*\_*grid* was used to ensure when the user clicked inside the slide puzzle that they clicked on a tile a not in between one.

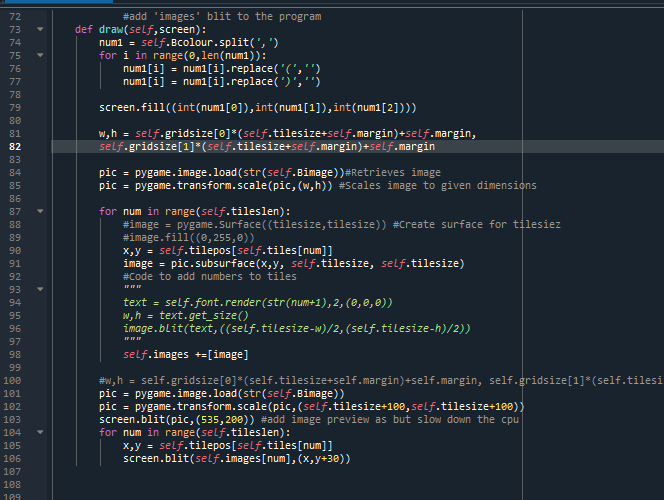
The third set of functions were *show\_score, complete* and *cords.* The ­*cords* function was created first in order to determine the actual coordinate of every tile in real time and the goal coordinate for every tile. Its purpose was initially to determine how to ensure the coordinates were correct. It was a layering process for the entire project. The *show\_score* was used to display the users score and would congratulate the player once they solve the puzzle and *complete* was used to determine when the puzzle was solved thus when to congratulate the player.

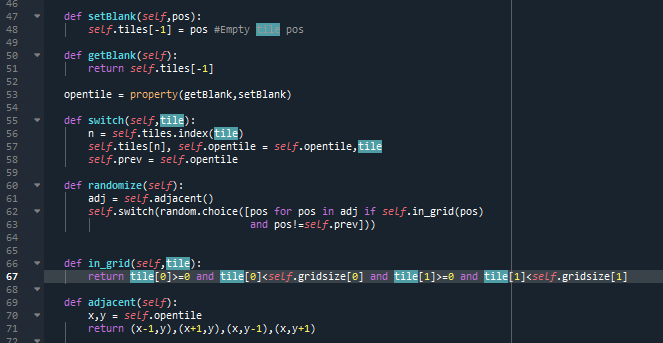
The function *Slidepuzzleimage,* and *setTile­* were used to read csv files and retrieve data that would be used in the program such as adding images and setting tile positions.

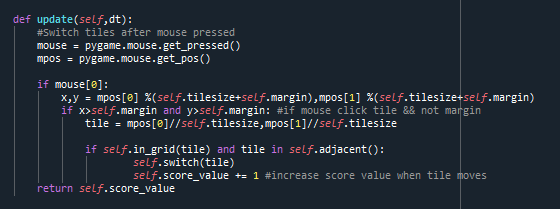
The main is where everything is called and initiated for the program. The size of the program screen is coded here as well as when to call the functions to ensure the program runs correctly. Two version were created: one that would be slide puzzle with images for tiles and another with numbers for tiles.

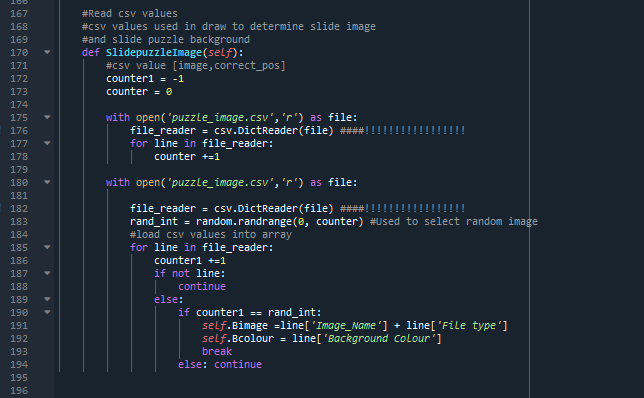
# Code Snippets

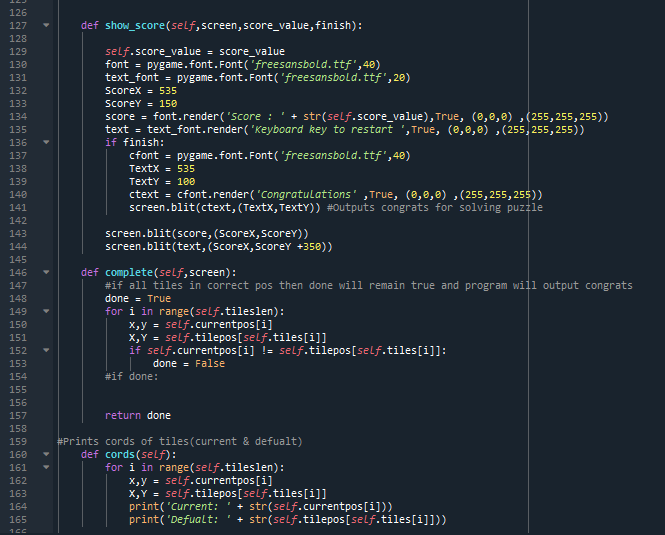


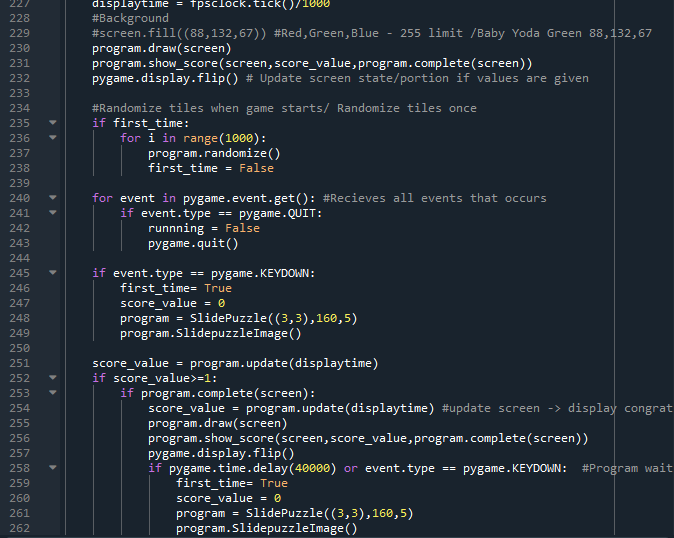
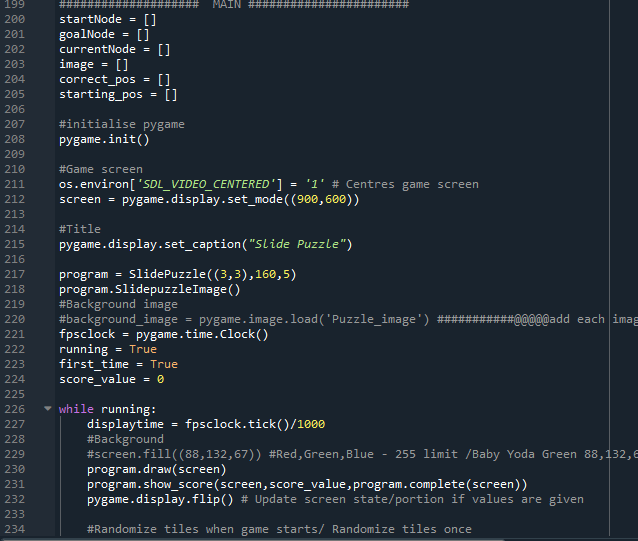
****

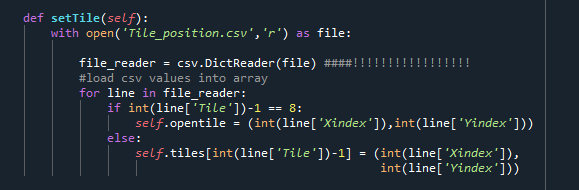
****

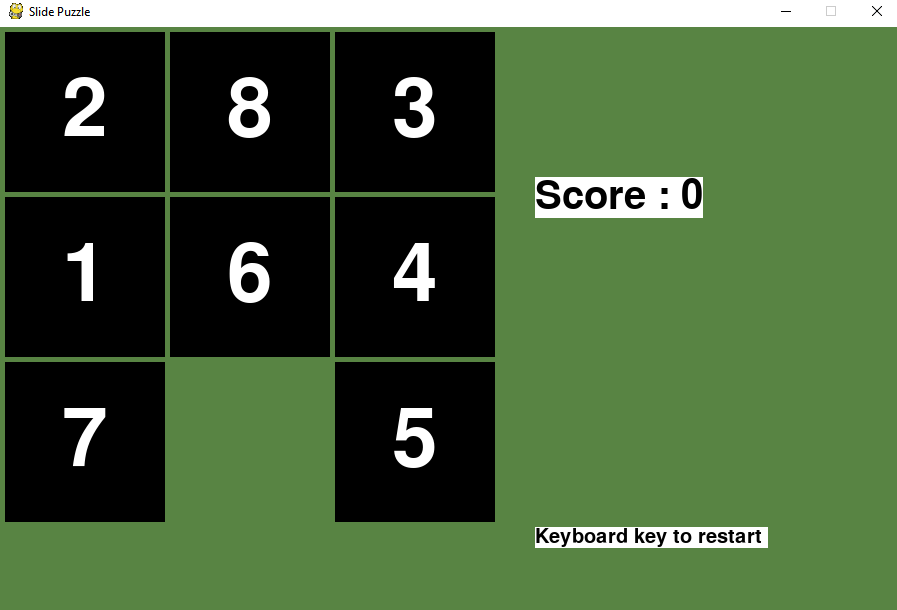
****

****

****

****

****

****

# 

# Reference:

GeeksforGeeks, python display. Site acceded: <https://www.geeksforgeeks.org/python-display-text-to-pygame-window/> .Date: 8 June 2020

Proprof Brain Games. Site accessed: <https://www.proprofs.com/games/puzzle/sliding/> . Date accessed: 27 May 2020

Pygame Index. Site accesed: [pygame.event — pygame v2.0.0.dev7 documentation](https://www.pygame.org/docs/ref/event.html) . Date accessed: 8 June 2020

Youtube,How to program slide puzzle(in Pythom) :

<https://www.youtube.com/watch?v=afC3dq9MeJg> . Date accessed: 5 June 2020

Youtube,Speed Coding| Python- sliding puzzle game:

<https://www.youtube.com/watch?v=vzzEP_YI424> . Date accessed: 5 June 2020

Youtube, PyGame tutorial for beginners - Python game development: <https://www.youtube.com/watch?v=FfWpgLFMI7w> . Date accessed: 5 June 2020

Youtube, Youtube Tutorial on how to read write and parse csv files : <https://www.youtube.com/watch?v=q5uM4VKywbA> Date accessed: 5 June 2020