

University of New Haven



Tagliatela College of Engineering - Computer Science

CSCI - 6651- 01

Introduction to Python Scripting

Fall 2021

PLUMBER PUZZLE GAME

Professor: Dr. Mehdi Mekni

Group- 3

1. Kavya Sandireddy
2. Alisha Kareemulla
3. Abhinav Andrews Giduturi
4. Jordon Saleh
5. Jason Ash

ABSTRACT

- Clean tap water is not a given in the world. Across the world, in major cities and small towns, drinking water is contaminated by industry, aging infrastructure, and government oversight. To majorly focus on the piping infrastructure problem, the game is designed with awareness in mind
- This puzzle game is all about transporting water from source to the destination by connecting the pipes correctly and avoiding the wrong routes. When the player progresses through the levels, the project aids in the education of others about the importance of water safety. The player must complete three levels in order to win the game. Each level has a different level of difficulty. Furthermore, when the person attempts to play the game, they face several obstacles.
- During this procedure, the player will learn about different barriers that might lead to faulty infrastructure. This information will aid the player in conserving water in future, as well as having fun when playing the game by encountering problems and overcoming them in order to win. This project is implemented by using Python modules like 'pygame', 'tkinter' and PyCharm interface.

INTRODUCTION

- This is a fun thrilled puzzle game where a player could not only have fun but also train and boost their problem - solving skills, as well as vision and speed skills. It helps in creating awareness among individuals irrespective of their age, gender and profile.
- Individual playing this game must go through the 3 levels in order to complete it. The later levels will be unlocked only if they complete the current level of the game.

Motivation For The Game



- Creates awareness among several individuals irrespective of age, profile.
- Helps in educating people about the awareness of water safety when they come across different levels when they pass through each level.
- Player will not only have fun but also train and boost their problem - solving skills, as well as vision and speed skills.

Alignment of 3 C's

- **Curiosity:** The project will be a puzzle game about planning a grid of pipe placements to get water from a source to an exit while avoiding pollution, dry wells, and all sorts of obstacles. The subtext will be embedded with the game play loop so that the entire experience is a holistic and entertaining game.
- **Connections:** The game will use python, pycharm and pygame as a solution to creation of the game. Other assets will be made with art programs like Adobe Photoshop. The game will be made from scratch as to practice coding in python and learning the pygame libraries.
- **Creating Value:** The project will connect with concurrent water issues in the U.S and the greater world. Our water retention and scarcity message will connect to the idea that we should design better infrastructure to keep pure water within our area of usage. The game can show some ways in which we can conserve the water as themes and dialog.

Phases of the Game

- PHASE-1 Brainstorming and planning in designing the game
- PHASE-2 Creating assets and learning the packages being used.
- PHASE-3 Coding and Implementation
- PHASE-4 Testing and deployment

Tools used for designing the Python Game and Packages being used :

- PYTHON
- PYGAME
- TURTLE
- TKINTER
- PYGAME INTERFACE

Implementation of Game

- First, the pipe objects are initialized with a dictionary containing their type and their default rotational value
- Next, the objects are created in sprite windows with the defined dictionary values, and a mirror dictionary is also used to compare the current game state to the winning solution
- The player can rotate pipes to create a path from the top left to the bottom right, making sure every route is closed
- These rotations are recorded, and compared to the solution every time and if the solution equals the current game state, it is considered a “Win”
- 3 Main things must work for the game to function
 - Organized Coordinate grid
 - Mapping of correct pipes to grid spaces
 - Rotatability of tiles, and saving the state

Grid and Tile Mapping

- This grid and tile map functions help in mapping coordinates with tiles, and allowing mouse collisions
- The tiles are in both columns and rows within spacing of 50 pixels from each other.
- The grid is alphanumeric, one number and letter (IE A1, A2,A3)
- This allows organized level design without having repeat spaces or double numbers to parse
- Assigning these tiles to specific placements is paramount in creating levels, each grid space is occupied by a tile type stored within the level dictionaries

```
#Coordinate system, (IE battleship) for the pipe grid, utilizes pixel measurements of 50 increment
```

```
coords = {'A1': (25, 50), 'A2': (25, 100), 'A3': (25, 150), 'A4': (25, 200), 'A5': (25, 250), 'A6': (25, 300),  
          'A7': (25, 350),
```

```
level1_pipes = {  
    'A1': {straight_pipe, 180}, 'A2': {curve_pipe, 0}, 'A3': {curve_pipe, 270}, 'A4': {t_pipe, 90},  
    'A5': {curve_pipe, 0}, 'A6': {straight_pipe, 90},  
    'A7': {t_pipe, 180}, 'B1': {straight_pipe, 90}, 'B2': {straight_pipe, 0}, 'B3': {curve_pipe, 180},  
    'B4': {straight_pipe, 90}, 'B5': {t_pipe, 270}, 'B6': {curve_pipe, 0},  
    'B7': {straight_pipe, 0}, 'C1': {curve_pipe, 0}, 'C2': {curve_pipe, 180}, 'C3': {straight_pipe, 180},
```

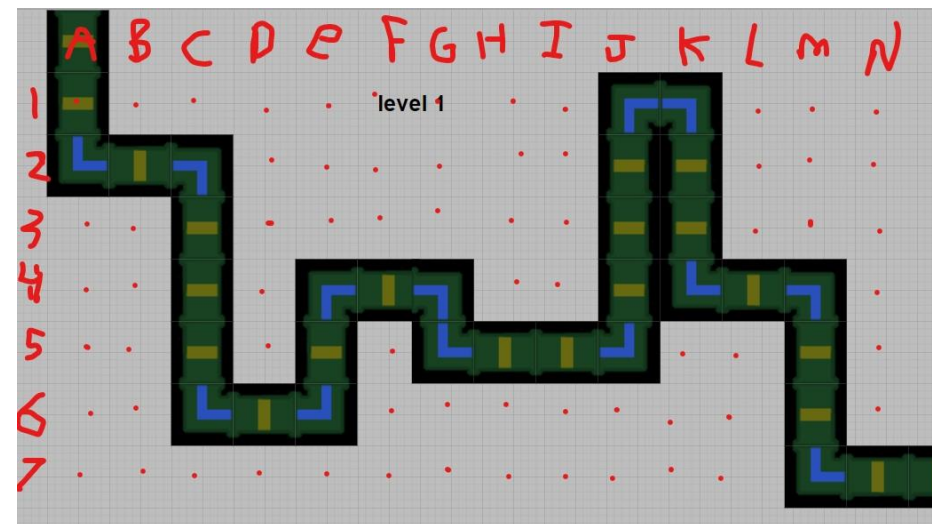
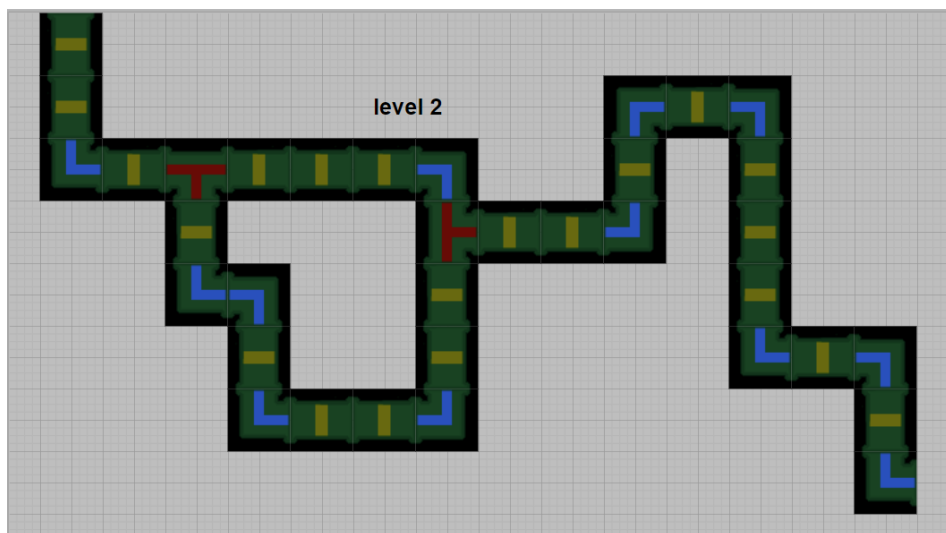
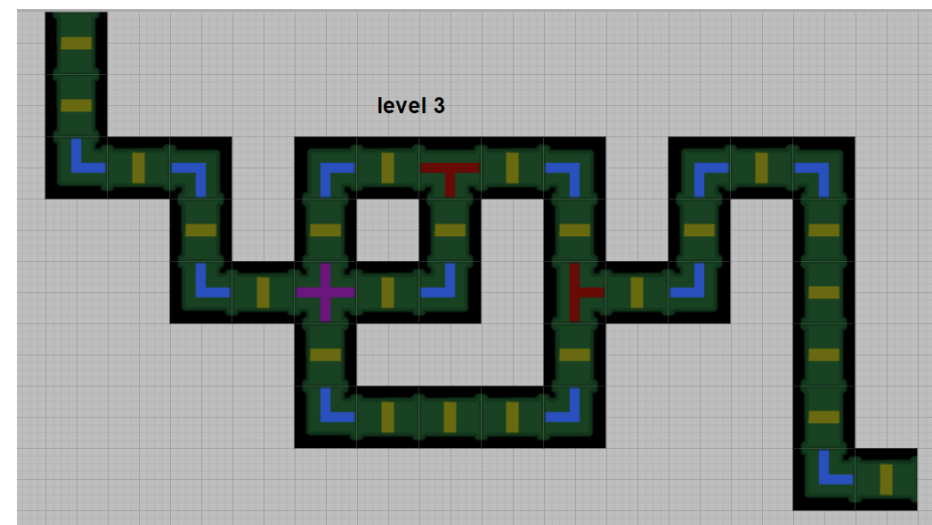
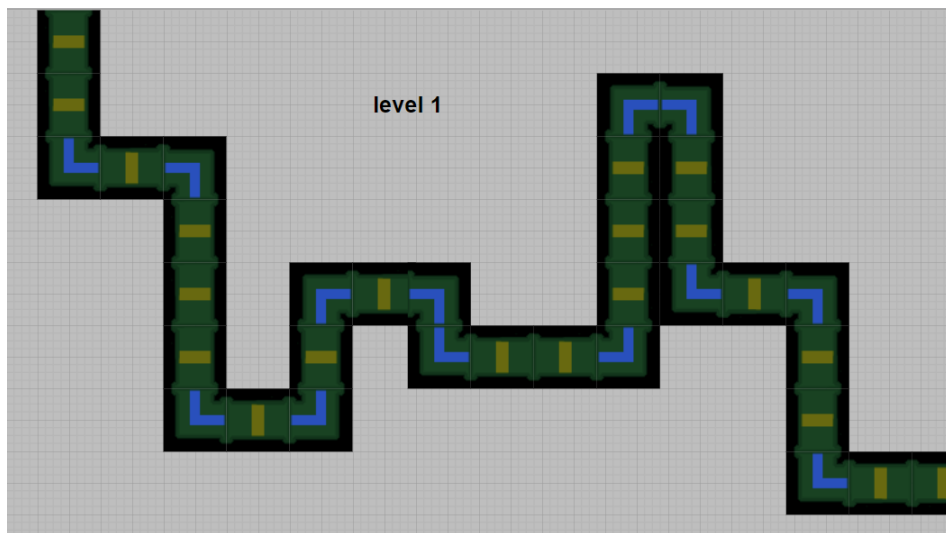


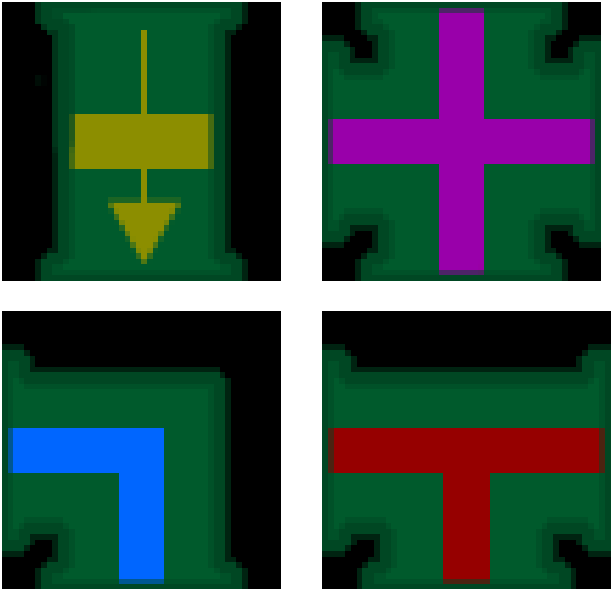
Image Rotation

- This Image rotation function is the main function which rotates the tile 90 degrees once the user clicks on a pipe.
- All the pipes pass through the function once a user clicks on the pipe.
- The modified pipe would be restored in the grid, along with its corresponding degree value once it has been rotated
- Pygame must copy and rotate the copied sprite, because of graphical rendering shortcomings
- Once a pipe passes 270 and goes to 360, its rotation is reset back to 0 so it may continue to rotate between fixed values
- Without a reset, the degrees would continue to climb past 360 and become much harder to check

- Each pipe has 4 permutations (Besides plus pipe) meaning that we must record each in 90-degree intervals, and allowing 13 total pipe permutations
- Levels must be designed for maximum 1 solution for logic checks to work easier
- The levels ramp up in complexity, but will always end a straight connection, even if multiple lines are needed to close a loop
- This system allows easier level design, and expedites level creation to a system

SP = Straight
 CP = Curved
 TP = T-Pipe
 PP = Plus pipe

TP-0	TP-90	TP-180	TP-270
CP-0	CP-90	CP-180	C-270
SP-0	SP-90	SP-180	CSP-270
PP-1	PP-1	PP-1	PP-1



Comparison of the Player Grid and the Original Grid

	A	B	C	D
1	SP-0	SP-90	TP-180	SP-O
2	CP-90	PP-270	CP-O	CP-90
3	TP-90	SP-90	SP-O	CP-180
4	CP-180	SP-O	CP-90	SP-O

SP = Straight
CP = Curved
TP = T-Pipe
PP = Plus pipe

Consider these are winning conditions in a matrix.

(A,1)=0
(A,2)=180
(B,2)=0
(B,3)=0
(B,4)=0
(C,4)=0
(D,4)=90

Once player aligns pipes from start to finish, then the coordinates of the game state dictionary match the solution matrix

State(0,1)=Solution(1,0)

State(1,0)=Solution(1,0)-----→Correct

If state matches solution, then the player wins the game.

Challenges faced in implementation of Game

- Intersection of Pygame interface and Tkinter.
- Bugs caused by Pygame window rotations
- Creating advanced logic for handling win and game state
- Designing assets for each part of the game
- Level design with only 1 proper solution

Things to improve in future

- Advanced levels could be implemented using other data structures
- Multiple paths could be added and increase complexity.
- Implement obstacles
- User interface can be optimised, Animations can be added for water flow from initial to final point.
- Multiple levels to be added, Streamline a design tool

Thanks for Listening!