Requirement:

Python 2.7.13

opencv-contrib-python 3.3.0.10

Tensorflow <=1.15.0

The programs were orignally ran on windows machine, for Linux/MacOS, please check the directory sign accordingly("\\"->"/")

The python files:

Path extraction:

./video parser - template match/match-init.py

parameters: --level [int] --input [string.mp4]

This program will take the level you want to track path from(the level is used to find the annotated level from folder "./Processed"), and the video of that

level(recorded play of that level, the video should be stored at "./videos").

Then this program uses the players sprites(saved at "./video parser - template match/player\_ref\_diff") to match the player location frame by frame.

Finally it will generate two files containing the path information in format of [level].txt under folders "./video parser - template match/results-dig" and

"./video parser - template match/results-no-dig"(as names suggest, one contains the dig action and the other one does not).

Path generation:

./path\_train.py

This program trains all extracted sequences and generates new sequences. Credit goes to <https://github.com/NELSONZHAO/zhihu>.

Extra data extraction:

./gen\_path/train\_markov.py

parameters: --level [int]

This program will take the annotated level information(from "./Processed") and path information(from "./video parser - template match/results-no-dig") from level 1 to input level. Then train the two Markov models and Gaussian distributions

for enemies and golds. The models will be save in its directory as "markov\_stat.pkl", "half\_markov\_stat.pkl" and "stat\_eg.pkl".

Level generation:

./gen\_path/path\_gen.py

parameters: --input [str.txt]

This program reads the given path information(from "./path\_data"), and generates a new level from it.

It will also require the "markov\_stat.pkl", "half\_markov\_stat.pkl" and "stat\_eg.pkl" files from its directory.

The generated level will be stored as annotated text at "./gen\_path/filled\_path/filled\_map\_[input]".

Visualization:

./gen\_path/visualize.py

parameters: --input [str.txt]

Code mainly copied from Dr.Guzdial's work on visualing Super Mario Brothers levels from annotated data.

This program will take input of generated annotation file(from "./gen\_path/filled\_path/[input]") to create an image that shows what the level looks like in actual game.

The produced image will be save at "./gen\_path/visualized/[input].png"

Evaluation:

./evaluate.py

This program assesses the level size and the proportion of empty space and interesting tiles both in the generated levels and the original levels.