MATCH-3

1. DESCRIPTION

Match-3 is a game with a goal to set three or more squares with the same color in one line (vertical or horizontal) by swapping two adjacent squares. Player receives points for each such arrangement, in accordance with the formula:

$$number\ of\ points = 10 * deleted\ squares * current\ level$$

With the passage of deleting more and more squares, player can promote to higher levels, each harder than previous (there is more and more colors on a board). When player has no move which would allow him to align three or more squares in a line, the game is over. Afterwards, the player is asked to write his nickname, which is saved with a result in file *result.yml*.

2. REQUIREMENTS AND INSTRUCTION

These are needed to play Match-3:

- Python Interpreter
- Library PySide2 (can be installed with a command *pip install pyside2*)
- Module PyYAML (command *pip install pyyaml*)

Before playing, ensure that the root directory of a game contains file *config.json*, with following data:

- an array with square colors (key *squares_color*) – there must be at least 13 different colors (if less, an exception is thrown while launching the program), which names are compliant with CSS color standard – they can be expressed as #RRGGBB or with one of 140 supported names: https://www.w3schools.com/colors/colors_hex.asp,

  It is highly recommended to use different hues, so that each color could differ from another.

- color, which clicked squares will be marked in (key *mark_color*) – compliant with CSS color standard as well, it is strongly recommended not to write a color which is already present in *squares_color* array, so that marking squares would always be visible,

- size of board (*board_size*) – an integer from 5 to 12 (too big or too small number will cause an error while running a program) which states how many squares there will be in a single row/column.

With no config file or malformed file launching a game would be impossible.

Exemplary content od *config.json* file

Playthrough

Once the config data is adjusted, a game can be started. To start the game, launch file *main.py* using Python Interpreter. Then an interface will appear on a screen – board with squares on left side, and cyan bar on the right. This bar contains: number of the current level, number of received points, button “Swap” and the best result (taken from *result.yml* file).

To swap two adjacent buttons, click on them and then press button “Swap” – it will be enabled only when those two squares will make a threesome/foursome etc. when swapped. After clicking a square, it will be marked with 5px border frame, which color is the same as it has been set in config file (key *mark_color*). Re-clicking a button will unmark it.

After pressing “Swap” button, three or more squares with the same color will disappear and every square above will fall. During the process, clicking the squares will be impossible.

Promotion

When player gets so many points to promote to a next level, a message box with congratulation is displayed. To continue a game, press “Ok” button. Then the board will be deleted and new will appear.

Game interface with board of size 10 x 10

Loss

When player has no move left, a dialog box is displayed. The box informs a player how many points he received and asks him to write his nickname. After writing the nick, click button “Save result and quit”. Then, nickname with result is being saved to *result.yml* file.