Environment

The main project contains 2 scripts, `frontend.py` and `backend.py`

python-3.9.18

package needed:

```
pyzmq
PyQt5
```

To successfully run the hint functionality we provide, it's necessary to get `verifyta` following [official guidance](#).

And always remember to add its path into system’s environment variables, and test if it successfully executes in your `cmd`.

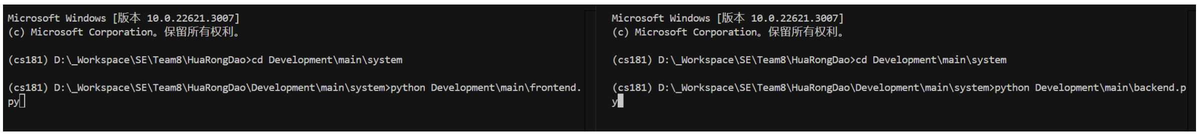
Tips:

If the `cmd` has successfully executed `verifyta`, but terminal in VSCode doesn't take that, please try to restart your VSCode, or your computer if necessary.

For the security of the execution of code, our data structure of file path:

```
├── system/
|    └── Development/
|        ├── main/
|            └── frontend.py # frontend
|            ├── backend.py # backend
|            ├── Check_tool_template.xml
|            ├── Check_tool.q
|            ├── 0.png #Chess image files, 0~6.png
|            ...
|            ├── 6.png
```

Please run frontend.py and backend.py at the same time in different terminals:



```
Microsoft Windows [版本 10.0.22621.3007]
(c) Microsoft Corporation。保留所有权利。

(cs181) D:\_Workspace\SE\Team8\HuaRongDao>cd Development\main\system

(cs181) D:\_Workspace\SE\Team8\HuaRongDao\Development\main\system>python Development\main\frontend.
py
```

```
Microsoft Windows [版本 10.0.22621.3007]
(c) Microsoft Corporation。保留所有权利。

(cs181) D:\_Workspace\SE\Team8\HuaRongDao>cd Development\main\system

(cs181) D:\_Workspace\SE\Team8\HuaRongDao\Development\main\system>python Development\main\backend.p
y
```

And always run the code in the path of system, instead of any of its child directory, for example:

Your `cmd` should looks like:

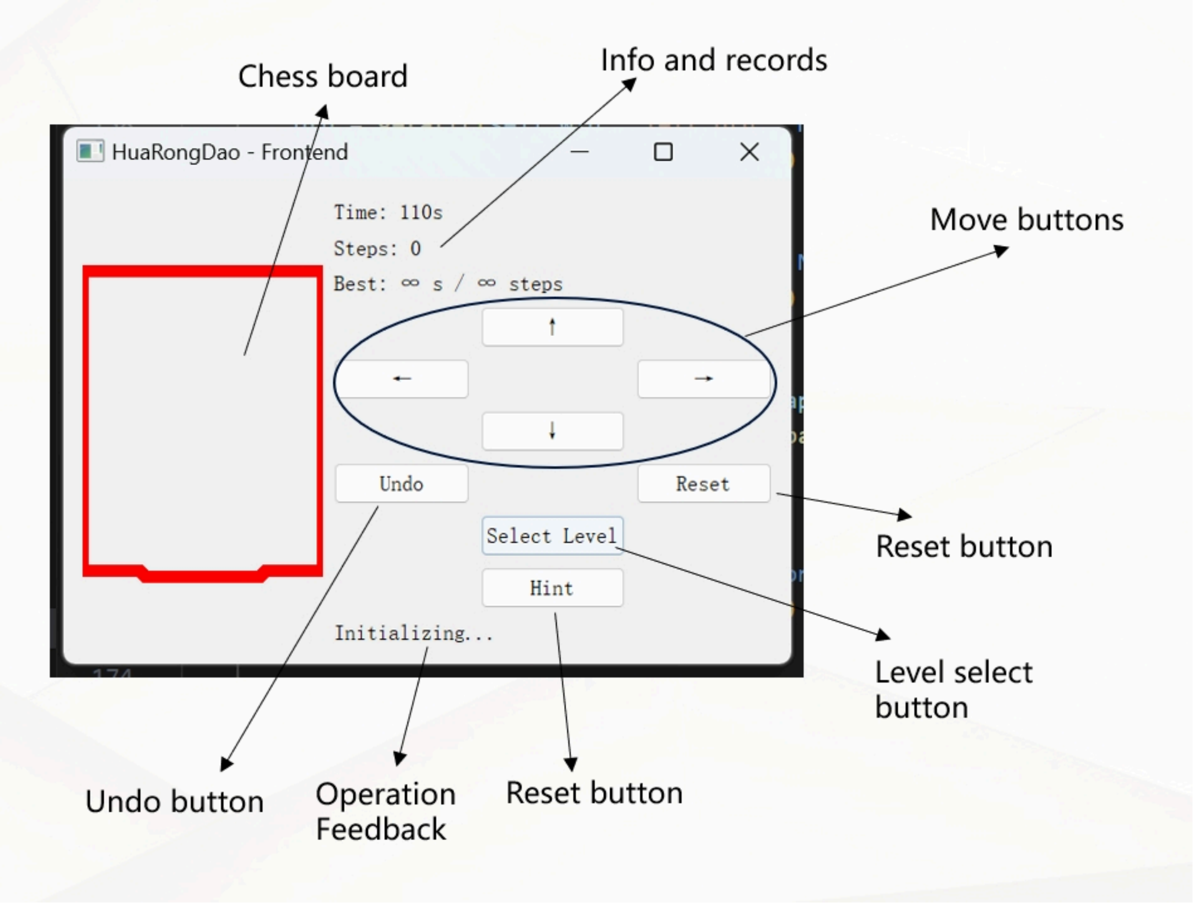
```
D:\Projects\...\system> python Development\main\frontend.py
```

This is to ensure chess images and UPPAAL files can be found.

If you hope to make use of official API, you could find scripts of `api2backend.py` and `UIControl_api.py` in `testing` helpful. Given that official API may not be necessary, we put it in `testing` directory.

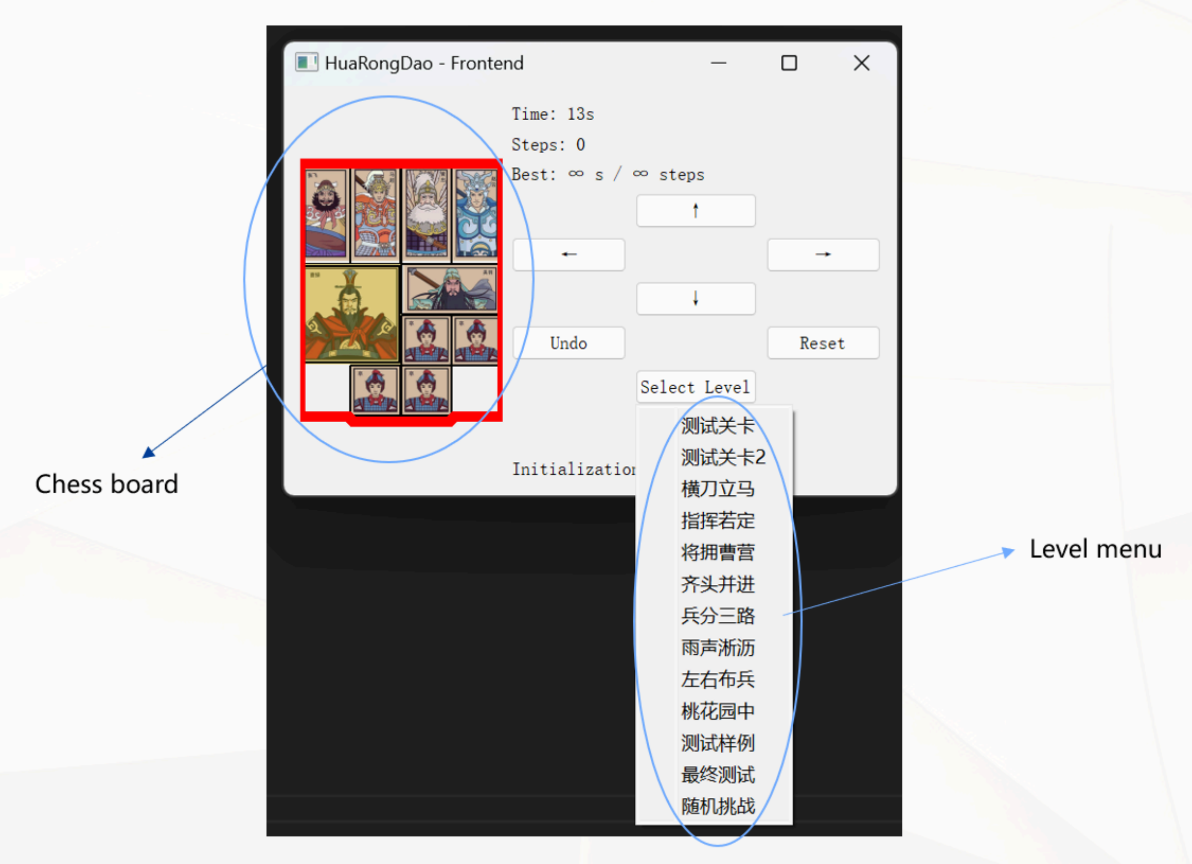
UI instruction

Initial panel



The initial panel sets like this.

Gameboard and level menu



Time: 13s

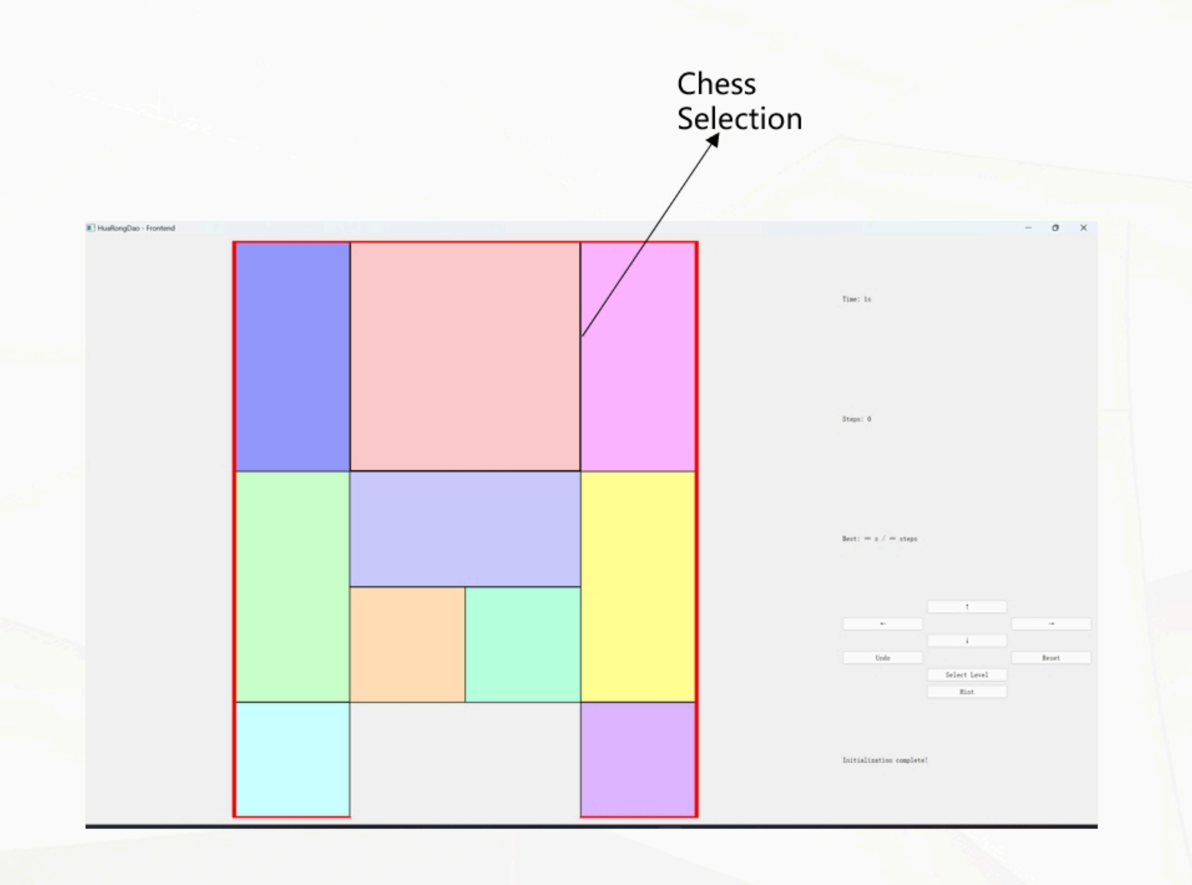
Steps: 0

Best: ∞ s / ∞ steps

Initializatior

If a chess is selected, it will be highlighted with yellow.

The level menu is quite easy to understand.



Time: 1s

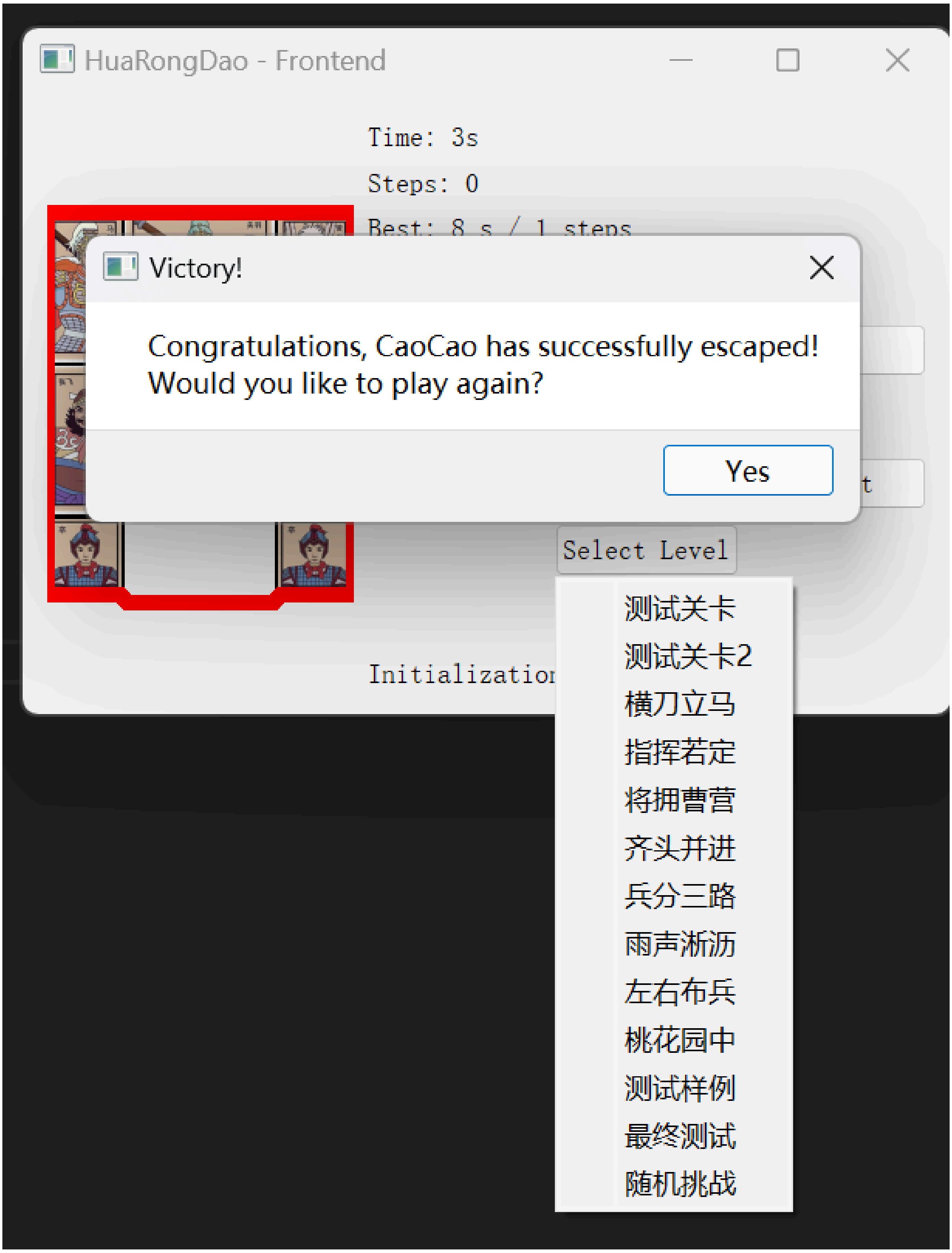
Steps: 0

Best: ∞ s / ∞ steps

Initialization complete!

If we cannot find image files, our frontend will paint chess with pure color. And then selection visualization will turn to black frame, instead of highlight.

Success



HuaRongDao - Frontend

Time: 3s

Steps: 0

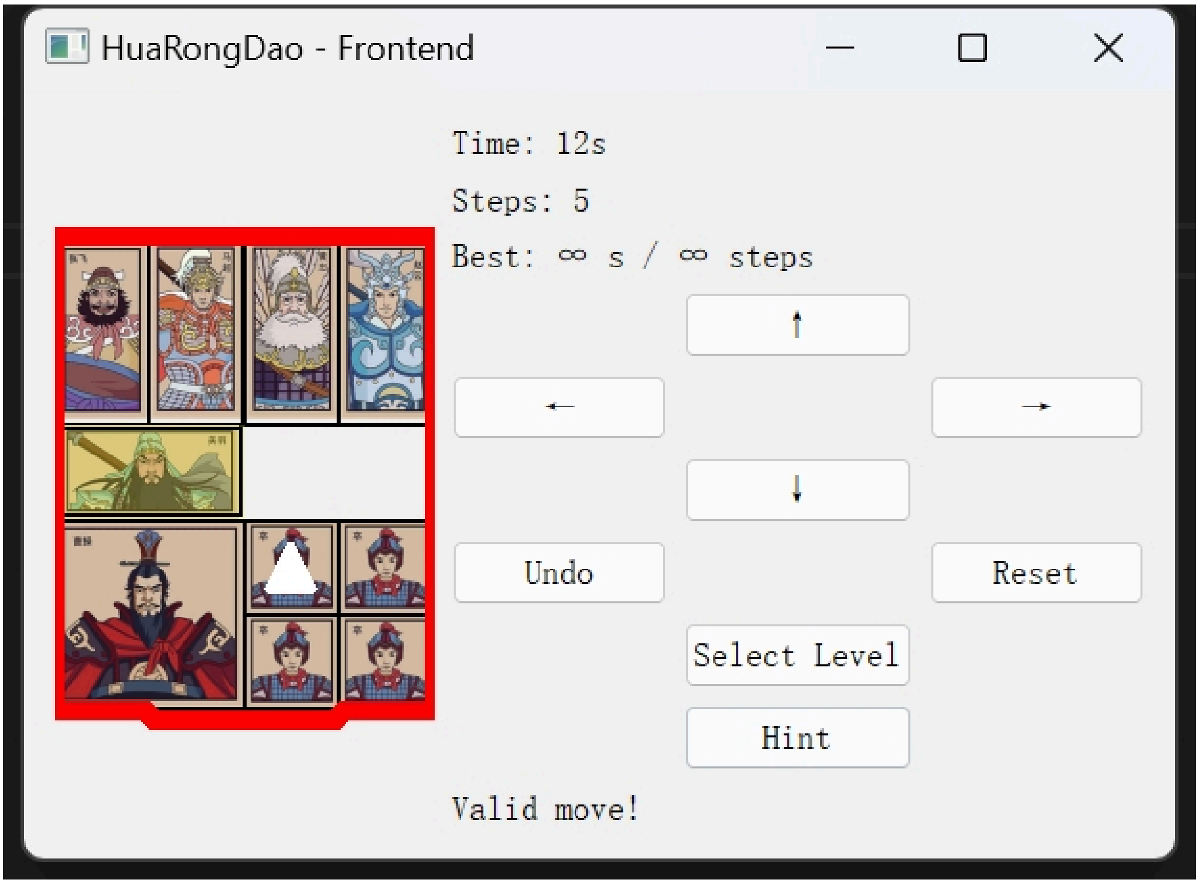
Best: 8 s / 1 steps

Victory!

Congratulations, CaoCao has successfully escaped!
Would you like to play again?

On victory, it will pop up a message telling you about the success. And the level menu will also pop up in order to recommend you to try other levels.

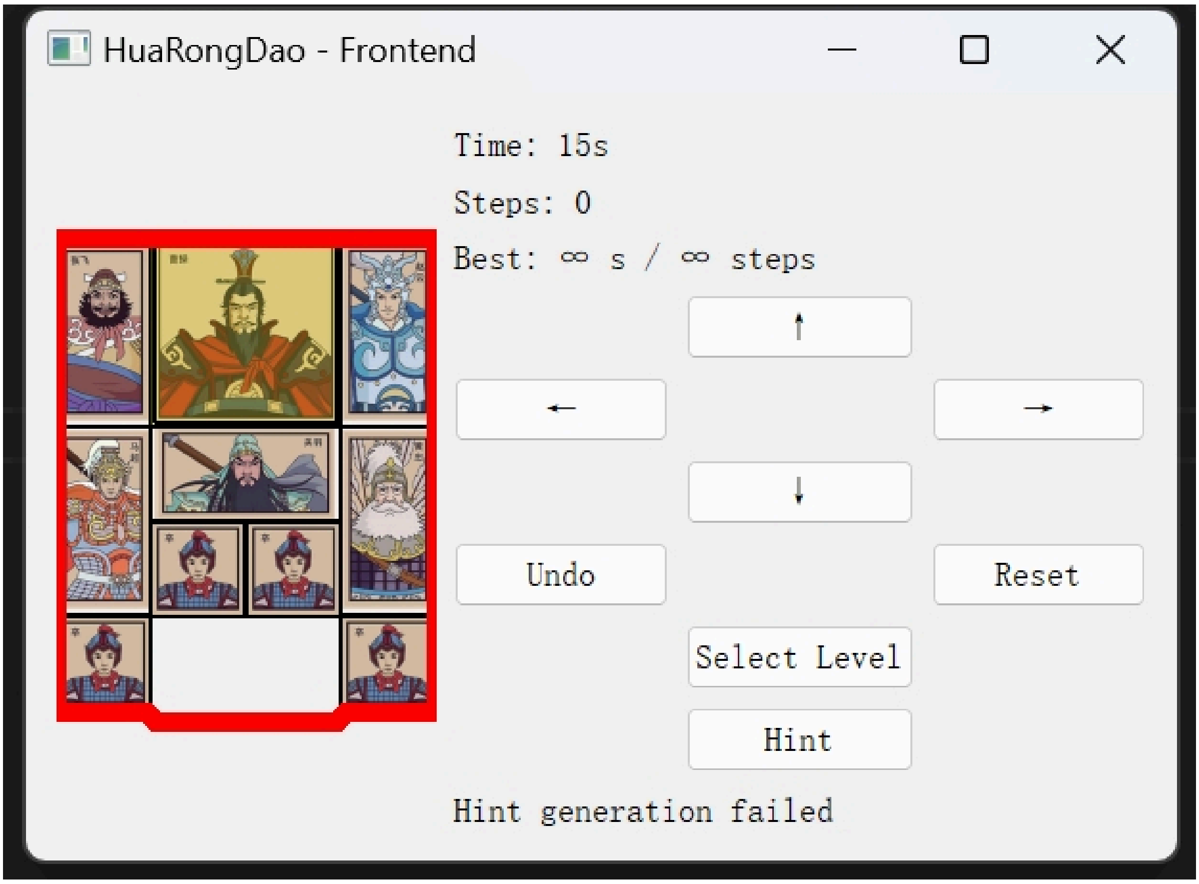
Hint



We specially developed a hint functionality:

On clicking "Hint" button, if it's solvable, you'll get a instruction visualize as white arrow.

If it's not solvable for our UPPAAL tool, it will complain about this within 3 seconds:



Operation instruction

1. At initial situation, the only valid operation is to select level. At the same time, all the other operations are invalid or inaccessible. Users do have the privilege to click them, but it won't mean anything, and it doesn't arise errors.

2. If any level is selected, all the operations are free to choose.

   To move a piece, you can just click on your ideal target piece on the board and it will turn to yellow. Then you can move it with arrow button that are set on the panel besides the board. All invalid moves will be checked and they will not affect the game.

3. The reset, undo and select level is always open and accessible.

4. To use our hint function, here's the tips:

   If hint is generated successfully, you'll notice a white arrow on next piece you should move.

   After 3 seconds, it disappear by itself. Or, you can click on that piece, the arrow will disappear, too.

   If hint fails immediately, there might be something wrong with your local UPPAAL file or running path.

   If hint executes for a long time(3 seconds) and fail, then the game is not solvable for UPPAAL. In this case, if you click Hint for a lot of times, our backend will handle all of them, which ends up in waiting for $3 * t_{Times\_that\_you\_click}$ seconds. So click it only when it's needed.