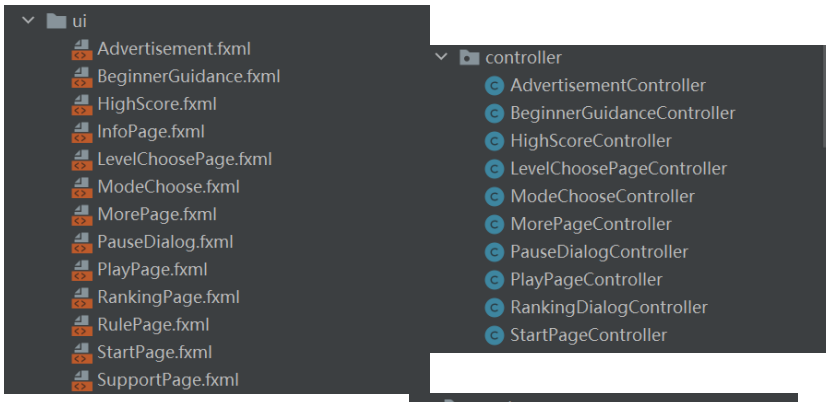
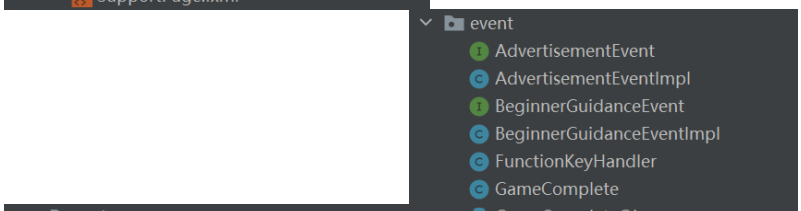
README

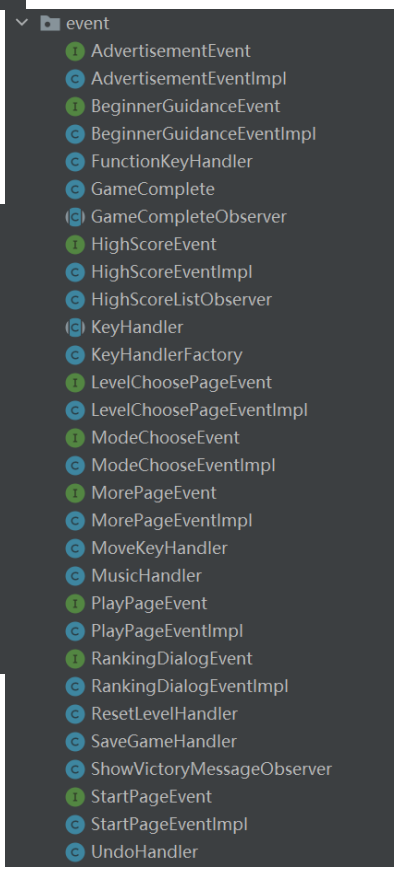
@author 20124917 -Gaole DAI

Maintenance

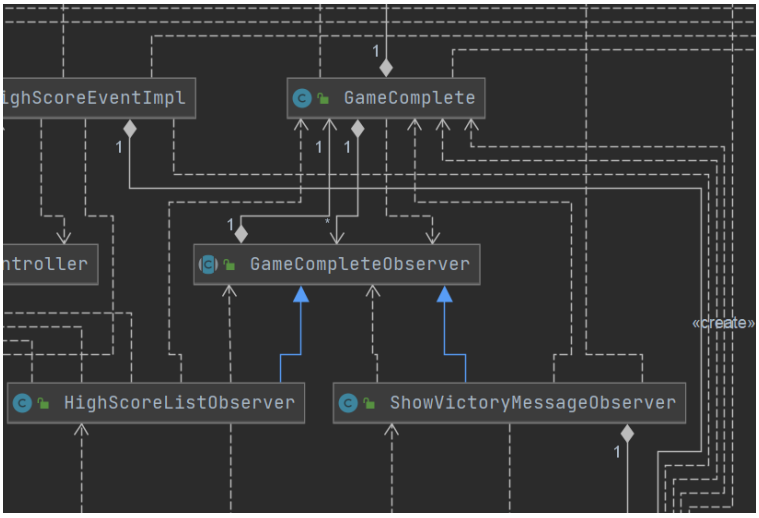
- By applying **MVC GUI design pattern**, divide the original classes into eight packages - **controller**, event, **model**, main, manager, service, util and **view** (including ui file in the resources).



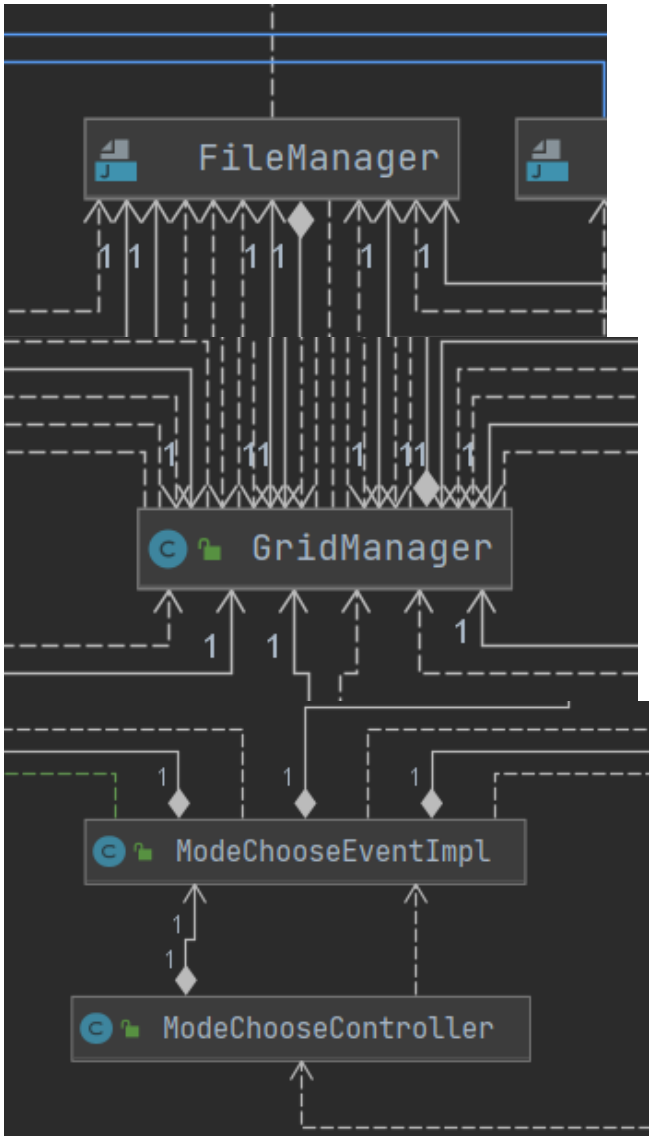




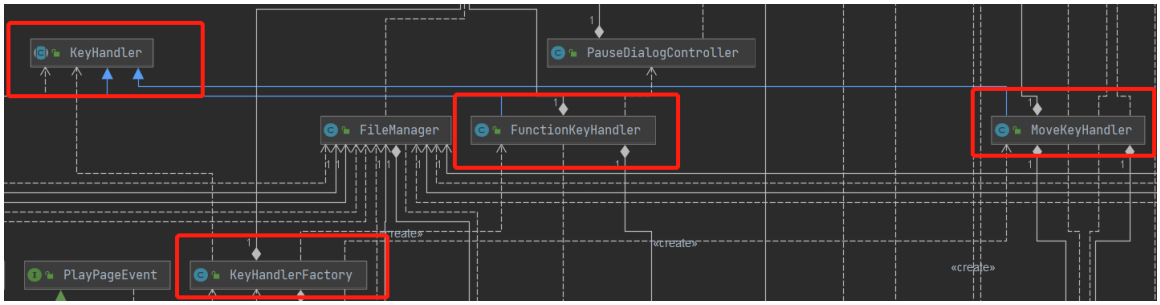
- Place all the controllers in the controller package, and by applying **Single Responsibility Principle**, refactoring the main class and gameEngine, GridManager and FileManager are responsible for all the works related to grid and file. Applying **Interface Segregation Principle** and for better maintenance afterwards, controller have its interface and implement class, all the controller event's interfaces and implement classes in the event package. GameGrid and Level classes manage the game data, in the model package. GameEngine provides the core service for the game. GameLogger and GameObject utilize the game.
- Applying **Observer Pattern** for decoupling. Notify the high score list observer and victory message observer to update every time the game completes, because the info of this round of game has been changed.



- Applying **Singleton Pattern** to ensure only one instance of a class is alive at run time. Only one GridManager and FileManager should take responsibility of the grid and file management and for every interface's implementation class, they should only have one object to handle the controller's method implementation.



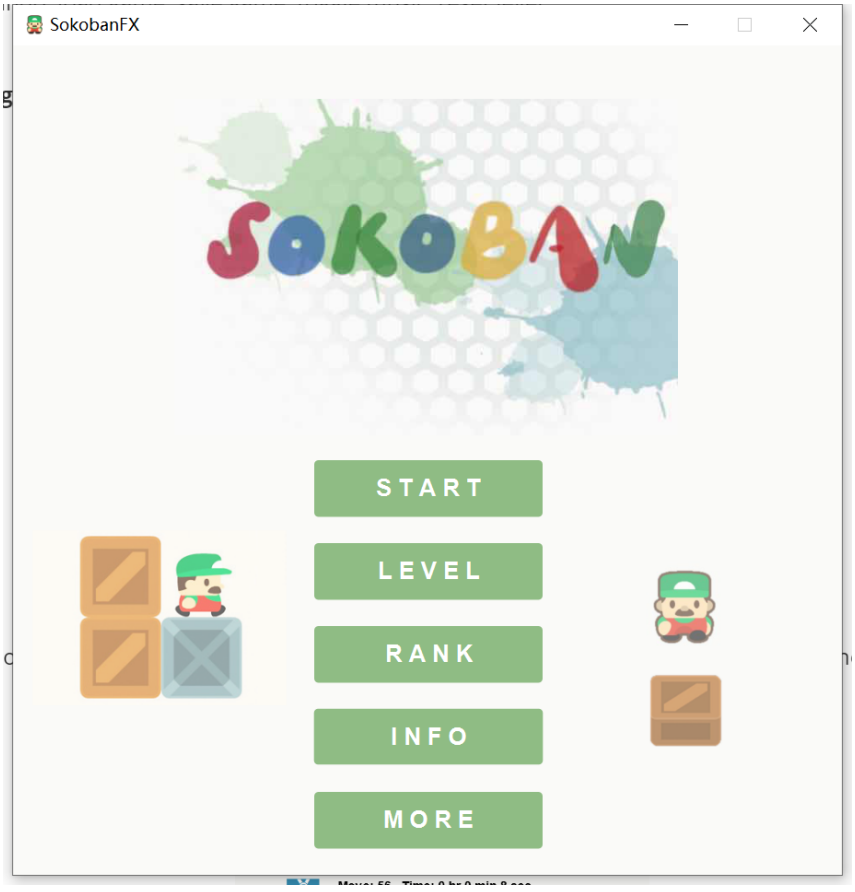
- Applying **Factory Pattern** to make the code more robust, less coupled and easy to extend.

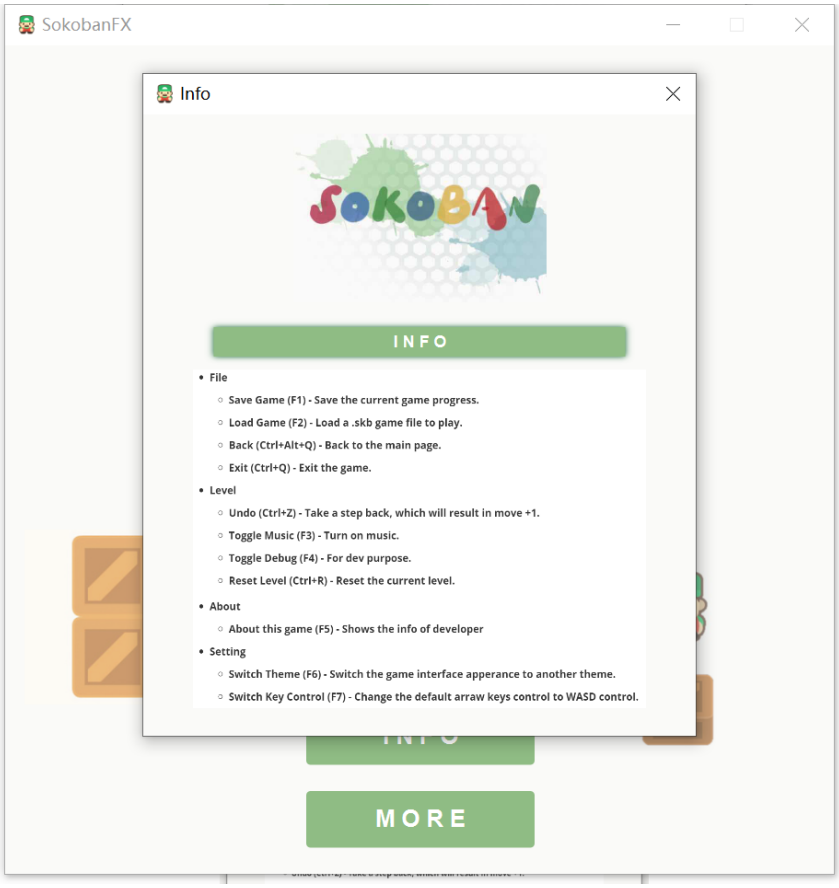


Extension

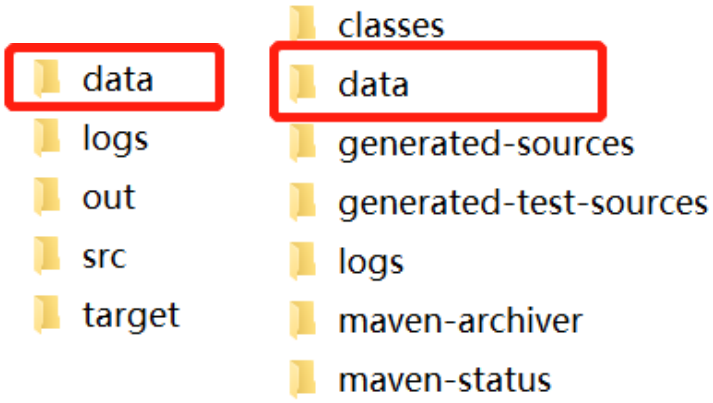
Basic Feature

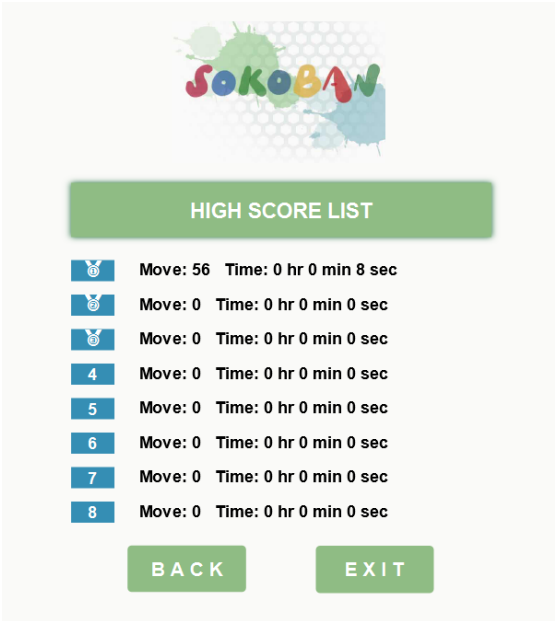
- Fix bugs, including undo, load game, save game, toggle music, reset level.
- Add **Junit tests**.
- Start page with **image** and **info** button.





- **Permanent** High Score List, stores in a file named **data**, at the same directory of the java runnable file.

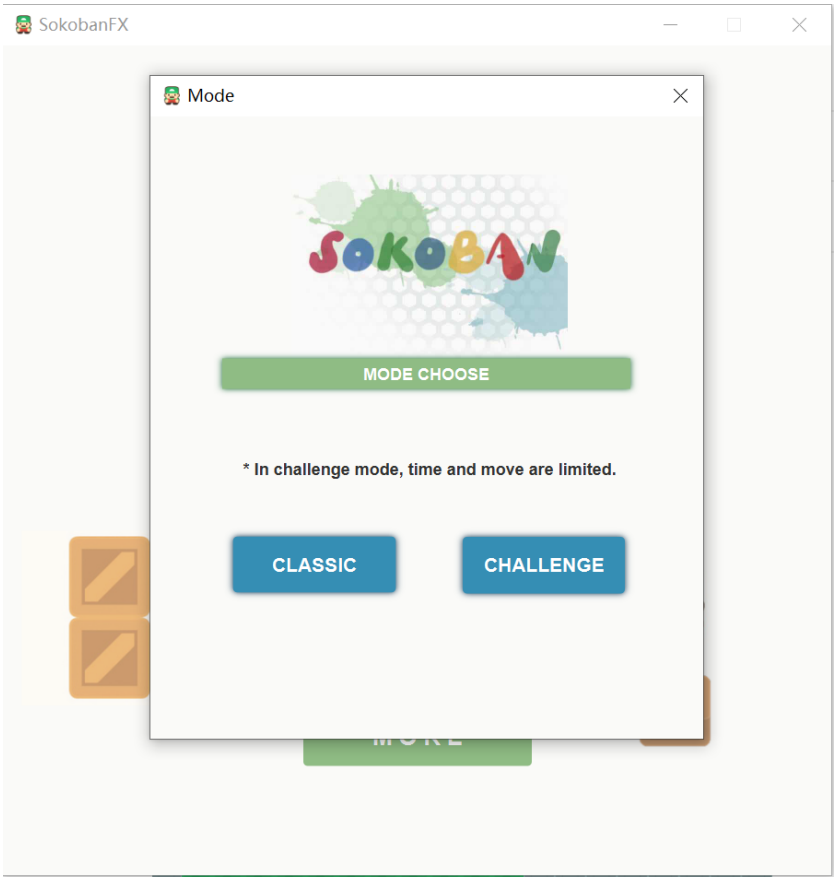




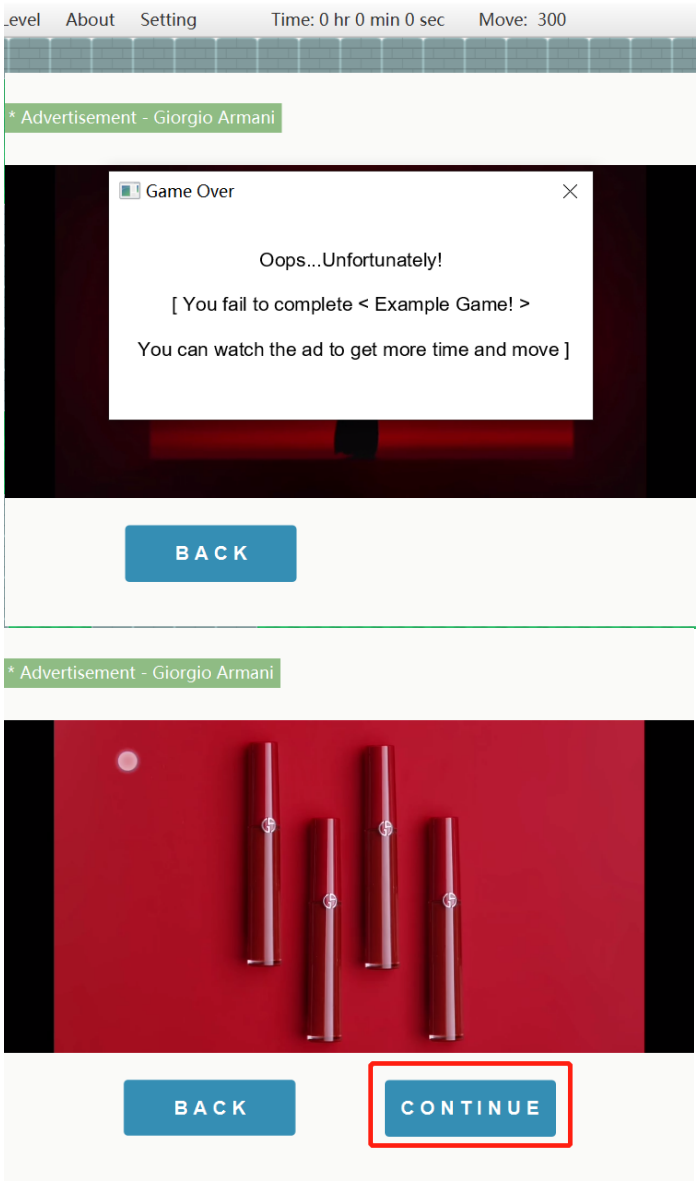
- Use **Maven** build files.

Additional Feature

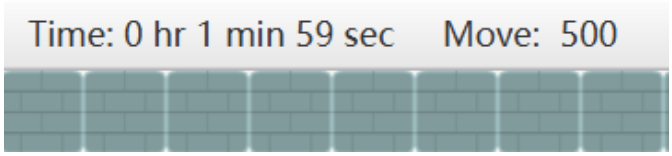
- ## Start Page
    - Add **Challenge Mode** and **Advertisement** life saving pattern to make it more like a real game.
        - Click Start Button -> **Challenge Mode** and Classic Mode. In challenge mode, player need to complete the game in 6 minutes and 1000 moves.



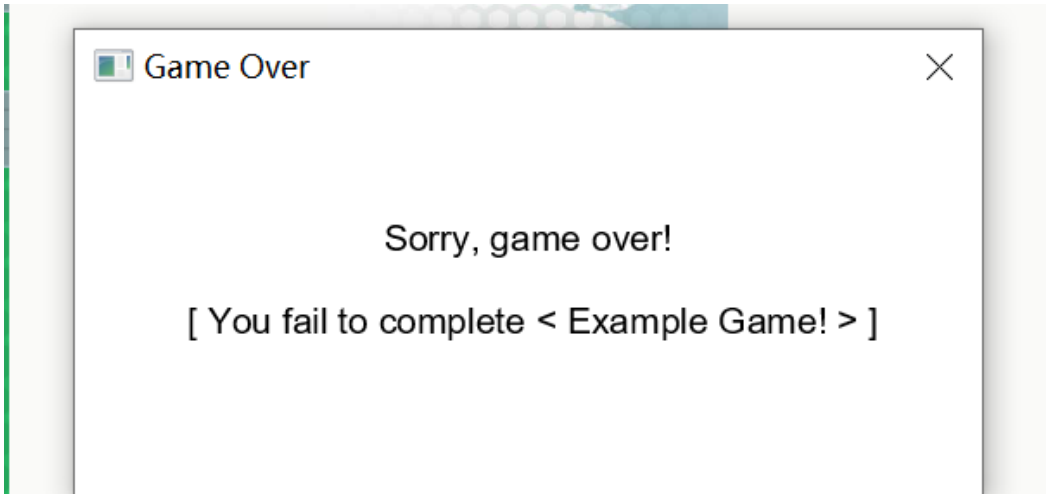
        - If players can't complete the game in limited time or moves, they have a chance of recovery -> watch advertisement.



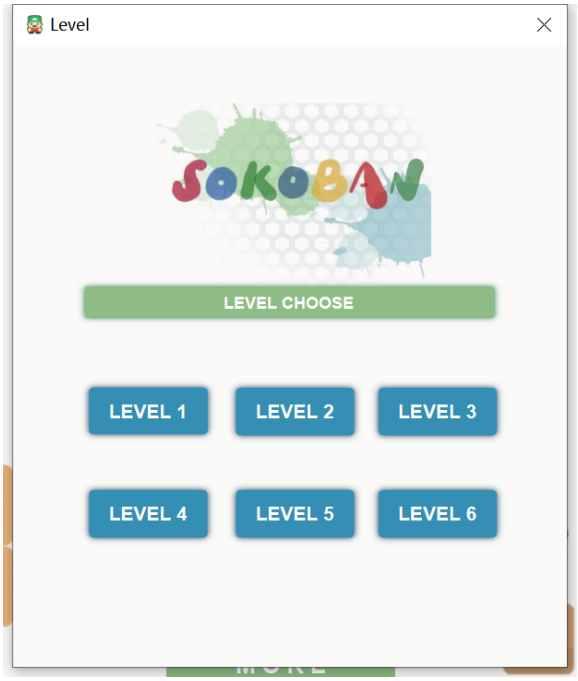
- If players click CONTINUE they will gain more time and move.



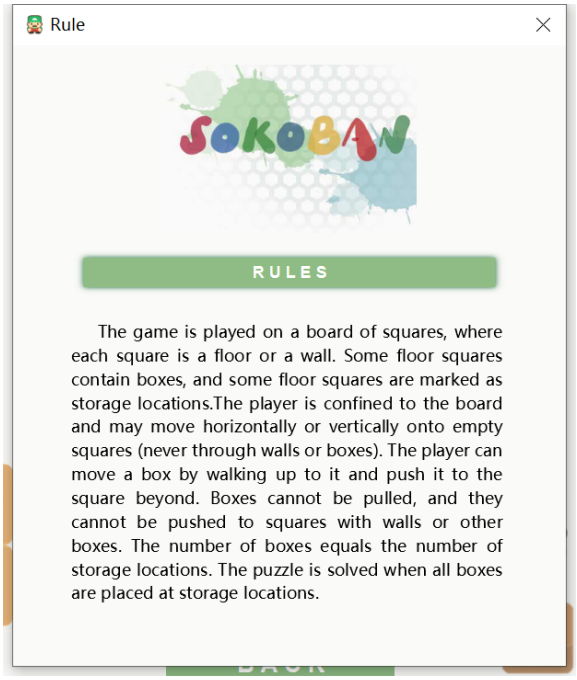
- But if they fail to complete this time, they fail.



- Add **Level Choose** function. Players could choose a specific level to try. Also, add **one more additional level**.

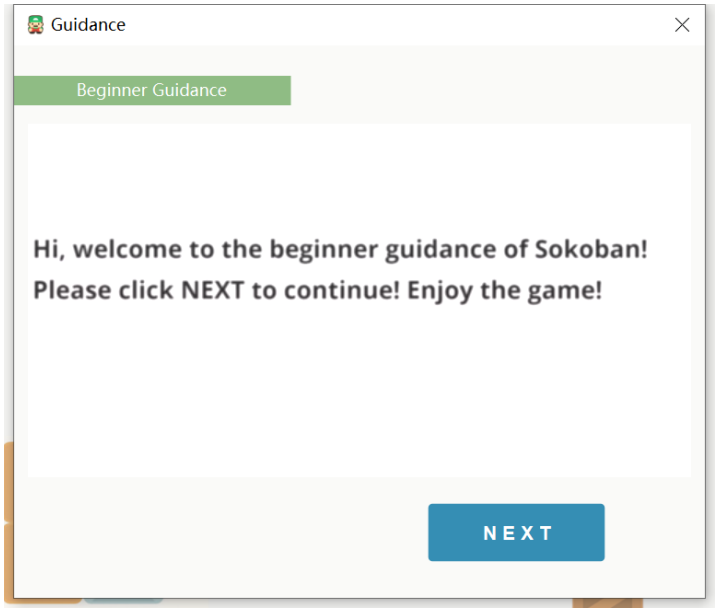


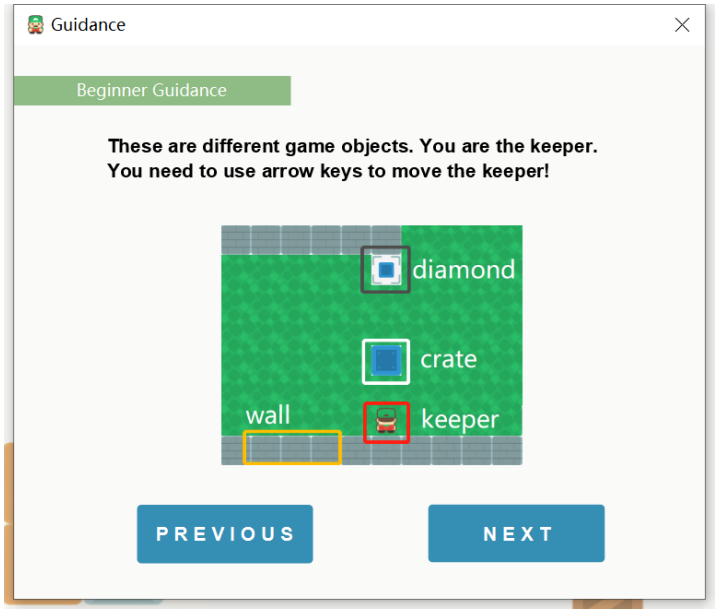
- Add **Ranking** button so players don't need to enter the game play page to **view current ranking**.
- Add **More** button.
  - Click Rules Button - Showing the rules of the Sokoban.



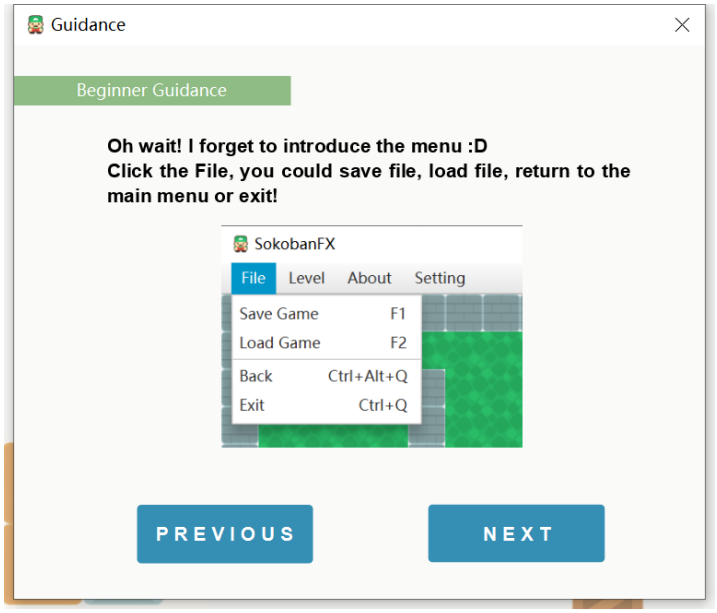
  - Click Support Button - Support the developer like a real game.

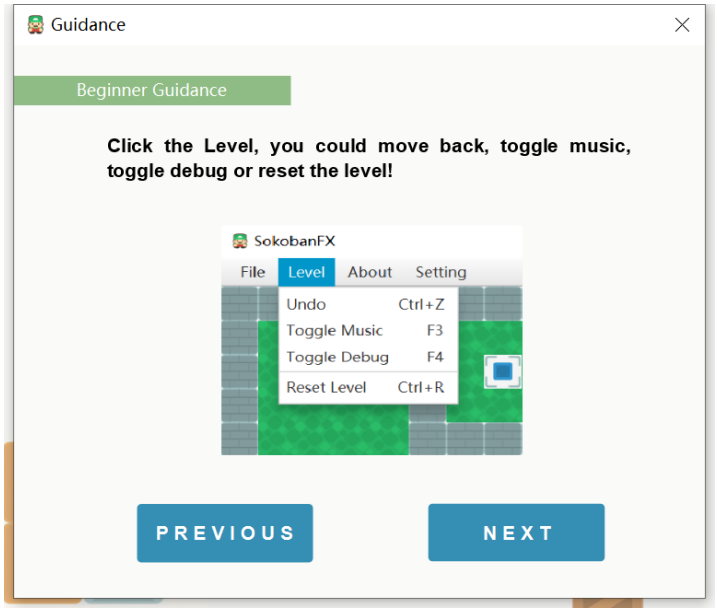
- Click Guidance Button - Interactive Beginner's guidance, showing all the game operations.



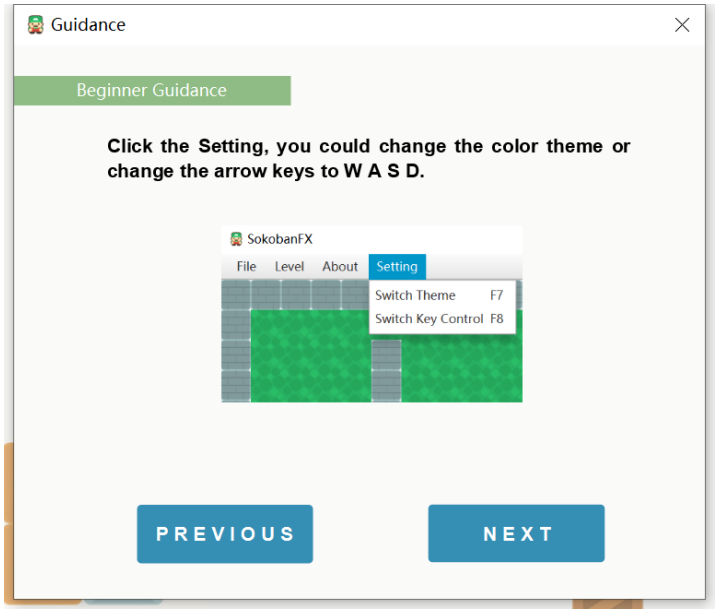


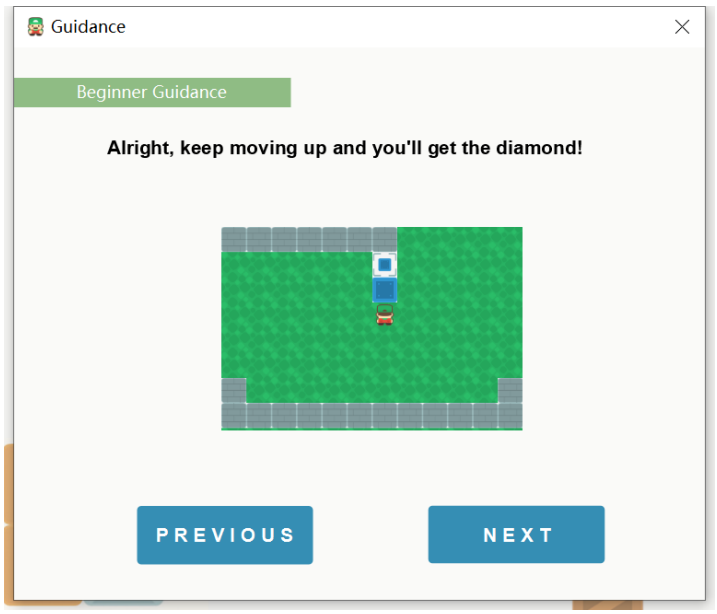

Guidance
Beginner Guidance
Remember! Your target is the diamond, so press UP to push the crate into the diamond.
PREVIOUS
NEXT

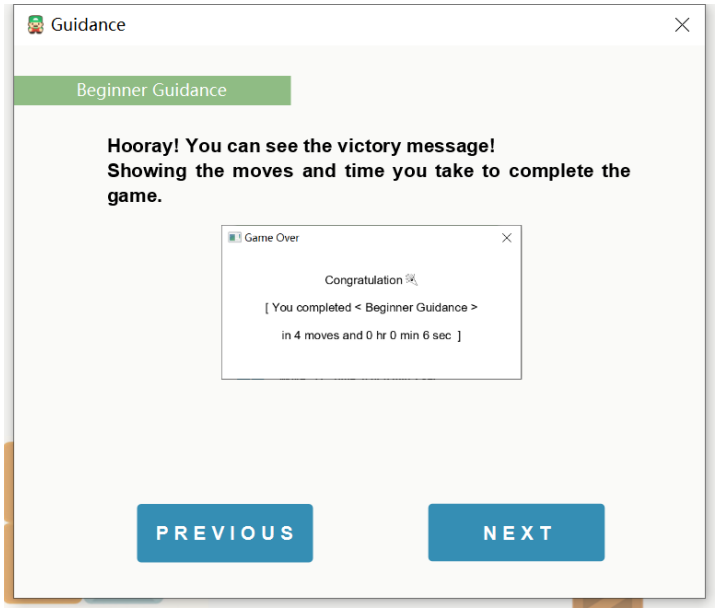

Guidance
Beginner Guidance
Oh wait! I forget to introduce the menu :D
Click the File, you could save file, load file, return to the main menu or exit!
SokobanFX
File
Level
About
Setting
Save Game F1
Load Game F2
Back Ctrl+Alt+Q
Exit Ctrl+Q
PREVIOUS
NEXT

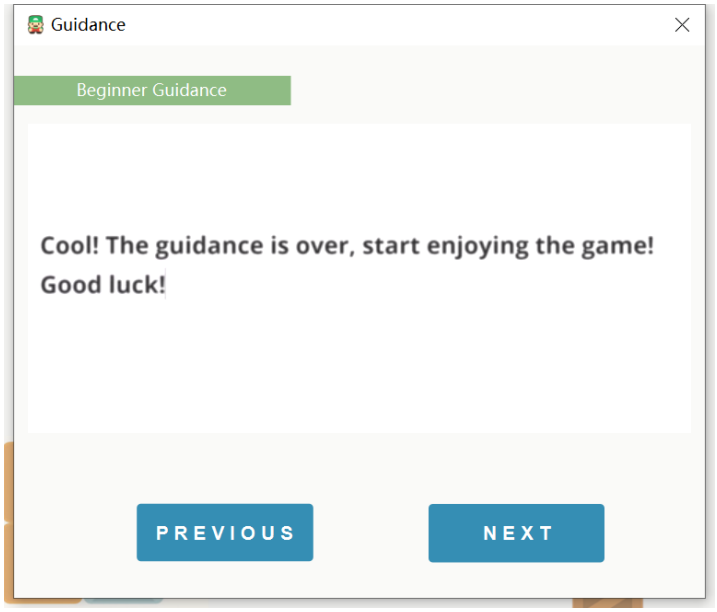

Guidance
Beginner Guidance
Click the Level, you could move back, toggle music, toggle debug or reset the level!
SokobanFX
File
Level
About
Setting
Undo Ctrl+Z
Toggle Music F3
Toggle Debug F4
Reset Level Ctrl+R
PREVIOUS
NEXT


Guidance
Beginner Guidance
Click the About this Game, you could view developer's info!
SokobanFX
File Level About Setting Tim
About This Game F6
PREVIOUS
NEXT

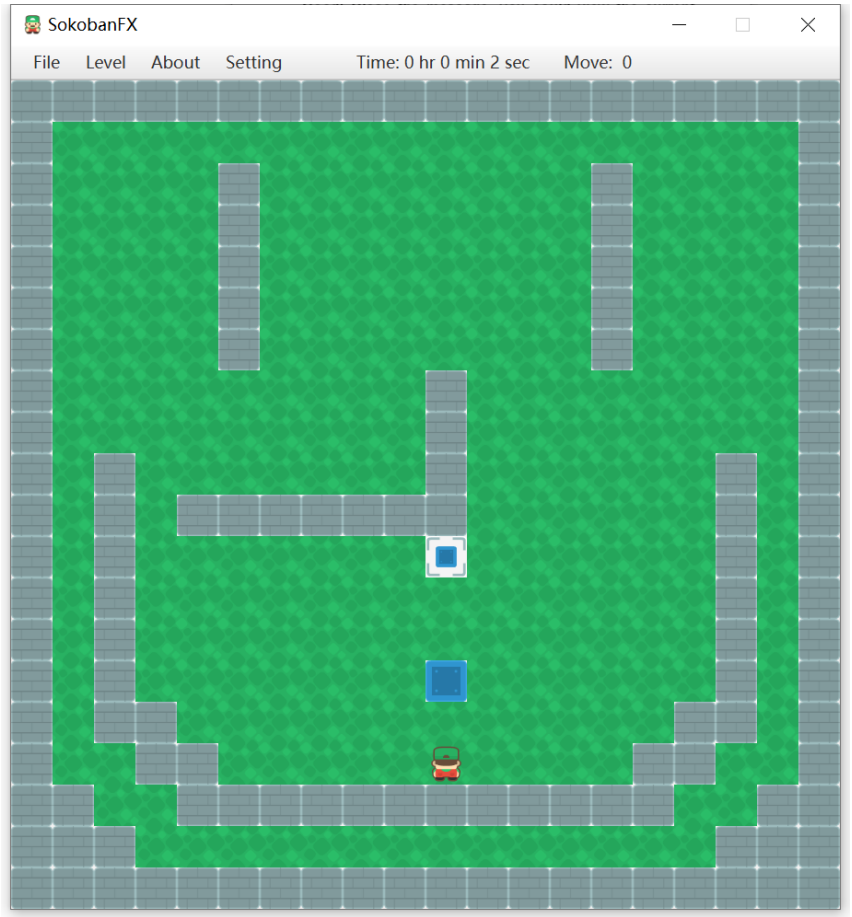

Guidance
Beginner Guidance
Click the Setting, you could change the color theme or change the arrow keys to W A S D.
SokobanFX
File Level About Setting
Switch Theme F7
Switch Key Control F8
PREVIOUS
NEXT


Guidance
Beginner Guidance
Alright, keep moving up and you'll get the diamond!
PREVIOUS
NEXT





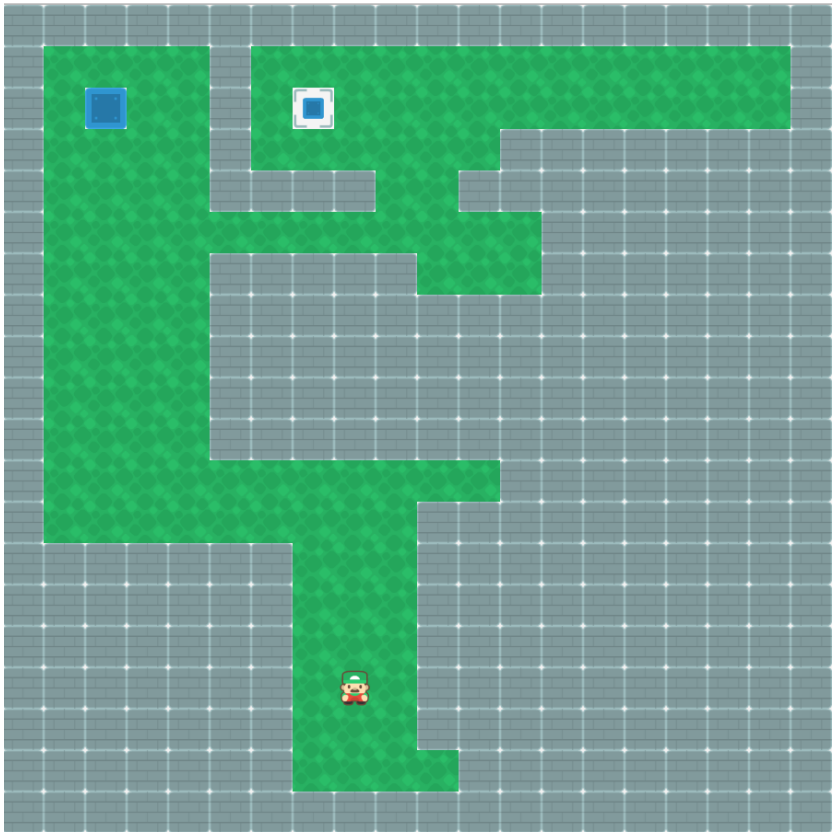
- Finally, Players could try to play the easy beginner map.



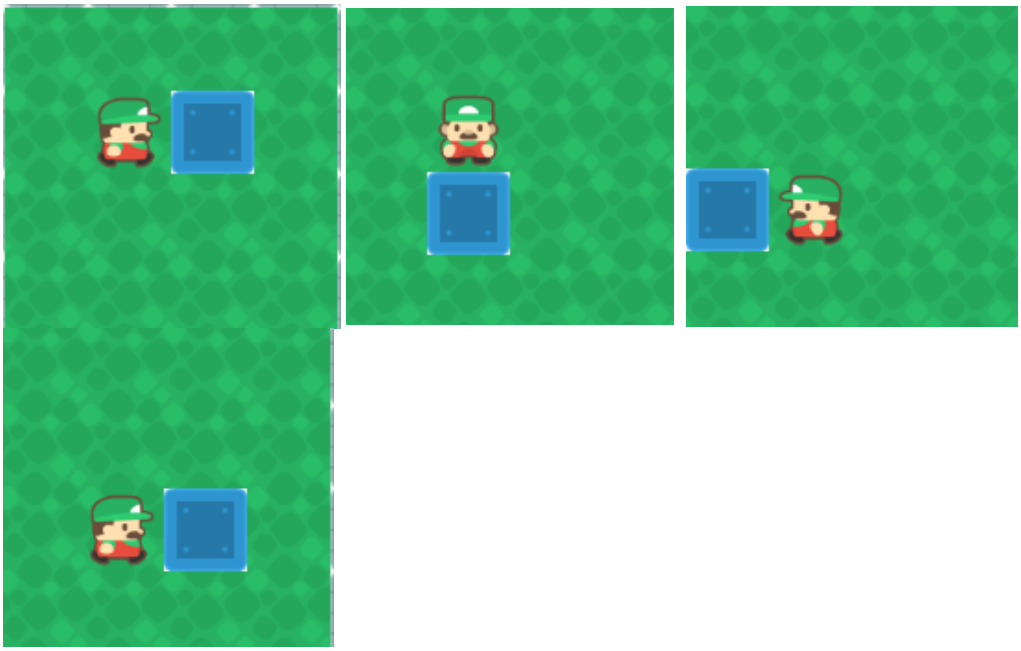
    - Click **Back** Button to return to the main menu, click **Exit** Button to exit the game.

- **Play Page**

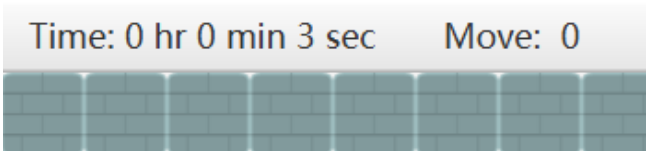
    - Modify the **appearance** of the game main interface.

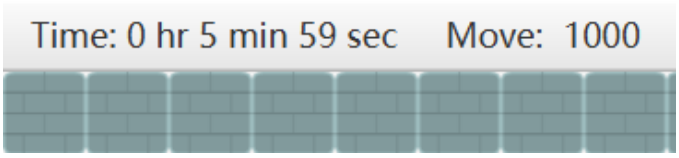


    - Keeper's appearance **changes** according to player's key arrow keys.

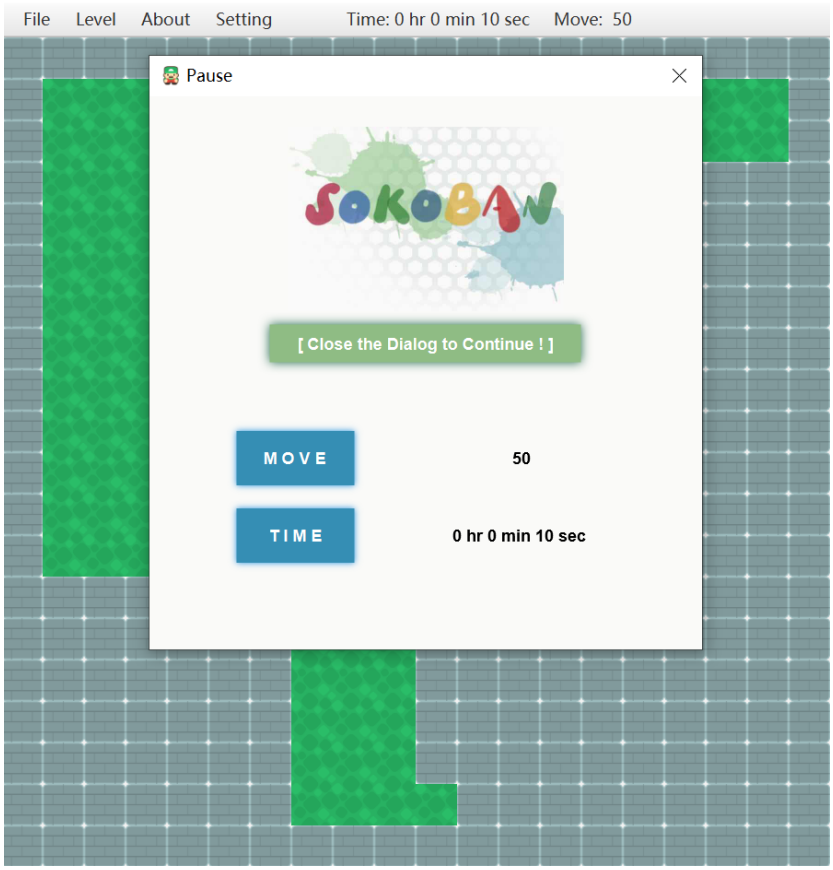


- Display **Real-Time Timer** and **Move Counter**, stores the time and move data in the .skb file together with the game data.

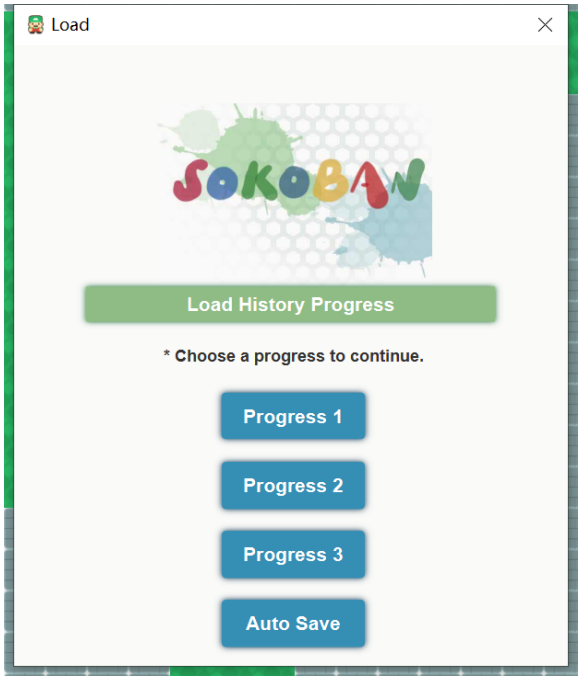


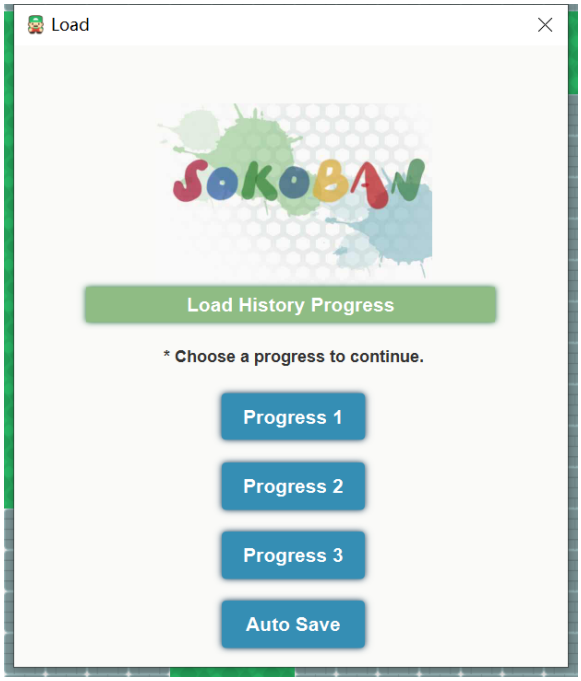


- **Pause** game function. Press **SPACE** key to pause the timer and count. The total time and count player has consumed will be showed on the dialog window. Game will **continue** if players close the window.

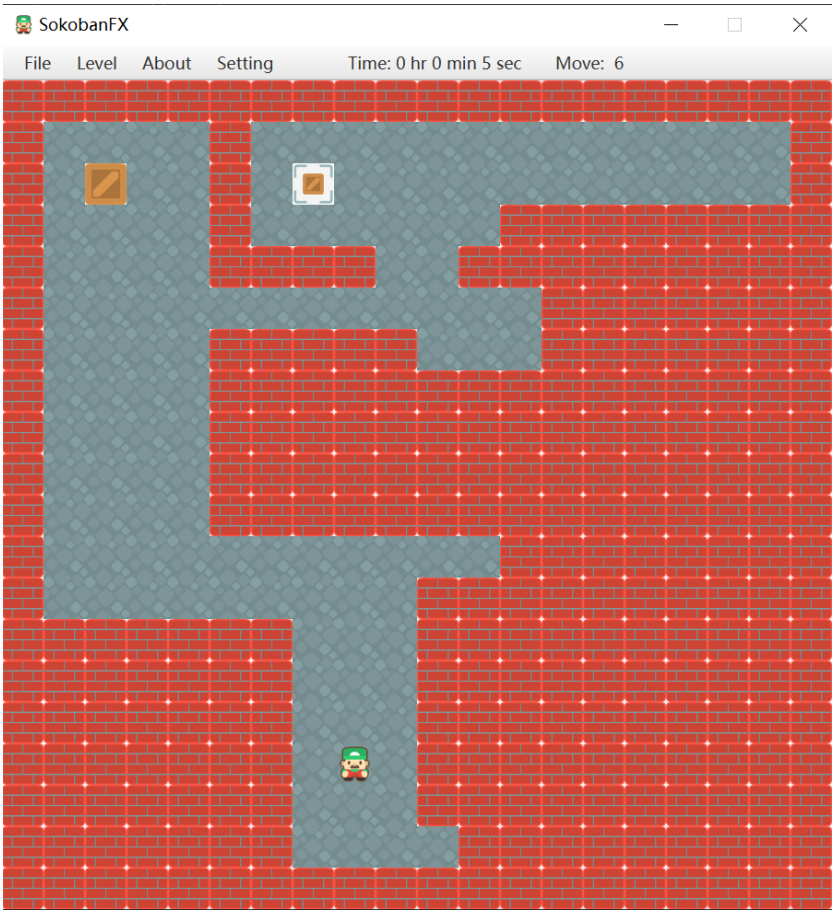


- Add **Auto Save** function, current game state will be saved to a file every **30** seconds.
- Change the **Save file** and **Load file** function. Without letting the player to choose the file path, they only need to choose a save position or a process to load.

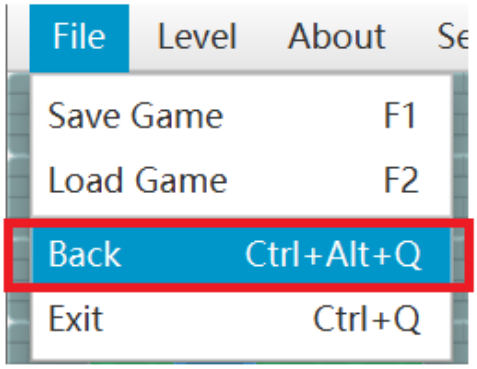




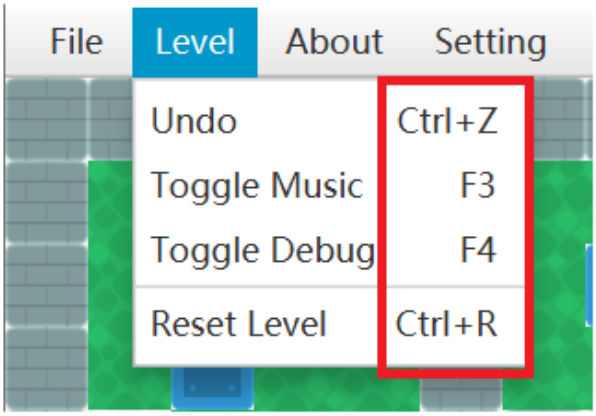
- Add **Switch Theme** function, **Switch Key Control** (from arrow keys to WASD) function.



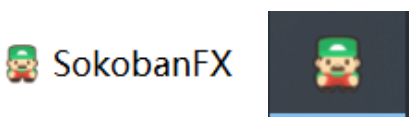
- Add **back** to main menu function in game start page.



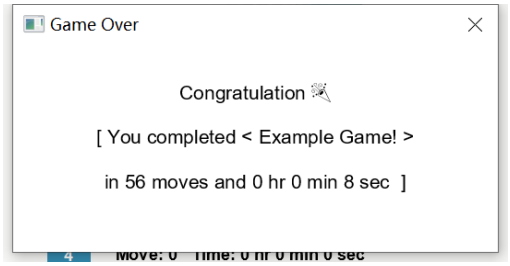
- Add **accelerate key** for all the menu Items.



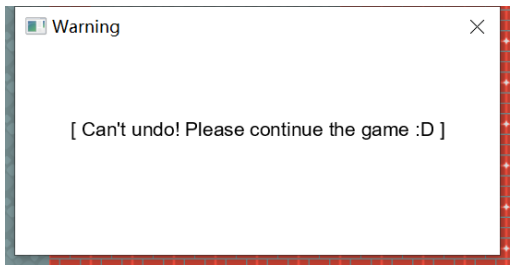
- Change the game Icon



- Modify the victory message



- Add prompt when player can't undo.



Junit Test

Class: Game Grid

Constructor: GameGrid (int columns, int rows)

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	testConstructor()	/	new GameGrid(3,3)	new GameGrid(3,3)	P	/	/

Method: translatePoint (Point sourceLocation, Point delta)

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	testTranslatePoint()	new Point(0, 2), new Point(1, 0)	new Point(1, 2)	new Point(1, 2)	P	/	/

Method: getTargetFromSource (Point source, Point delta)

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	testGetTargetFromSource()	new Point(1, 1), new Point(0,1)	GameObject.CRATE_ON_DIAMOND	GameObject.CRATE_ON_DIAMOND	P	/	/

Method: getGameObjectAt (int col, int row)

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	testGetGameObjectAt()	0, 2	GameObject.DIAMOND	GameObject.DIAMOND	P	/	/
Passed: 2020/12/5	testGetGameObjectAt()	0, 1	GameObject.CRATE	GameObject.CRATE	P	/	/
Passed: 2020/12/5	testGetGameObjectAtWhenExceptionThrow()	4, 4	ArrayIndexOutOfBoundsException	ArrayIndexOutOfBoundsException	P	/	/
Passed: 2020/12/5	testGetGameObjectAtWhenExceptionThrowDebugActive()	sampleGameInput, highScoreInput	ArrayIndexOutOfBoundsException	ArrayIndexOutOfBoundsException	P	/	/

Method: getGameObjectAt(Point p)

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	testTestGetGameObjectAt()	null	IllegalArgumentException	IllegalArgumentException	P	/	/

Method: putGameObjectAt(GameObject gameObject, int x, int y)

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	testPutGameObjectAt()	GameObject.CRATE_ON_DIAMOND, 3, 3	false	false	P	/	/
Passed: 2020/12/5	testPutGameObjectAt()	GameObject.WALL, 2, 2	true	true	P	/	/

Method: putGameObjectAt(GameObject gameObject, Point p)

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	testTestPutGameObjectAt()	GameObject.CRATE, new Point(2, 1)	true	true	P	/	/
Passed: 2020/12/5	testTestPutGameObjectAt()	GameObject.WALL, null	false	false	P	/	/

Method: toString()

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	testTestToString()	/	"===\n===\n===\n"	"===\n===\n===\n"	P	/	/
Passed: 2020/12/5	testTestPutGameObjectAt()	gameGrid.putGameObjectAt(GameObject.CRATE, 0, 0); gameGrid.putGameObjectAt(GameObject.DIAMOND, 1, 1);	"C==\n=D=\n===\n"	"C==\n=D=\n===\n"	P	/	/

Class: Level

Constructor: Level(String levelName, int levelIndex, List rawLevel)

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	testConstructor()	"Just this level", 1, rawLevel	new Level("Just this level", 1, rawLevel)	new Level("Just this level", 1, rawLevel)	P	/	/

Method: isComplete()

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	isComplete()	/	false	false	P	/	/

Method: getName()

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	getName()	/	"Just this level"	"Just this level"	P	/	/

Method: getIndex()

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	getIndex()	/	1	1	P	/	/

Method: getKeeperPosition()

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	getKeeperPosition()	new Point(1, 2), new Point(0, 1)	GameObject.CRATE	GameObject.CRATE	P	/	/

Method: getTargetObject(Point source, Point delta)

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	testGetTargetObjectWhenExceptionThrow()	new Point(1, 2), new Point(2, 1)	ArrayIndexOutOfBoundsException	ArrayIndexOutOfBoundsException	P	/	/
Passed: 2020/12/5	getTargetObject()	new Point(1, 2), new Point(0, 1)	GameObject.CRATE	GameObject.CRATE	P	/	/

Method: toStringObjectGrid()

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	toStringObjectGrid()	"WWWWWWWWWWWWWWWWWWWW\nWS C W\nWWWWWWWWWWWWWWWWWWWW\n"	"WWWWWWWWWWWWWWWWWWWW\nWS C W\nWWWWWWWWWWWWWWWWWWWW\n"	true	P	/	/

Method: toStringDiamondGrid()

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	toStringDiamondGrid()	/	"====================\n====D==========\n====================\n"	"====================\n====D==========\n====================\n"	P	/	/

Method: toStringCombGrid()

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	toStringCombGrid()	/	"WWWWWWWWWWWWWWWWWWWW\nWS CD W\nWWWWWWWWWWWWWWWWWWWW\n"	"WWWWWWWWWWWWWWWWWWWW\nWS CD W\nWWWWWWWWWWWWWWWWWWWW\n"	P	/	/

Class: GameEngine

Constructor: GameEngine(InputStream gameFileInput, InputStream highScoreInput)

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	testConstructor()	sampleGameInput, highScoreInput	new GameEngine(sampleGameInput, highScoreInput)	new GameEngine(sampleGameInput, highScoreInput)	P	/	/

Method: isDebugActive()

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	isDebugActive()	/	false	false	P	/	/

Method: setGameComplete(boolean gameComplete)

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	setGameComplete()	true	true	true	P	/	/
Passed: 2020/12/5	setGameComplete()	false	false	false	P	/	/

Method: getCurrentLevel()

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	getCurrentLevel()	/	level.get(0)	level.get(0)	P	/	/

Method: setCurrentLevel(Level currentLevel)

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	setCurrentLevel()	level.get(2)	level.get(2)	level.get(2)	P	/	/

Method: toggleDebug()

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	toggleDebug()	/	true	true	P	/	/

Method: getTimeCount()

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	getTimeCount()	/	20	20	P	/	/

Method: getMovesCount()

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	getMovesCount()	/	10	10	P	/	/

Method: setMovesCount(int movesCount)

Date	Test	Inputs	Expected Outcome	Test Outcome	P/F	Fail Reason(if fails)	How to Improve
Passed: 2020/12/5	setMovesCount()	2	2	2	P	/	/