Task ID	Description	Bonus Task Points	Author
1	Handling FileNotFoundException when loading a non-existing map	2	Ching Hang MAK
2	Automating demo procedure	5	Ching Hang MAK
3	Reporting deadlock when detecting any crate cannot be moved	2	Kin Ip MONG
4	Handling invalid maps when loading the maps	2	Shayan IMRAN
5	Implementing Undo feature	4	Shayan IMRAN
6	Supporting multi-player	5	Chung Wing CHU
7	Updating selected level after returning to the Level Select Pane from Game Playing pane	2	Wing Yin HO

1. Handling FileNotFoundException when loading a non-existing map

Task Description

The original implementation does not handle FileNotFoundException when loading a non-existent map. This task requires the program to catch the exception and popup an alert to its users and remove the corresponding level from the game. Test procedures include how the program may trigger this exception.

Related Techniques

Exception Handling, JavaFX Alert

Test Procedures

- Load the maps from the “maps” folder
- Select 01-easy.txt and start playing
- Delete 02-easy.txt
- Win 01-easy and go to the next level
- The game should popup a dialog warning that 02-easy is missing
- The game should then load 03-easy.txt

2. Automating demo procedure

Task Description

Automate the demonstration procedure with TestFX (https://github.com/TestFX/TestFX).

Related Techniques

Automated GUI testing for JavaFX

Test Procedures

- Run the written tests to emulate the demonstration procedure
- The tests should follow the demonstration specified in the requirement of PA2

3. Reporting deadlock when one crate cannot be moved

Task Description

In the original implementation, the program will only report deadlock when all the crates cannot be moved and the winning condition is not satisfied. This task requires the program to report deadlock when any of the crates cannot be moved before arriving at a Destination.

Related Techniques

JavaFX

Test Procedures

- Start a game
- Push one crate to a non-destination location where the crate cannot be further moved
- The game should report a deadlock

4. Handling invalid maps when loading the maps

Task Description

The original implementation of the project does not report any alert when loading invalid maps. NOTE: an invalid map can be any file that is not a valid map such as a txt file, whose first line is not a number. This task requires the program to popup an alert dialog if an invalid map is detected in the map folder, or if the map of the next level while playing is an invalid map. The invalid maps should be skipped when playing.

Related Techniques

Exception Handling, File IO, JavaFX

Test Procedures

- Prepare a folder containing invalid maps such as random text files
- Start the game and let the game load the folder
- Select an invalid map
- An alert should popup and warns that this is an invalid map
- The map should be removed from the list

Note: You may choose other ways to handle the invalid maps. We may also accept other proper implementations.

5. Implementing Undo feature

Task Description

The original implementation does not support undo when playing the game. This task requires the program to add a button in the game playing pane. Clicking the undo button should undo one move. Using this button, the user should be able to continuously undo the moves and go back to the initial state of the game.

Related Techniques

JavaFX, Collections

Testing Procedure

- Start a game
- Make some moves
- Click undo once and it should undo one move
- Make some moves
- Click undo until it goes back to the initial state of the game

6. Supporting two-players

Task Description

The original implementation of the game only supports single player. This task requires the program to support two players. Specifically, more than one player is allowed to play the game concurrently. The game play pane should contain two canvases, one for each player. One player uses wasd for input and the other player uses navigation arrow (D-pad) for input. The two players should not block each other. The first player reaching the winning condition wins the game.

Related Techniques

JavaFX, Multi-threading

Test Procedures

- Enable two-players feature
- Start the game and the game pane should show two canvases displaying two maps
- Two players can simultaneously play the game in their own canvas
- When a player reaches the winning condition, a dialog pops up on each canvas, notifying its player

7. Updating selected level after returning to the Level Select Pane from Game Playing pane

Task Description

In the given JAR, the selection of the map on the Level Select screen is not updated when the User goes from the Gameplay pane to the Level Select pane by any of the following means:

1. After clearing at least 2 levels (the first one being the map selected by the User on Level Select, and the other ones being the subsequent maps loaded only by clicking the “Next level” button), the User clicks “Return” on the pop-up.
2. After clearing at least 1 level and getting a “next level” map deadlocked, the User clicks “Return” on the pop-up.

This task requires that, after returning from a level that is not selected by the User on the Level Select screen, the ListView for level selection must highlight the actual level and update the Canvas for level preview and the “Play” button should use the actual level.

Related Techniques

JavaFX

Test Procedures

- On Level Select screen, choose /assets/maps/ as the Map Directory and choose “03-easy.txt” to play.
- Play the game to clear “03-easy.txt” level.
- When the level clear pop-up appears, choose “Next level” to go to “04-easy.txt” map.
- Play the game to clear “04-easy.txt” level.
- When the level clear pop-up appears, choose “Return”.
- Now the program should return to Level Select screen, while the level “04-easy.txt” is highlighted in the ListView and rendered on the previewing Canvas.
- Click on “Play” to play “04-easy.txt”. The program should go to the Gameplay screen where its user can play “04-easy.txt”.
- Play the game to clear this level again.
- When the level clear pop-up appears this time, choose “Next level” to go to “05-normal.txt” map.
- Play the game but make moves to get the game deadlocked.
- When the level deadlocked pop-up appears, choose “Return”.
- Now the program should return to Level Select screen, while the level “05-normal.txt” is highlighted in the ListView and rendered on the previewing Canvas.