Actor

mainMenu:
MainMenu

mainWindow:
MainWindow

LoadGameMenu:
LoadGameMenu

LoadGame:
LoadGame

Saved Files
[0-3]

GameSettings

InputManager

Select load game

SwitchPanels(LoadGame)

RemoveAllPanels()

Display Panel(loadMenu)

wait for save selection

Selects save slot

LoadObjects(saveFile)

Actor resumes control
when the game has
been loaded and
displayed.

Get Save Data

Return Save Data

thisBuild: BuildAssets

Note: The loaded Players and GameTiles have
any user-defined attributes stored within them.
They are not created again during load game.

Players are already configured with name,
location, color, turn order, and other attributes.

GameTiles are delivered with proper color, wall
information, and their individual address on the
grid.

Unpack
Players(4)
GameTiles(81)
NextTurn(player index)

BuildAssets(players,gameTiles, nextTurn)

gameBoard: GameBoard

Build (GameTile[][] gameTiles, int rows, int cols)

set
rows/cols

create grid = GameTile[rows][columns]

loop   x<rows

loop   y<cols

grid = add GameTile[row][column]

gameBoard

Set GameTiles[][]

GameSettings.Input Manager. Set Game Tiles

Set GameTiles[][]

Set Players[4]

Configure
InGameUI

game = GameController(inGameUIPanel, reloadedGameBoard, players, nextTurn, false);

game:
GameController

Set GameController(game)

Set Player Positions

GameSettings.MainWindow.ShowPanel(inGameUI)

MainWindow.ShowPanel(InGameUI)

Prepare for Next Turn

Note: This call to MainWindow, results in
MainWindow removing all current panels, and
displaying the InGameUI Panel.

The InGameUI Panel contains the gameboard
and will allow the players to proceed with input
for next turn.

The board is displayed in same condition as it
was saved, as previously configured, and play
can resume. The game has been loaded.