7 Undo

As a last exercise in combining testing and development, we will extend JPACMAN with a working undo button. By pressing undo while playing or after just having died, the user can undo his last move, as well as any monster moves that occurred after it, after which the game will enter the halted state, from where it can press undo again.

Since this is a testing course, your primary focus should be on the way in which you test this extension. In case you are running out of time, you are advised to specify test cases without working implementation, instead of an implementation without test cases.

Exercise 52 Provide a use case capturing the undo requirement, in the style of the doc/pacman-requirements.txt document. Think of possible situations that can occur, and describe the desired behavior.

Exercise 53 Summarize your design decisions in the style of the doc/pacman-design.txt document.

Exercise 54 The Move class implements the Command pattern, which was invented to facilitate undo. Write test cases for a Move.undo method, and extend the Move class accordingly.

Exercise 55 Its probably easiest to keep track of a stack of moves in the Game class. Write test cases that ensure proper pushing of moves made and popping of moves that must be undone, and provide the underlying implementation.

Exercise 56 Extend the JPACMAN state machine diagram, the corresponding test cases, and the Engine implementation to cater for the new undo event.

Exercise 57 Extend the JPACMAN UI with a new undo button which activates the corresponding functionality in the Engine class.

Exercise 58 Given your experience with adding Undo to JPACMAN, reflect on the JPACMANs design. Describe refactorings that you found necessary, or suggest improvements to the design and code base.