Kathryn Lutz
10/17/2018

# Chip's Challenge: Final Design Explanation

The final design incorporates singleton implementations of Chip & the map which are used throughout the files. The factory pattern is used to generate different kinds of cells that are placed throughout the board and inhibit the up, down, left, and right movements of Chip. And, the state pattern is used to keep track if Chip has gathered all of the pieces in both levels in order to complete the level at the end. The "escape" key can be used to exit the game at any time, the user controls Chip's movements with the up, down, left, and right keys, and there is a twist in both of the levels such that they add an extra challenge for the user in each of the levels. The game is started from "Main" which launches a pane and a screen where the game is played.

If I could do it over again, I would do a few things differently. For one, I would have added another implementation of the state pattern that would have states such as "canOpenDoor," "needsKey," and "needsFlipperToSwim." An additional state pattern implementation would allow for more Chip's Challenge elements to be used throughout the game and would also allow me to add more interesting game features. I would also reduce the minor amount of code duplication used between the "goUp()," "goDown()," "goLeft()," and "goRight()" functions. Regardless of these decisions, the implementation of the factory, singleton, and state methods allow for an interesting game.