Aidan Lewis

Chip’s Challenge

Discussion

My final design centers on the Game class. This class was responsible for starting the game and bringing together all of its necessary features. One of the most key classes is the Level class, which given a level number, returns the game grid for that level. For each level, there is a Grid object that is composed of many Cell objects. There is an overarching Cell interface that lays out the necessary functionality for each Cell. For example, each cell must be able to return whether it can be entered (an Open Cell can be entered, while a Wall Cell cannot be entered) and whether it is deadly (an Open Cell is not deadly, but a Water Cell is if the user cannot swim). Additional functionality each cell needed was a way to determine whether it was a winning space (only true for the Portal Cell), and whether or not a cell was on land (only not true for the Water Cell). Many different types of cells implement this Cell interface according to that cell’s necessary functionality. The most common cells are the Open Cell (essentially empty cells) and the Wall Cell. There are also cells for each of the items that can be collected (Chip Cell, Fish Cell, Key Cell), and cells that contain water (Water Cell). Finally, there are cells for doors (Door Cell), the gate in front of the finish portal (Portal Gate Cell), and a cell for actual portal (Portal Cell). Actually designing the level was a fairly manual process, since I simply add each cell where I want it to be for each level to set up the board. In addition, the player is initialized on the board. After the level has been generated, play can begin. Any time an arrow key is pressed, the player will attempt to move to the corresponding cell (as long as it is enterable). When the player goes onto spaces with a key, that color will be added to the Player objects key list. If the player goes onto a Door Cell, it will only be enterable if the Player object has the correct color key. When the player lands on a Chip Cell, the chip remaining count on the Grid object will be decremented. If the player lands on the Fish Cell, the Player object’s swim attribute will be flipped to true, meaning water will no longer be deadly. After there are zero chips remaining, the portal gate cell will be enterable, and portal can be reached. Any time the player lands on a space with an object, the cell is then replaced with an Open Cell. The scoreboard is updated every time the player moves. After the portal is reached, the board is reset and reloaded for the new level. The player’s state is tracked at all times to determine whether the player is dead (the game should end), on land, in water, or has won the game (in which case the next level should be loaded, or if the last level has been reached, the game has been beaten).

If I could start from scratch again, I would probably have had a separate Item interface along with several implemented items, such as keys, chips, fish, and doors. Currently, when the player lands on a space with a key, the Player object takes the key into its key list, and then the Key Cell is replaced by an Open Cell. I think better way to have implemented this would have been to make keys, chips, fish, and doors simply items that are on Open Cells. This way, when the player landed on this cell, it would take the item, leaving behind an Open Cell. So with this approach, I wouldn’t have had to constantly replace Key Cells, Door Cells, Fish Cells, and Chip Cells with Open Cells after they had been landed on. After all, the item is not part of the cell itself, it is merely something that happens to be in the cell.