Alex Zicaro

Software Engineering

Chip’s Challenge

My final design is as such: MapExplorer is used to implement and bring together all of the other classes. It calls on Chip to be drawn, LevelMap to draw the specific map that is needed and Enemy when needed for level two. I handle most of the cases in the startMoving method that includes the event handler. Almost all of the program reacts to the movement of Chip except for the initial setup. I found it easiest to implement a strategy for LevelMap since it contains strategies of different levels that will be drawn depending on what method of LevelMap is called. Observer pattern was used so the enemies can follow Singleton was used to have only one instance of LevelMap. These allowed me to flush out the design of the classes easier and helped me understand what each of the jobs of the specific classes were to be.

I would up changing my UML diagram from what I originally had because I had not planned having enemies to begin with as well as decided to handle the LevelMao class differently than I had originally planned. It is still not the perfect implementation as I would like it but it does what is intended and there are no objective flaws in my design that I see.

If I could restart and do something different I would implement the abstract class for LevelMap better. It works fine as is but can be confusing and contains some passing of data back and forth often which I could have avoided.

|  |  |
| --- | --- |
| **Pattern Name**: Singleton | |
| **Class Name** | **Roll in Pattern** |
| MapExplorer  (Calls on the singleton to get the data from it that is neeed) | Client |
| LevelMap  (Only one instance created, data is stored within) | Singleton |
| **Purpose:** MapExplorer calls LevelMap to create a specific map. That information is kept in LevelMap and is the only instance of LevelMap that exists | |

|  |  |
| --- | --- |
| **Pattern Name**: Strategy | |
| **Class Name** | **Roll in Pattern** |
| LevelMap  (Abstract class to be extended by strategies) | Strategy |
| LevelMap1  (Extends LevelMap) | StrategyA |
| LevelMap2  (Extends LevelMap) | StrategyB |
| **Purpose:** LevelMap has a specific strategy for drawing the level, LevelMap1 and LevelMap2 | |

|  |  |
| --- | --- |
| **Pattern Name:** Observer | |
| **Class Name** | **Roll in Pattern** |
| Enemy  (Moves when Chip is moved) | Observer |
| Chip  (Moves and notifies observers) | Observable |
| **Purpose:** An enemy needs to observe a chip so the enemy knows when to move, only when Chip moves | |