The Observer design pattern was used in our programming. Specifically we used it to change the value of the game timer. In Unity, Text Scenes can have a script attached to it, and be set to watch the value of a public variable in the script which changes what is displayed accordingly. This same pattern was applied to display the lives as well. Unity allows for your program to be compiled to work on a variety of operating systems, such as Linux, Mac OS, and Windows. For this project all group members are using Windows for programing and testing, but if compiled for a different operating system the program should act exactly the same. This is an example of the Bridge design pattern. We used the Wrapper design pattern. The Unity engine acts as a wrapper. It takes the C# classes our group wrote, and connects them with the scenes designed inside the engine user interface. Scenes can only be created in the Unity engine and without it would be uninterpretable and unable to interact with our C# code. In order to keep track of which room the player is on and then display it, the State design pattern was used. Players start on the first room, and when they reach the top and enter the second room the room state is changed. When the room state is changed the scene shown on screen is updated.