Our project 3 included a few design patterns in each category. We included the builder creational design pattern, which we used for our map building and character building. The map is built mostly in the globalvars class and the player's character is built in the player class. We also used singleton in the creational design pattern. We have different classes handling different objects in our project, which the class can be later referred to by other classes. One example would be the player’s character, which is originally defined in the player class, and it is being referred to in the controller class. The is only defined once and has only one instance. For structural design pattern, we included a proxy. The wall and floor are objects declared as blocks and could be referred to in other classes, mostly in the collision class. We used three behavioural design patterns, one of them is chain of responsibility. The functions in our code would mostly only handle one job and will pass the next job to the following function. An example would be character jumping and the jumping event listener are split into two functions. The second behavioural design pattern is command. We set the character to only move whenever the correct button is pressed. And the last behavioural design pattern is state. The character status is stored as different states, which one of the examples being the health of the player’s character. The player character will have three lives, and when it touches an enemy, it reduces by one. When the character falls between the gap, it reduces to zero.