Question 6:
To implement the command design pattern to undo the eaten pellets, I used the unity input manager and created a new C# file to handle the undo process.
Firstly I implemented the “Game Editor” into the input manager to handle the data package needed for the command pattern.
Next I created a new C# file called “Object Poll & Undo” to handle the required processes. I then called the unity engine input handler and initialized the required fields to the script (7 different pellet gameObjects).
In between, I also linked the “Player Controller” script to handle the deletion of pellets (by moving the pellet underneath the map, which is the same as the ghost gameObjects) when they are consumed. To actually undo the consumption process, I created several lines of conditions and inputs to listen for the player’s actions. When the player presses the F key, the data package will be received by the script and the pellet game objects will be snapped back into place, as I have hard programmed in the specific coordinates the pellets needed to go.

Question 7: (Scratch build unity project, no in-game implementation)
Game management system I would pick to implement into a game of pac man is a “Powered up” manager.
The manager works by using a singleton design pattern, two different observer patterns and a command design pattern. The manager is first created through a singleton pattern, where it checks if there is a separate instance of the manager running. The manager then creates observer subjects on the power pellets, and for every frame they keep check on the state of the pellets. And in between frames (update function), if a subject has detected that a power pellet has disappeared. It would call a method to start a timer where another subject from the second observer would take over and watch over the delta time float object.
The subject from the second observer is also responsible for sending timer-related information to the rendering engine of the game architecture. When delta time reaches a certain point. The second observer pattern would send a data package through a command pattern to the rendering component (and later the AI component) of the architecture to update the weakened blue sprite of the ghost to the flashing blue and white variant.
Once the designated time limit for the power pellet has been reached, the second observer pattern would repeat the aforementioned steps to tell the command pattern to change the sprites of the ghost back to their normal state and AI.