My design essentially had one interface called IDisplay, which served as an interface for all other display classes, such as KeyDisplay and ChipDisplay (except for LevelDisplay). Level Display was the overall display class, and was responsible for creating and using all the other displays. Key served as an interface for all key types (red, blue, etc.) and KeyWall served as an interface for all key wall type (red, blue, etc.). Level was an interface for each individual level (in this case just Level1 and Level2), and GameSim was the overall controller of the games features (responsible for detecting key presses and such, and calling methods that then used other classes and methods to perform various actions). Anything that changed, such as keys and chips being picked up and the inventory updating happened in a variety of different classes, but they all originated from some kind of call in GameSim while the game was running. Chip, ChipItem, Portal, and Inventory were the only remaining classes. They didn’t have interfaces, as they needed their own functionalities different from most of the rest of the classes. The basics for forbidding or allowing movement across the screen was that there was a boolean array the same size as the amount of cells/square in the window (25x25 array) and anywhere there was a floor tile or some tile chip could step on (such as a key) was a false value, while a wall tile or tile that was supposed to block chip was a true value. For blocks with values that could change (like a key wall), within game sim was a method to check if chip had the prerequisites to step on those tiles. If he did or did not, the value for that tile would change to false or true respectively.