1. Description of Control Flow

Inside each of Player’s doSomething function, if the player is in the walking state, it will call a function called activate(). This function checks if the player is on a square or the same square as a Baddie. This function checks if the Player “landed” (its ticksToMove variable is 0). Then for each Baddie and square that it is on the same spot as, the player calls that Actor’s setActorToActivateOn function passing in a pointer to itself, and a bool to indicated whether it landed on it or not. The setActorToActivateOn function then saves a pointer to that player. Then, inside CoinSquare’s doSomething function, it checks to see if there was a player who activated it and whether it landed on the square or not. If both are true, it passes the Player pointer to its own updateCoins function to update the Player’s coins accordingly.

Similar logic was used for interactions between other objects that were not players. The setActorToActivateOn function was overloaded so that it could store an Actor pointer instead of a Player pointer when needed.

1. Unfinished functionality

I am unaware of any bugs or missing functionality my program has

1. List of Assumptions

I noticed that for some boards like board 2, there was a general flow of which direction Players usually move (usually they can’t walk onto a Directional Square from the direction that the square is pointing at). However, after teleporting, the players are allowed to move in a random direction which is sometimes in the opposite direction of the general flow of the board. So sometimes after teleporting, my players would walk onto a Square from the opposite direction and immediately turn around. I assumed this was okay.