**3.? Creating a Game:**

The “Creating a Game” integration test will be testing the integration between a handful of methods and classes, namely:

**View:**

CreateGameView, LobbyView, GameView, JoinGameView

**Model:**

HanabiGame

**Controller:**

ServerComm, AIController, HanabiController, JSONParser

The initial view should be in the CreateGame view while the HanabiController has a connection to a mock server. The user should enter the parameters: (4, 60, “abc123”, false) (see Section 3.2 of the DD). Once user has entered the relevant information in the CreateGame view, the “create a game” button in the view is clicked. It should call the listener in the sendCreate(4, 60, “abc123”, false) method in the ServerComm class, which should intern send a message to the server to create a new game. Once the server returns a messege confirming that a new game has been created the createGame() method in HanabiController should be called. The view should be updated to show the LobbyView view (see Section 3.3 of the DD).

After a game is created the view for the game creator being displayed should be the LobbyView. The view of the final person wanting to join the game should be the JoinGame view. The user should then enter parameters: (1, “game1”, “abc124”), and click the “join” button (see Section 3.2 of DD). It should call the listener in the sendJoin(1, “game1”, “abc124”) method in the ServerComm class, which should intern send a message to the serve to notify it that this player would like to join. Once the last player has joined the game the server should send a message to the ServerComm class. The ServerComm class should then notify the HanabiController that someone has joined the game using the joinGame() method (see Section 3.3 of DD). In the model the startGame() method in HanabiGame class is called and each player receives their hand (player 1: 1R, 2Y, 3G, 4B, 5W, player 2: 1R, 1R, 3G, 3B, 5W, player 3: 1R, 1Y, 1G, 1B, 1W, player 4: 4R, 5Y, 2G, 4G, 2G ) (see Section 3.1 of DD). The fuse and information tokens should be initialized to 3 and 8 respectively. The discard pile, fireworks piles, and log should be empty. The view is updated to show the GameView view and start with player 1’s turn. Checks after a game has been started should ensure that the hands stored in Player hand instance variable match the artificially created hands above.