Manual Tests

- Downloading Game - test that the download from the website downloads the correct version of the game
- Loading Game - loading game startup from the jar file
- Chef movement - The chef moves with the correct keys
- Chef Collision - The chef collides when interacting with an object
- Buttons - buttons follow through to the correct menus
    - Main menu
        - Start - allows the user to start playing the game
        - Load Game - loads the game screen
        - Tutorial - opens a tutorial page to help explain the game basics
        - Settings - opens a settings page
        - Exit to desktop - quits game
        - Mode selection - the mode of the game reflects the button picked, and only one can be selected at a time
        - Customer - chooses the number of customers that need to be served in order to complete scenario mode, ranges from 1 - 5
        - Difficulty - chooses the difficulty level of the game, applies to both game modes
    - Gameplay
        - Home - saves the current state of the game and takes you back to the main menu screen
        - Star - powerup is applied
        - Shopping trolly - correctly allows players to buy items
        - Grab Item - allows chef to pick up items
        - Place Item - allows chef to put down items
        - Chop - if applicable changes item status
        - Bake - if applicable changes item status
        - Cook - if applicable changes item status
        - Make _ - if applicable asseblems dish
        - Submit Order - if applicable submits completed dish
    - Tutorial
        - Done - takes the user back to the home screen
    - Settings
        - Full Screen - a check mark allowing the user to turn on or off the fullscreen option
        - Back - this button takes the user back to the home screen
    - End Game Screen
        - Return to Home Screen - takes user back to the home screen
- Assets - All assets rendered on the screen correctly

Manual Tests

In order to compensate for the classes and functionality that couldn’t be tested using unit tests, we used the method of manual tests. This includes UI elements, and functionalities that require user input. For classes that required user input such as Chef, we would run the game and press the input keys W, A, S, and D, and would monitor what the game is displaying to ensure the class is handling user input correctly. To ensure collision detection was functional again we would run our game and control our chef to come into contact with our collison borders, we observed to make sure our chef is unable to make it pass these borders, therefore we know collision detection is functional. The UI elements can be tested by running the game and checking that each element is displayed correctly.
Navigation and gameplay buttons are tested through observation, we make sure each navigation button is functional and performed correctly, the gameplay buttons are tested by having multiple playthroughs of the game. We made sure each button is displayed at the right position and functions correctly. Since these elements couldn’t be automatedly tested, we were very meticulous when carrying out these manual tests, and would repeat them often to ensure all problems and bugs are discovered and fixed.