The Manual Testing Done By Class

CameraManager
To test the functionality of the camera the game was loaded up and each of the functions had their own tests.
Public CameraManager() was tested when the game launched, the player has a starting location and the camera is centred on this person. It was able to pass this test.
update(), when the player moves, the camera follows them, this was tested by moving the camera in all directions and seeing if the camera moved. It was able to pass this test.
boundry_Check(), to check if this stops the player from going out of bounds or off the screen, the player walked around the entire map to ensure that there was not any place that could send the player out of bounds.
catchup(), while a precise test is not available, the general motion of whether the camera follows the player was tested by moving the player in each direction and checking whether they were followed by the camera.

DayCycleManager
nextDay() was tested by sleeping, and then seeing if the day was incremented in the top of the screen indicating that a day had passed, then, on day 7, it was checked if it would pass to the end screen. This test was passed.
render() was tested as the time and day were on the top right of the screen, various activities were performed that would increase the time and day, and these were recorded and compared to the expected values to see if the function was working correctly.

End Screen
The end screen was tested for functionality by first making a list of all the elements that should be drawn to the screen, this included different scenarios based on the score and what achievements had been unlocked. All of these scenarios were tested by achieving them, and then recording what the end screen functionality was like. Initially these tests were not passed as the tables were not being drawn correctly, and were overlapping. Iterating over this test with changes allowed for the final correct functionality of the end screen to be produced.

EnergyMeter
The energy meter is drawn on the top of the screen, when performing an activity there is a set amount of energy that is lost. This was used to calculate how the energy bar should look after spending a certain amount of energy, and then when performed the theoretical value was compared to what was drawn to the screen.
This was testing the calculateBar, render, setup and energymeter functions

Game
The game was tested by launching the game, it was recorded that music should begin playing, and we had an independent recording of the music that we used to compare when the game launched. The game played the same audio so it passed this test.
As well, the menu screen should be drawn, and we had the plan for the menu screen to compare to, making sure that it was properly drawn.

GameMap
The game map has various functions that were tested,
The GameMap(), render(), and getSpawnPoint() functions were tested by loading the game, and if the player spawned into a visible map then it was deemed to have passed. It was able to do this.

The update() function could only be tested in a roundabout way, as we did not have a way of manually verifying the time that it was updating. However the minted functionality of updating

the map state was tested and demonstrated by moving the character around the map and the map moving and sprites moving.

The isInArea(), insideCheck(), and toggleLayerVisibility() work together and were tested by walking through the different buildings, and noting whether or not they would toggle visibility. These tests were completed multiple times to ensure consistent results and they were able to pass every time.

GameScreen
The game screen initialises various components and draws them to the screen.
The GameScreen() function was tested by loading the game, and then comparing predicted starting positions and character sprites to what was being drawn. It was able to pass this test.

The show() function was tested when the game launched, ensuring that all the proper ui elements were drawn to the screen, and making sure there were no anomalies. It was able to pass this test.

The render() function was tested in much the same way as the show() function, we predicted what would happen, and then upon trying our actions it would. When walked to an activity would the ui prompt show up. When we did an activity would the stats, time, and energy meter change appropriately. When we slept would the day change. It was able to pass all these tests.

handleInput() was tested by moving the character in all directions, it was able to pass this test.

pause(), saveCharacterPosition(), and resume() was tested by taking note of the players position, pausing the game, then unpausing and noting whether or not the player's position was different. It was able to pass this test.

resize() was tested by changing the resolution of the game in the setting menu, and seeing how the camera was affected. The function worked as intended, however slight changes were made for better player comfort.

The different activities were tested in the same way, with a player walking up to each activity and testing whether or not there was a screen drawn that would allow them to interact with the activity. Then when the activity was interacted with, the result on the stats shown on the screen was noted, and compared to the expected value change. This went through a few variations where it was not behaving as expected due to issues with how scores were being calculated. But was fixed by the final version of the game.

InGameMenu
The in-game menu is drawn to the screen with buttons to interact with that can change the resolution, go fullscreen, and quit the game. Each of these interactions were checked for functionality, and were properly working

MenuScreen
The menu screen shows at the start of the game. It was initially tested, with just the start, settings and exit buttons. It was able to perform all of these actions properly. When the leaderboard was added, there was initially a problem with the rendering as the previous menu screen and leaderboard were overlapping. As this problem was trying to be fixed, it was tested by opening the game to the menu screen and seeing if issues had been resolved. By the final version of the game, the menu screen was properly functioning.

SettingScreen

The setting screen was tested by opening the settings menu, and trying out the different functions it should perform, and checking if they were performing accurately. It was able to change resolution, toggle between fullscreen and windowed with default size.

StatsTracker

The visual component of the stats tracker was tested in other areas of the game and not independently as it does not draw itself to the screen without being called. Errors that were found in displaying scores were not due to incorrect rendering, but miscalculation in how the score was being added, which was fixed before the final version.