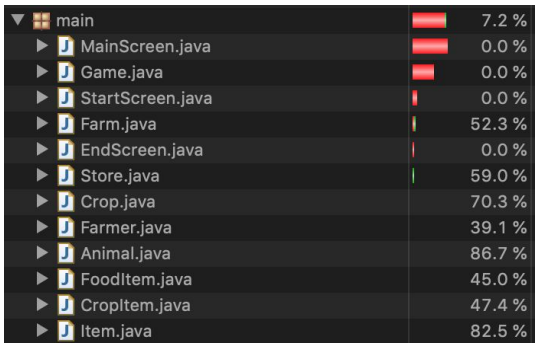
Project Report

Name: Sarah McLaughlin, **ID:** 83163304
Name: Daniel Gomes-Sebastiao, **ID:** 73548550

Structure Application and Design Choices:
Our structure for the assignment is simple. When we were designing the UML, one of our biggest decisions was deciding whether Crops, Animals, and Items should be under one superclass or their own separate classes. We decided that Crops, Animals, and Items were different enough to be their own classes as they had their own number of variables and methods, but that Crop Items and Food Items so closely related that making them subclasses of one large Items class was obvious. Otherwise, communication between classes only happens when an object is instantiated in another class. Another design choice we made, although not related to class structure, was making the store repopulate with random numbers of items. We thought this would add a fun amount of volatility to the game, even without random events.

Test Unit Coverage Explanation:



The unit test coverage mostly reflects what we did and did not test. For MainScreen, Game, StartScreen, and EndScreen, JUnit Tests would be ineffective, as they are complex classes that use the base classes which we have test coverage for. In the Case of Store and Farm, lower test coverage is based on two things: one is updating arrays, which we couldn't do because our crops, animals, and items have static IDs and can't truly be equal to any other object, and the other is for parts of methods that return null. All the other classes' percentages vary based on how complex the methods are. Farmer has low coverage because its only methods are getters and simple setters, which are not included in our JUnit Tests. Overall, the test coverage is low because we have most of our code in the GUI.

Thoughts and feedback on project:
We have appreciated the support from the course authorities for this assignment. One of the main issues around the project was juggling with the pandemic and for Sarah, having to return to her home to the U.S. during time that the assignment was active. Having the scope of the assignment reduced certainly was an aid to us. We did however find the assignment quite unstimulating. We both have experience in java, and have built similar projects like this at a highschool level. We understand this project will likely have some java new-comers and understand that the assignment's main goal was on the software development process, but would have liked to have engaged in a more professional atmosphere.

What did and didn't go well:
Together, we have enjoyed collaborating on this assignment. Although we were on other sides of the world due to the pandemic, the time difference had mostly worked to our advantage - one member working for the day and the other picking up where the project was left off. We admit that we underestimated the time the assignment would take. Because of this, we shared a series of long code sprints throughout the last week before the assignment was due, which led us to being quite overwhelmed. They certainly were tough, however there were a few things that helped smooth things out. We were both familiar with version control and would meet up regularly over video chat to discuss our goals or any issues we were facing at the time. In hindsight we should have spent more time on class design, although at the time of our first draft we were feeling quite under control. We ended up making quite a few changes as we developed the CLI. In particular, we realized that we needed to assign a unique ID number to most of the elements in our game. Without this the fundamentals of the game were not working securely, that is the main operations like buying an item from the store or using items on a crop or animal. Overall, having active communication and sharing the goal of achieving well in this assignment was the core driver for us to complete the assignment.

Effort in hours:
Sarah: 55 hrs
Daniel: 50 hrs

Agreed % Contribution:
Sarah: 50%
Daniel: 50%