MS3 Report: Syntax Sisters

Vision

Our current vision is a Pet simulator that supports two types of animals, camels and dogs. The simulator tracks the pet's status in different categories: health, happiness, nutrition, and energy. In order to maintain these different statuses the user can choose to perform different actions including feed, play, clean, walk, nap, and train. The more actions performed, the more hungry the pet gets. The user can also pay to bring their pet to the vet when diagnosed with a sickness or buy items in the shop to increase the pet's statuses. Another feature as a part of walking your pet is the possibility for your pet to run away. This can be prevented by purchasing and using a leash or microchip. Finally, there are several minigames the user can play in order to earn money for their pet. The addition of the possibility for the pet to run away (which ends the game), taking your pet to the vet, and the minigame ideas are new from previous plans. We thought these additions made the simulator more complex and interesting for the user.

Summary of progress

Since MS2, our pet simulator has really transformed. At MS2 we had working terminal functionality and our setup for the program, but we only had a couple basic capabilities of being able to create, name, feed, and play with your pet. Since then, we not only implemented other activities of similar structure–walk, clean, nap, ride–but also completely expanded the options. The statuses of your pet: health, happiness, energy, and nutrition, get updated based on your choices for these capabilities, sometimes resulting in your pet getting diagnosed with a sickness. This sickness is tracked and your pet can get treated for its appropriate sickness in the vet feature, returning some of the lost statuses. Similarly, there is a training feature that allows your pet to unlock skills relating to speed, strength, or intelligence. These skills can increase your status bars for energy, health, and happiness, respectively. There are 4 skills you can earn in each category, maxing you out at /14 for your status bars, meaning you have longer before your pet will run out in any of the categories. If your pet is low, however, we have implemented a shop feature that allows you to purchase items that will boost your pet's statuses, as well as a leash that could come in handy on walks. All of these

features are consistent across the simulator, so before you buy a leash in the shop, you cannot use it on a walk, for example.

We also implemented money functionality: the ability to both earn and spend money. We have multiple ways you can earn money for your pet through minigames. You can battle another pet by trying to pick the correct options for points which earns you money. You can play a cooking minigame where your meal gets judged and awarded money based on its quality. You can also risk your money by playing blackjack and trivia, both of which have the capability to make you lose money as well as gain it. And lastly, there is a chance minigame that has the user try to stop a simulator when there are a certain number of dollar signs shown on the screen where 5 or more will earn you money. These minigames and features can be played as many times as the user wants, but your pet will continue to get hungrier the more actions that are performed, so you cannot go forever without using Feed. All of these functions are described for the user in a Help function. The last thing we added since MS2 was making the game quit whenever any of the pet's statuses reach zero. If happiness, nutrition, or energy reach zero, the pet leaves to find a better owner, and if health reaches zero, the pet dies–both effectively ending the game. Our pet simulator really transformed between MS2 and 3 and we are very happy with the functionality we ended up with.

<u>Activity breakdown since MS2</u>

Bella Falkenberg (alf253):

Hours: 12

- Vet function and sickness feature
- Leash and microchip feature
- Skills feature and its impact on the maximum levels for status
- Nap function
- Test suite
- Documentation

Megan Cowart (mec328):

Hours: 17

- Chance minigame using a simulator

- Help and Ride functions
- Decrement nutrition periodically
- Debug and fix format across the board
- Manually tested everything

Ena Kovac (ehk75):

Hours: 16

- Battle minigame
- Train minigame
- Shop function
- Test suite
- Walk, Nap, Status, Clean

Claire McHugh (cdm234):

Hours: 12

- Cooking minigame
- Blackjack minigame
- Trivia minigame
- Added functionality to parts of the game for money (items having a cost, spending money, losing money, earning money)

Productivity analysis

As a team, we were quite productive throughout our project sprints, successfully accomplishing what we had planned. Our estimates for the tasks and features we set out to implement were accurate, allowing us to effectively manage our time and expectations. We also followed a more individual timeline for sprints as our workloads varied significantly per person per week which worked well for us. The project went smoothly although we occasionally felt constrained by the terminal-based interface, which lacked the flexibility and user-friendliness of more sophisticated GUIs like those available in Java, this limitation did not significantly hinder our progress. Ultimately, we adapted well to these constraints and are proud of the work we achieved.