CS112 Final Project: Sebastian Paredes, Lisa Phan, Alexander Womack

While our original game design was meant to be a cross between The Sims and Stardew Valley, watching over a group of villagers while also getting a good sense of their simulated personalities and growing attached to them, our final product has steered much more towards the former than the latter. This is primarily because our changes in programming have come about over a realization that we were too ambitious with our project. Creating the village and having a detailed set of villagers doing their daily tasks in a Sim-esque format was a suitable task on its own, and spending the extra effort to simulate personalities in these pixelated characters would have likely taken too much extra time for us to have succeeded. In general, our capacity for events have been drastically reduced, with the game meant to be more of a society simulator in which you watch the villagers continuously go through their daily tasks as the seasons slowly change.

There were a number of core parts to our program that we kept the same. For example, our abstract person Class being the backbone for the behavior of each occupational subclass, the event log and the sidebar keeping tally of what has happened while allowing the user direct opportunity to interfere, and the general structure of the classes and in how they related to each other (e.g. World). All of the parts of our program that we kept in were very purposeful decisions, as they provided the backbone for us to expand on further if we wished. For example, our Plant class originally allowed us to plant flowers randomly throughout the village to add to nature, a class that I later on used to also add trees into the game, which are much larger and more intricate than the flowers graphically and also change with the seasons! Similarly, a created Structures class which originally was meant to just contain the houses of the villagers were expanded upon so that the workplaces of each occupation type would also be created. The main differences came about through the cutting of content to create a polished finished product by the due date. There were a number of things that we originally planned to implement in the game which did not come to fruition: extra subclasses of Person, for instance (there are currently five in the game, when we originally planned to have closer to ten), extra characteristics that would be randomly assigned to characters upon their creation, and other similar examples of us trying to branch out more from our foundation but realizing that doing so would simply require too much time and resources to be feasible.

In some cases, the foundation itself was too complex or error-inducing to implement fully into the program, one of these being the movement of the villagers, which although we were close to implementing with a full collision system with the surrounding environment, were unable to get past the multitude of errors that it caused and scrapped it for submission. Other times attempting to create new events ended poorly in ways we were not able to decipher, letting us stick with the original events that we had already planned and were working on implementing (e.g. the plague event that gets other villagers sick and dying, the murderer event where a villager becomes a murderer and kills anyone they come into contact with, etc.)

Other attempted additions were much more successful and were woven seamlessly into our original design. For example, we added a minor beat of story in the form of an introduction given to the player by God themself, a matching final screen where God judges your results based on your performance in not leading the village to doom, and a variety of extra details to increase the atmosphere of the game, such as adding more music and sound effects.

While in some ways the scope of our project was significantly reduced as we came to terms with our overambition, we also made an effort to add content of our own in order to make up for that, expanding on the foundations we had already laid down with existing code and making sure the game felt like its own experience rather than a poor imitation.