Name of Game: Spot

Team Members:

- Emily Zhou (ez4cc)
- Li-Chian Wang (lw3qc)
- Claire Frank (cm3ap)

GitHub Link:
https://github.com/UVA-CS4730-F17/game-project-spot

Game Pitch:

Explore the Osterheim family's life through the eyes of their dog, who works to comfort them through their time of struggle. Spot is an original visual novel that explores how we deal with grief, reminding us all to try to exercise healthy coping methods and to take care for each other during great hardship.

How to Play:

To play Spot, all you need is a mouse or laptop track pad. To begin the game, click on the start button in the main menu. Once in the game, you click on the dialogue box to continue to a new set of dialogue. When given a number of dialogue options, you click on an option to lock in your choice and continue on through the game.

If you wish to speed up the game, click on the "skip" button at the top of the dialogue box. It will quickly move through the dialogue, no longer requiring clicking on the dialogue box to progress through the game. You can exit the skip mode by clicking on the dialogue box again and it will return to normal speed dialogue and normal functions.

If you wish to look at a log of all of the dialogue so far, click on the "log" button at the top of the dialogue box. It will pull up a list of previous dialogue which can be scrolled through using the mouse (such as swiping on the track pad or using the scroll wheel on a physical mouse). To exit the log, press the "back" button at the top right of the screen.

Amount/Type of Content Available:

There are five playable levels, each featuring two to three scenes to play through. In these scenes, the player usually chooses actions for Spot to do by selecting an option presented in a dialogue tree above the dialogue box. There is also an ending level, which is plays one of two endings, depending on the choices made in the five previous levels. In this ending level, the player simply clicks through the dialogue.

Playtesting Report:

CS-4730-Student Playtesters Report:

We got a lot of positive feedback around the art assets and music in the game; people seemed to really dig Emily's drawings. People asked for more feedback around the choices they make in the game, saying that the dialogue itself was not enough. We took this into account and chose to add in facial expressions for the characters that would change in response to the choices the player made. Had generally positive feedback around the story itself, with people finding it emotionally engaging. There was some general confusion about some of the extra buttons that did not work, which was to be expected because they were not meant to work as we were planning on removing them before finishing the game.

Non-CS-4730 Student Playtesters Report:

Claire's Roommate, a Chemistry major:

Claire got yelled at for making a game that made her roommate cry. Claire considers that good feedback as the purpose of the game is to be emotionally accessible and immersive. She found the controls generally intuitive, though pointed out it would be a good idea to have an explanation of the controls at the beginning of the game for players not familiar with the control scheme for visual novels. She also found the extra broken buttons confusing, but again that was to be expected and will not be an issue with the final game as those buttons will not exist in the final build.

Emily's Roomates (non-game class CS major, Psychology/Cog Sci major):

The non-game class roommate was impressed with the assets and expressed curiosity and interest in the fate of the characters in the story. She believed that the game was well-written but the interface was a little strange because there were extra buttons. We later removed them from the screen. The controls were intuitive to her and she picked up how to progress without prompting.

The non-CS roommate thought that the gameplay was smooth and that the emotional impact was not as sad as she had been expecting. She felt that the correct choice was sometimes a little obvious, but maintained that she also did not always choose the correct answer. She found the controls intuitive but did not use the skip or log buttons though they were available to her. Total gameplay took about 20 minutes, but her opinion was that she would have liked to play the game again to get a better experience, which was a positive sign.

Lessons Learned:

Visual novels are super hard - branching is a thing

Designing and building an engaging visual novel is difficult because of the amount of writing involved and tracking the branching. For this game, we had reduced the scope to prune the story branches to make it possible to have a final build for the story. This meant that version control via Google Drive and having the storyline work and art work being handled by one member for each was possible. However, if the story were to expand, managing a natural flow would be difficult. Additionally, the workload would increase exponentially and would require better tools for version control beyond just a simple text document.

Just because it runs in Unity does not necessarily mean it will build

We ran into the issue where in the original Visual Novel framework we were using would run fine in Unity itself but would crash if we tried to build the actual game. After doing some online research and working with a TA, we came to realize that the framework was outdated and the original japanese code could not be easily modified to actually build on our computers, mostly because none of us are fluent in Japanese. We quickly settled our losses and found a different framework to work with that ended up being easier to use than the original one.

How to work with Joker Script (Google Translate is great)

Joker Script is the framework we ended up using for our game. It was created by Shikameoku MK (シケモクMK) as freeware and can be used for mobile game development in addition to computer game development. While the documentation is in Japanese, we were able to use Google Translate to read it and found it fairly intuitive and easy to use for what we wanted. We definitely recommend it for people who wish to create their own visual novels.

Also, if you google Joker Script, mostly Batman stuff shows up. If you want to look up where we found the framework, go to https://mightquiverwalk.itch.io/joker-script.

Asset management/creation

Creation of assets and development of the code is easiest when both processes are decoupled. Over the course of the project we created over 30 original art assets for the game, which taught us that art takes time and things are simpler if art and code can be kept independent. It became apparent at the start of development that having placeholder values for the art assets that were yet to come was important to allow for testing and progress on the technical side. Additionally, having a common schema to follow for naming conventions, organizing the scripts and branching in a common way so that different team members could work on different parts of the novel helped us be successful and implement the game with fewer catastrophes.

Additionally, due to a version control issue that wiped a huge amount of work by accident, we learned to store different copies of assets in different places. The original full-format art was stored locally and the .PNG files were stored in the master repository to avoid cluttering the game with unnecessary bulk. Additionally, we kept one folder separate from the Unity folder with all art assets as well as copying those same assets into the requisite asset folders in the game files.