

CSE 20212 Final Project Report: The Pokemon Project

J. Patrick Lacher
John R. Adams
Douglas Schmieder

The README.md in our repository (<https://github.com/placher/pokeproject>) contains a user manual describing how to install and play our game. As the readme indicates, the game files for the final game are included in the “ThePokemonProject” folder. The main folder of the repository also contains development files, a full copy of the SDL2 library and SDL2_Mixer, all of the reports we have generated for the project, and some supplemental PDFs that may prove helpful in expanding the gameplay experience.

Looking at our completed project, we’re happy to report that we were able to meet all of our primary goals. We have character movement through a vibrantly-illustrated world, dynamic battles between the player and NPCs, and we were even able to implement contextual background music in every area. Our overworld areas were all drawn by hand, and are rendered as SDL textures, with collision detection functions to implement walls and other solid objects, and our battle screen also makes use of our own drawn textures for the background and the Pokemon themselves, which took hours of tweaking to get SDL’s cooperation.

One of our hardest areas of implementation was the battles. We have an incredibly efficient data-structuring system, with trainer classes containing the different pokemon and the pokemon containing similar vectors of move objects. With this effective of a storage system, we were able to implement a command-line level battle function with all sixteen original pokemon types and thirty-nine unique pokemon that remained faithful to the original game’s battle algorithms with relatively little effort, having a working system in just over an hour. We then realized that, in a graphics based game, to not implement a major portion of the game in SDL was lazy on our part, so we refocused our efforts, and turned our attentions to trying to recreate the in-game battle sequences of the source material. We had some issues differentiating the `SDL_PollEvent()` handlers for moving the character sprite and controlling battle operations, but we eventually worked out the bugs, and were able to fully implement battles rendered in the same style of the original game and with different display information for each of the thirty-nine pokemon.

Something that we added near the end of the project was music. It was always a feature we had wanted to implement from the beginning but we were never sure how to go about it. When we finished the battle system, we looked into learning `SDL_mixer` to implement overworld music and battle themes. It was a little tricky to make sure the music started only once each time, but in the end we were able to implement unique music for the overworld, each battle, the win screen, and the lose screen.

With any programming project this size, there are always bugs. We were able to eliminate nearly all of them through rigorous playtesting, but one persisted. In some of the later rooms, if you try to backtrack off a cliff the character sprite will appear to jump around the corner to the nearest valid space. This isn't game breaking, and most players won't notice it, but the bug is there. Other than that, though, our code actually runs remarkably smoothly.

We've all had a great time working on this project, and are looking forward to showing you the game we've spent the last three months building.