Grant Boddy
03/14/2025
CSC 4330 – Programming Language Concepts
Dr. Tamhankar

4330 Project Midpoint Report

Project Reflection: Murder Mystery Game Development

When I first started this project, I had my sights set on building an Escape Room game using a Domain-Specific Language (DSL) to handle puzzles and interactions. The idea was that players would explore different rooms, solve clever puzzles, and piece together clues to escape. But as I started developing it, I ran into a major challenge: connecting the puzzles to the actual story in a meaningful way. It felt like I was just stringing together logic problems without a real narrative flow. The puzzles were fun, but they didn't feel like they belonged in the game world.

That's when I had an idea: What if I turned this into a murder mystery instead? A mystery game would still use a lot of the same foundations, exploring different locations, interacting with objects, and solving problems, but now it would have a stronger story driving everything forward. Instead of players randomly finding puzzles, they'd be given all the evidence upfront and tasked with analyzing it to uncover the truth.

Expanding the Concept:

Like any project, this one has been filled with trial and error. Before I even wrote a single line of code, I spent a ton of time just working on the images for the game. I needed:

- **Different backgrounds** for the crime scene, evidence table, and interrogation rooms.
- **Character images** for suspects, detectives, and other NPCs.
- **Evidence files** like handwritten notes, fake text-message threads, and crime scene photos.

This part of the project took way longer than I expected, choosing the right images, editing them, and making sure they fit the game's tone was a process in itself. But that wasn't the only challenge. Writing the story and creating the evidence was just as difficult. I didn't want to make the mystery too obvious or too confusing, so I had to carefully balance how much information the player gets at the start versus what they uncover through questioning suspects.

From Concept to Code:

Once I had the story, characters, and evidence planned, it was time to actually build the game. Instead of jumping in blindly, I designed the layout for the full program first.

First, I set up a file system with organized folders for assets, scripts, and text-based data. Then, I built the core files one by one:

- **graphics.py** to handle images and text renderinggraphics.
- **dialogue_system.py** to manage character conversations.
- **game_engine.py** to control how the game flows between different scenes.
- **main.py** to bring everything togethermain.

Finally, I started implementing each feature, beginning with the intro scene, then the evidence system, and then the suspect interrogation mechanics.

Some Challenges Experienced:

One of the biggest challenges has been growing the program while constantly revising old code. Every time I add a new feature; I have to go back and update the existing files to make everything work together. For example, when I introduced the dialogue system, I had to change how text was displayed in graphics.py. Every time I added new evidence or a suspect, I had to update multiple files to make sure they appeared in the right place.

This has made the project way more complex than I originally expected, but at the same time, it's been an amazing learning experience. Seeing everything come together, watching the detective talk to suspects, examining crime scene photos, and slowly piecing together the truth, makes all the effort worth it.

Next Steps:

Alright, now onto the fun part: what's next!

- **Finishing The Game:** Obviously, I still need to finish the game. It's coming along nicely, but it's still not done and still needs lots of work. For example, more characters, more evidence, refined backgrounds, possible sound effects and music.
- **More Dynamic Dialogue:** Right now, the dialogue follows a set path, but I want to introduce a branching system. If a player questions a suspect about certain evidence, that should affect their response.
- **Polish & Aesthetics:** The game is functional, but I want it to *feel* like a polished detective game. This means adding transitions, animations, and some ambient crime-noir music to set the mood.
- **Finalizing the DSL:** My original idea of a Domain-Specific Language is still very much part of the project. I need to refine it so it allows:
    - Simple script-based definitions for new rooms, characters, and interactions.
    - A way for external users (hypothetically, future game designers) to create new cases using my system.

Final Thoughts:

This project has been a wild ride from start to finish. What started as an Escape Room game turned into a fully interactive Murder Mystery, and while that shift came with its fair share of challenges, it also made the game way more engaging. Along the way, I've gained tons of experience in:

- **Designing a Domain-Specific Language (DSL)**: Creating a structured way to define rooms, evidence, and character interactions using a custom script format.
- **Parsing and interpretation**: Writing a DSL parser to process scripted game events, ensuring that dialogue, interactions, and evidence function dynamically.
- **Syntax definition & structured data processing**: Developing a format that allows the game to interpret text-based scene descriptions, character dialogue, and clue interactions efficiently.
- **Programming Language Concepts**: Applying tokenization, execution, and parsing techniques to transform simple text-based inputs into a fully interactive mystery-solving experience.
- **Game structure & design**: figuring out how to connect different systems.
- **Programming large-scale projects**: keeping everything organized and manageable.

There's still work to be done, but I'm proud of how far the game has come. Now, all that's left is to solve the case!

Timeline:

Here's the revised timeline for wrapping things up:

- **Week 1:** Finalize all evidence items and characters. Implement interactive evidence and questioning system.
- **Week 2:** Implement and improve dialogue and questioning system.
- **Week 3:** Final polish: UI tweaks, all image files, animations, and more.
- **Week 4:** Test, debug, and finalize the project for submission.

Code Snippet:

Below is a snippet of code from my main.py file. You can see the different files and functions coming together as well as how I'm registering and organizing the evidence items.

```python
import pygame
from graphics import Graphics
from game_engine import GameEngine, DialogueScene, EvidenceBoard
from mystery_dsl import MysteryDSL


def main():
    # Initialize systems
    graphics = Graphics(740, 650, "Murder Mystery Investigation")
    game_engine = GameEngine(graphics)
    dsl = MysteryDSL(game_engine)

    # Register evidence items
    dsl.register_evidence(
        "crime_scene_photo",
        "assets/images/crime_scene.png",
        "Crime Scene Photo",
        "The body was found in the study at 11:45 PM. Notice the overturned chair and scattered papers."
    )
```