Final Project Proposal
Murder Mystery: Choose Your Own Adventure

Overview:
This project will be a Choose-Your-Own-Adventure game, centered around a murder mystery. We will use several files to contain each path, and make 16 endings, some of which are dead ends and others that will give you, the player, the opportunity to choose the culprit.
Doe encounters a young lady, Alice, whose older brother, Felix has gone missing. She hires them to investigate.

Suspects:
Mrs. Irma Caro is a shady business owner in a shady town. She makes deals with Felix's boss through Felix. She does not like him. He likes to flirt with her husband, David.
Mr. Peter Arnolds is a general goods shop owner whom Felix blackmails into giving up profits and storing legally dubious items. (Killer)
Ms. Trixie Stevenson is Felix's boss, who thinks Felix was an annoying busybody. He was in charge of blackmail, extortion and business deals with the smaller and shadier businesspeople.

Clues:
Request from Mrs. Caro: A letter requesting that someone else please be in charge of the deals between Caro and Stevenson. Also a request that Felix be fired.
Planner: Felix was planning to shake down the people he was blackmailing the day he went missing. Called it "shaking down the tree".
Knife: One of the generic ones. The local general goods store carries these. Covered in blood. No fingerprints.

Solidifies and showcases:
- Inheritance
- Polymorphism
- Abstract classes
- ArrayList usage
- Calling methods from different classes.
- Recursion.
- Overwriting or overloading .toString()
- Objects and their constructors.

Necessary Classes:
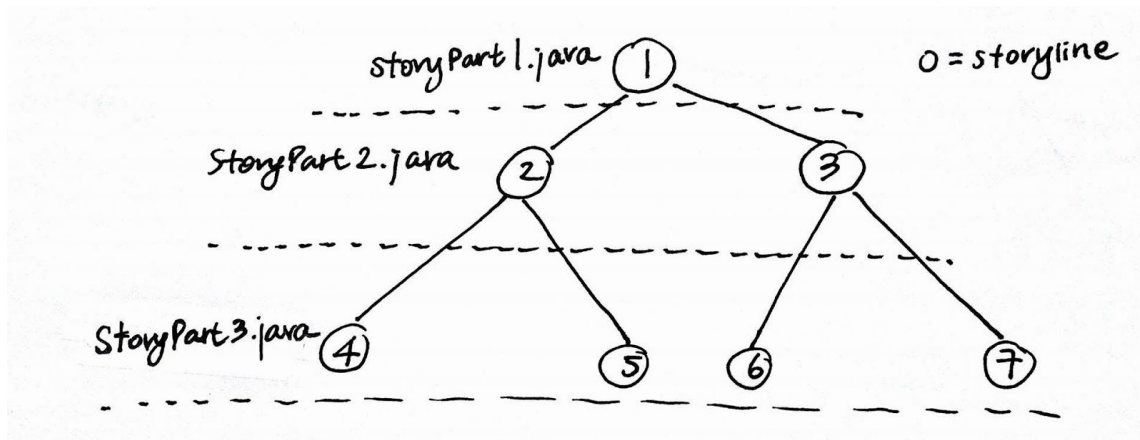
1. **Woo.java**
   Attributes:
   - checkPoints -- ArrayList that consists of a unique list of numbers, which denote the path that the user has taken, adding on numbers to show the user's choices as he/she continues.
   - Inventory -- An ArrayList that contains instances of Clues. As the story continues, user may accumulate Clues.
     Methods:
   - Juncture methods -- It takes user input, choice of path, in the terminal as a parameter. It adds on the corresponding StoryLine number to the checkPoints ArrayList. Also lets you into the inventory, take stock of things, and so on.
   - Then initializes the next StoryPart# class, with the last int in the checkPoints array as the input to find the next StoryLine that corresponds to the user's path..

2. **Clue.java**
   Attributes:
   - Name: name of Clue
   - Description: description of Clue
     Other:
   - Clue constructor -- initializes clue name and description.
   - Description accessor -- returns description of Clue.
   - Overridden toString() -- so that we see the name of the clue.

3. **StoryPart.java**
An abstract superclass containing abstract method *match*, which each of its subclasses will implement.

4. **StoryPart1.java (implements StoryPart)**

There will be six story parts: StoryPart1, StoryPart2, StoryPart3, StoryPart4, StoryPart5, and StoryPart6. Each will contain a different number of StoryLine methods.

- <u>Match</u> -- Inherits abstract method *match* from StoryPart. Takes an int (Woo.java will input the last number in the CheckPoints ArrayList), and uses if and else statements to match the input, the user's last choice, to the correct StoryLine method.
- <u>Storyline methods</u> -- There are a multiple of these methods within a single StoryPart# class, depending on the Part number. Examples of StoryLine method names include StoryLine3, StoryLine4, StoryLine14. Prints appropriate story with description of next steps. Returns an instance of Clue.