Project Report

**Team Name:** Silver Arrow

**Team Members:** Hailey Van Hoorn, Aman Ali, William Kwon, & Ethan Freeburg

**Brief Description:**

This is a game where the main character’s father wants to disinherit his eldest child from his fortune and company leadership. At the father’s retirement party, the invited board of directors are holding a vote for the new CEO of the company, and the father is set on recommending the main character’s more capable younger brother. The main character has to complete certain actions to manipulate the guests so that they will vote in their favor instead of the younger brother.

**User Manual:**

In the game you’ll find yourself within one of the rooms within the house. The 2 main options the user has in the rooms are to “**interact with \_\_\_\_\_**” one of the objects (including other characters) in the room or to “**move rooms**”. The interacting option will open a sublist of options which concerns your options in dealing with the object. Pay attention to the introduction of the game and dialogue when interacting with the other characters as they have tips for how to proceed. When filling the aforementioned blank in “interact with \_\_\_” please follow the format and please attempt to keep your description of what you wish to interact with to the bare minimum- we cannot account for all variances.

After getting a hold of the controls in the game, navigate the rooms and fulfill the conditions which are needed when dealing with the characters on the board of directors. Often you will need to add something to your inventory to deal with these characters. Manipulate the situation so that at the end of the night you are guaranteed to win the vote.

**Documentation for Items from the Solution Requirements List:**

(line # - Requirement)

|  |  |
| --- | --- |
| 0002 | Import another Python file [20 pts] (import time module) |
| 0005 | A Class [40 pts] (defines what the characteristic of all characters are. Some, but not all, of them are the character’s inventory, location, and movement capability) |
| 0023 | 2 Styles of Comments [10 pts] |
| 0038 | Try/Except Block [20 pts] |
| 0064 | A list that contains lists [35 pts] |
| 0065 | List [15 pts] |
| 0069 | Random Number Generator [20 pts] |
| 0102 | Python code that “walks” through the contents of a List [10 pts] |
| 0106 | Using += [3 pts] |
| 0115 | Using built-in list functionality [3 pts] |
| 0125 | Function Definition and Function Call (Call on line 0264) [10 pts] |
| 0129 | Function Definition with Default Parameters and Function Call (Call on line 0270) [15 pts] |
| 0133 | Using the formatting for strings (with print() statement) [5 pts] |
| 0142 | File Reading [35 pts] |
| 0157 | For loop [15 pts] |
| 0259 | A set [20 pts] |
| 0261 | While loop [15 pts] |
| 0263 | Assignment Statement [3 pts] |
| 0265 | Nested if Statement [10 pts] |
| 0268 | Dictionary [25 pts] |
| 0272 | print() statement [2 pts] |
| 0280 | Nested Loop [20 pts] |
| Total | =========================[351 pts]================================ |

**Design of Program:**

1. Class Item
   * 1. Attributes:
        1. name: Internal name of the object. Honestly, it isn't used that much
        2. actionDeclaration: The words which appear when the player enters a room
        3. intro: The intro called when the player decides to interact with the object
        4. option\_text: The text of the options presented to the player
        5. reaction\_text : the text when the player decides to do some action
        6. reaction\_reward: the value based on the decisions that you made.
        7. success\_check: checks if the completing action is made
        8. success: guarantees whether the completion action is made
     2. Methods:
        1. Interact - uses attributes to explicate the interaction
   1. Class map
      1. Attributes
         1. Map array (holds all locations, including the player, clues, suspects, and objects)
      2. Methods
         1. Move up / down / left/ right: Move the player around the map
         2. Display Action: Tell the player what actions they can take on that square
         3. Take Action: Take the action that the player has requested on that square
2. Def main
   1. Create the environment and assign roles
      1. Call class constructor, which also initializes all the objects on the map.
   2. Place the characters and give the characters an introduction
   3. While the character has not “dealt” with all the guests in a beneficial manner for the user:
      1. Ask the character what to do next
   4. end

**Flowchart from user side** (added for the sake of depth)**:**

Included separately as a pdf file

**Errors/Problems:**

When dealing with the inanimate type of objects (e.g. desk) the options are very similar and removes some freedom from the user. This is not a programming error rather an intentionally made one for the sake of simplifying the game.

**Questions/Messages:**