Repository Structure:

- Spots.py - Main game file. Runs the game, controls turns, and checks win conditions.
    - Includes game_setup, turn, game_end, and main functions
- bust.py - Implements the bust function which checks if a player busts when they have more than seven dice in their yard.
- chase_trick.py - Implements the chase trick, where players roll an increasing number of dice.
- diceplacement.py - Handles placing and burying after rolls.
- end_game_stats.py - Collects and displays game statistics at the end of the game.
- fetch_function.py - Implements the fetch trick, including rolling and selecting dice values.
- gobble_trick.py - Implements the gobble trick, including taking and returning treats.
- Howlfuc.py - Implements the howl trick, including drawing cards and rolling dice.
- player_card.py - Defines the Player and Card classes and card completion logic.
- random_roll.py - Generates random dice rolls used by different tricks.
- roll_over.py - Implements the roll over trick, rolling buried dice and updating the yard.
- trot_trick.py - Implements the trot trick, allowing dice movement and additional rolls.
- LICENSE - Information on the licensing of the program.
- README.pdf - File that explains the program and contains required assignment information.

How To Run The Program (Command Line):

1. Open the terminal and go to the project folder:

   cd path/to/project
2. Run the program:

   python Spots.py
3. Follow the prompts in the terminal to play the game.

Command-Line Interface

This program does not use command-line arguments.

All input is entered during gameplay through the terminal.

Data Types Used for Input

- Integers: used for numeric choices such as number of players or die value selection
- Strings: used for player names and selecting actions during the game

No external data files are required.

How To Use The Program:

Once the program starts, everything happens in the terminal.

You will first be asked how many people are playing (between 1 and 4), then each player will enter their name. The program determines the turn order automatically.

On each turn, the current player's info is printed to the screen, including treats, yard total, completed cards, and active dog cards. The list of available tricks is also shown with a short explanation of what each one does.

As tricks are selected by players they are temporarily unavailable until one trick is remaining and all tricks are refreshed.

The yard is where a player's buried dice are kept, and the yard total is the sum of those dice. Any die without a proper place on a player's active dog cards must be placed in the player's yard.

If the total value of a player's yard is ever greater than 7 the player busts. If a player busts their turn ends right away. They lose all progress on their active dog cards and their yard is emptied. The program checks for busts automatically whenever a player places dice.

To take a turn, the player types the name of the trick they want to use and presses enter. The program runs the trick and updates the player's attributes as required.

Once a player entirely fulfills one or more of their active cards, the player will be prompted to complete the card or not at the beginning of their turn. Completing one or more cards requires the player's entire turn. Completed cards are safe from busting and count towards the player's score. An additional way to complete cards is if a player fulfills all their active cards in one turn. In this instance, the program will automatically mark all of the player's active cards as completed without the player needing to spend a turn completing them.

The game continues until a player meets the win condition which is to complete six dog cards and the winner is printed to the screen with relevant data.

<u>Annotated Bibliography:</u>

Hague, A., Perry, J., Vickers, J. (2022). *Spots* [Board game]. CMYK.

Spots is a casual push your luck dice rolling game. Players complete their dog cards by rolling dice through different tricks they are able to perform on their turn. First player to complete six dog cards wins.

Attribution Of Code Contributions:

Method/function	**Primary author**	**Techniques demonstrated**
Bust function	Sean Tully	None
Chase Function	Noah Aurdos	None
Dice_placement function	Mackenzie Barrett	None
Stats function	Sean Tully	Visualizing data with pyplot
Fetch function	Mackenzie Barrett	List comprehension
Gobble function	Sean Tully	None
Howl Function	Samuel Onakoya	None
Player.__init__ method	Mackenzie Barrett	Composition of two custom classes
Card.__init__ method	Noah Aurdos	None
Card.check_completion method	Mackenzie Barrett	None

Card.__str__ method	Sean Tully	Magic method
Random_roll function	Sean Tully	None
Rollover function	Mackenzie Barrett	None
Game_setup function	Samuel Onakoya	Use of a key function
Turn function	Noah Aurdos	F-strings containing Expressions
Game_end function	Samuel Onakoya	None
Main function	Noah Aurdos	Sequence unpacking
Trot Function	Samuel Onakoya	Keyword arguments