Deacon Steiner

A) Design #1

C Game

- □ secret: Word
- □ wordList: WordList

- ● Start()
- ● makeGuess(guess: String)
- ● compareWords(secret: Word, guess: Word): Feedback
- ● printFeedback(fb: Feedback)

C Word

- □ letters: List<Character>

- ● length(): int

C Feedback

- □ correctLetters: int
- □ correctPositions: int

- ● isCorrect(): boolean

C WordList

- □ words: List<String>

- ● getRandom(): Word

B) Design #2

C Game

- □ secret: Word
- □ wordList: WordList

- ● start()
- ● makeGuess(guess: String): Feedback
- ● printFeedback(fb: Feedback)

C Word

- □ letters: ArrayList<Character>
- □ value: String

- ● Word(str: String)
- ● compareTo(other: Word): Feedback
- ● toString(): String
- ● length(): int

C WordList

- □ words: ArrayList<Word>

- ● WordList(initList: String[])
- ● getRandom(): Word

C Feedback

- □ correctLetters: int
- □ correctPositions: int
- □ wordLength: int

- ● Feedback(correctLetters: int, correctPositions: int, wordLength: int)
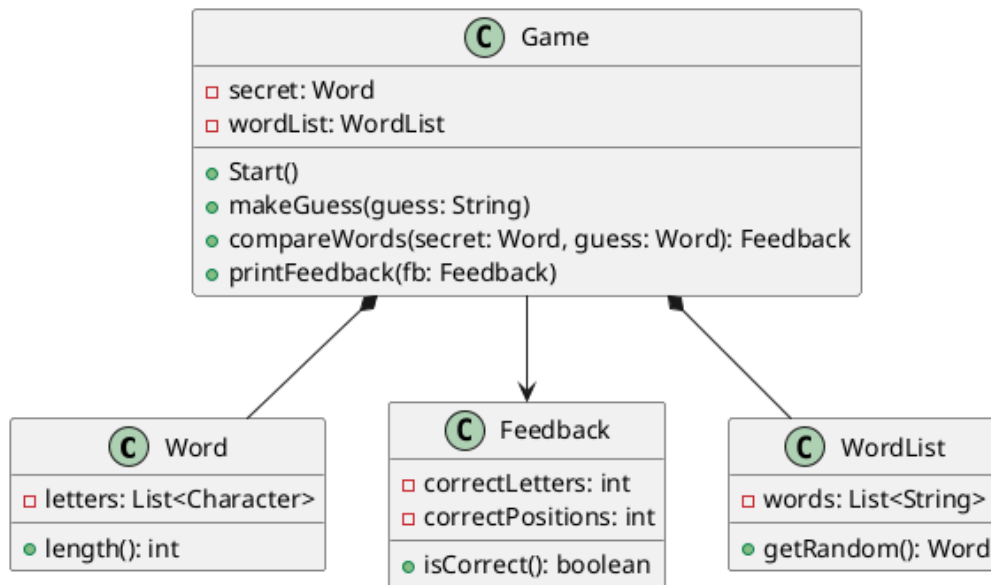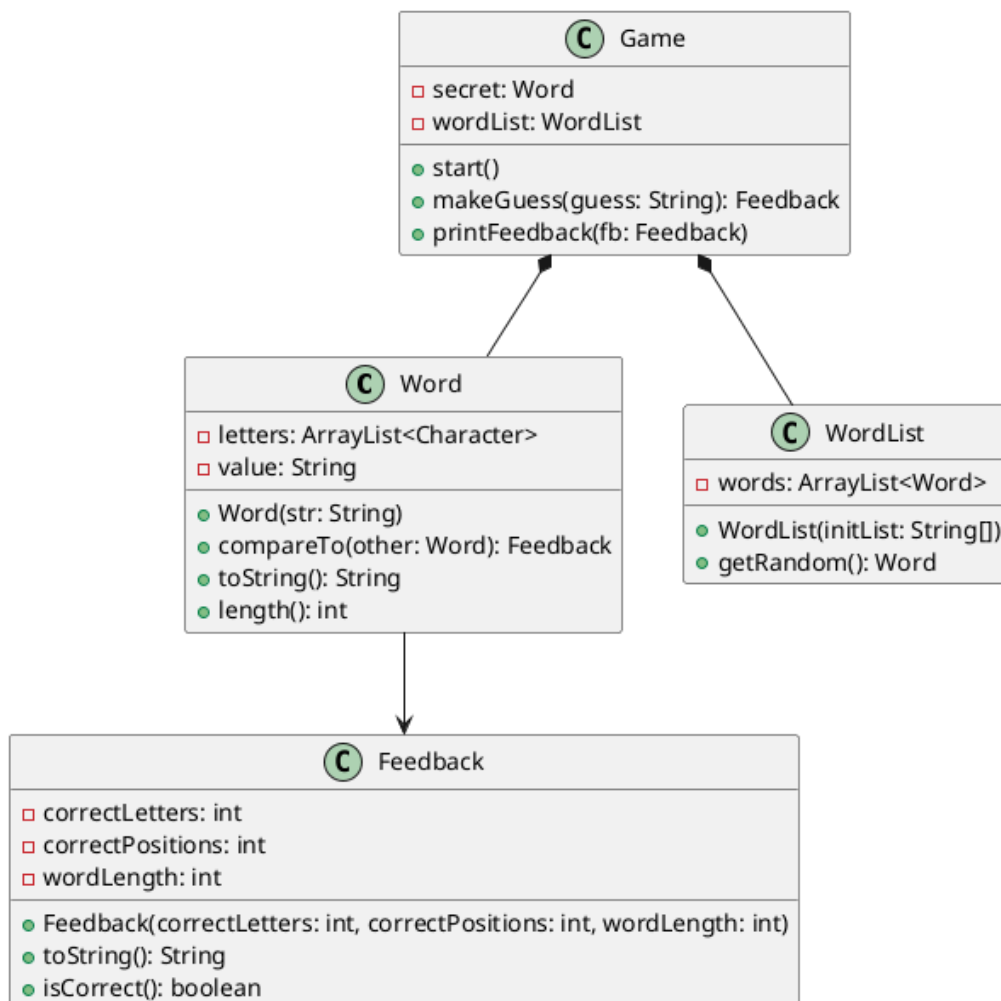- ● toString(): String
- ● isCorrect(): boolean

C) Implementation Choice

I have chosen design #2 because it better allows for separation of concerns and is a far more modular design that allows for class changes that don't require sweeping re-implementation across the program.

D) Instance Variables and methods

A) For this program, I decided to use four classes each with its own responsibilities.
   A) Word- which handled word creation and comparison logic.
   B) WordList- which stored the possible secret words and selected a random word from that list.
   C) Feedback- for post-guess results.
   D) And Game- used to initialize the game state and main loop.