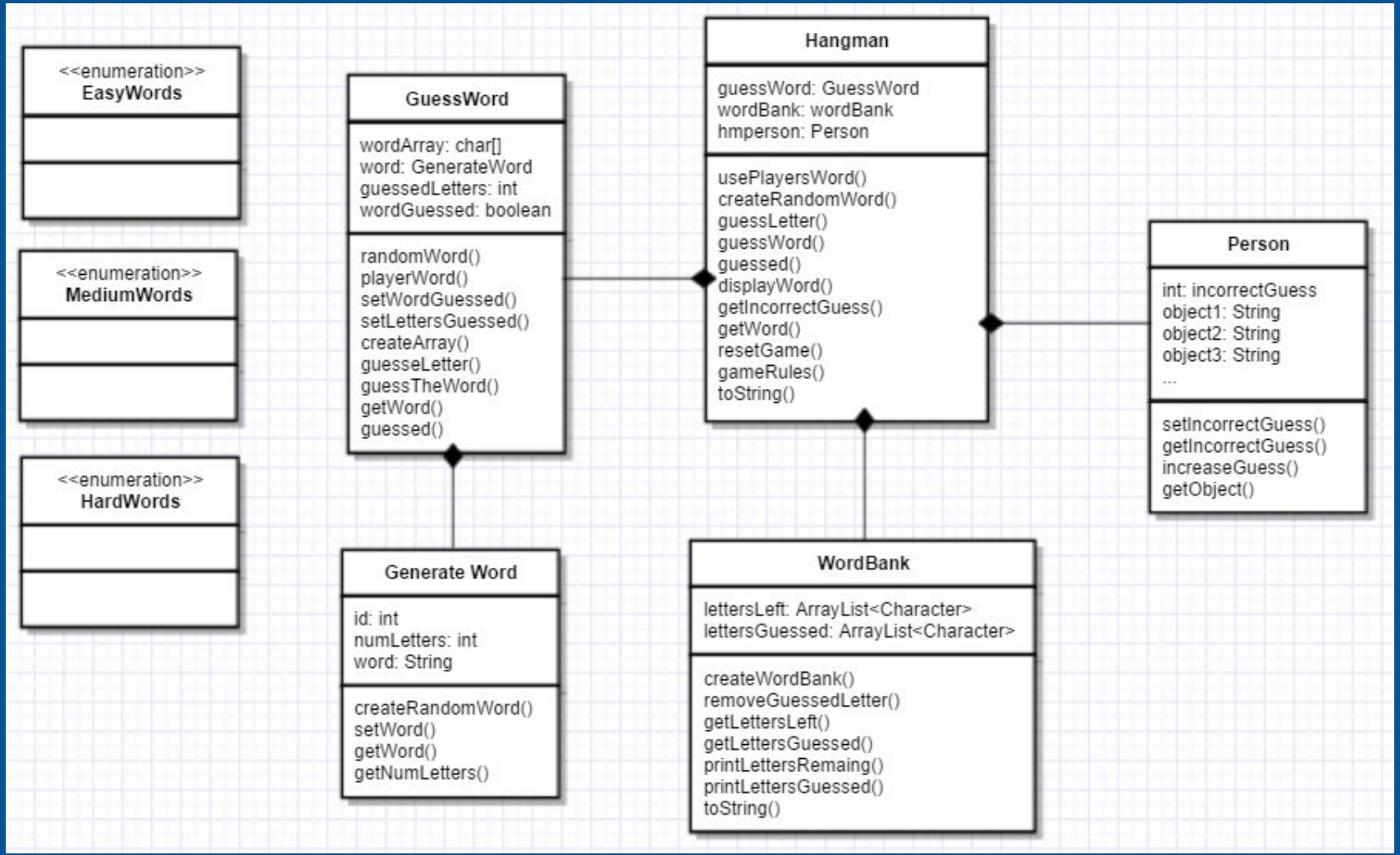
```
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   src/HangmanGame/HangmanMain.java

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Gittie Klein\OneDrive\Documents\MCO152 - Computer Methodology\Klein_G-mco152-2016F_hangman> git add .\src\
PS C:\Users\Gittie Klein\OneDrive\Documents\MCO152 - Computer Methodology\Klein_G-mco152-2016F_hangman> git commit -m "v
iew hangman after lose"
[master 889d4a9] view hangman after lose
 1 file changed, 1 insertion(+)
PS C:\Users\Gittie Klein\OneDrive\Documents\MCO152 - Computer Methodology\Klein_G-mco152-2016F_hangman> git push
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 524 bytes | 0 bytes/s, done.
Total 5 (delta 0), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/gittieklein/Klein_G-mco152-2016F_hangman.git
   b81aaaf..889d4a9  master -> master
PS C:\Users\Gittie Klein\OneDrive\Documents\MCO152 - Computer Methodology\Klein_G-mco152-2016F_hangman> git add .\src\
PS C:\Users\Gittie Klein\OneDrive\Documents\MCO152 - Computer Methodology\Klein_G-mco152-2016F_hangman> git commit -m "a
lways display to string after end of game"
[master a202d34] always display to string after end of game
 1 file changed, 3 insertions(+), 4 deletions(-)
PS C:\Users\Gittie Klein\OneDrive\Documents\MCO152 - Computer Methodology\Klein_G-mco152-2016F_hangman> git push
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 524 bytes | 0 bytes/s, done.
Total 5 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/gittieklein/Klein_G-mco152-2016F_hangman.git
   889d4a9..a202d34  master -> master
PS C:\Users\Gittie Klein\OneDrive\Documents\MCO152 - Computer Methodology\Klein_G-mco152-2016F_hangman> git pull
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 5 (delta 3), reused 5 (delta 3), pack-reused 0
Unpacking objects: 100% (5/5), done.
From https://github.com/gittieklein/Klein_G-mco152-2016F_hangman
```

HANGMAN

Gittie Klein

Tamar Neumann

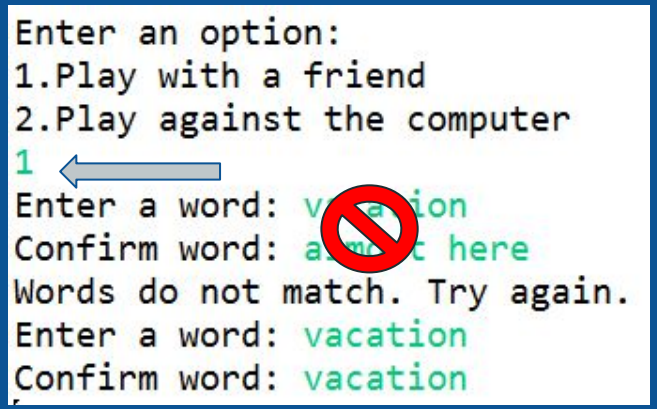

<<enumeration>>
EasyWords
<<enumeration>>
MediumWords
<<enumeration>>
HardWords
GuessWord
wordArray: char[]
word: GenerateWord
guessedLetters: int
wordGuessed: boolean
randomWord()
playerWord()
setWordGuessed()
setLettersGuessed()
createArray()
guesseLetter()
guessTheWord()
getWord()
guessed()
Hangman
guessWord: GuessWord
wordBank: wordBank
hmperson: Person
usePlayersWord()
createRandomWord()
guessLetter()
guessWord()
guessed()
displayWord()
getIncorrectGuess()
getWord()
resetGame()
gameRules()
toString()
Person
int: incorrectGuess
object1: String
object2: String
object3: String
...
setIncorrectGuess()
getIncorrectGuess()
increaseGuess()
getObject()
Generate Word
id: int
numLetters: int
word: String
createRandomWord()
setWord()
getWord()
getNumLetters()
WordBank
lettersLeft: ArrayList<Character>
lettersGuessed: ArrayList<Character>
createWordBank()
removeGuessedLetter()
getLettersLeft()
getLettersGuessed()
printLettersRemaing()
printLettersGuessed()
toString()

GAME RULES

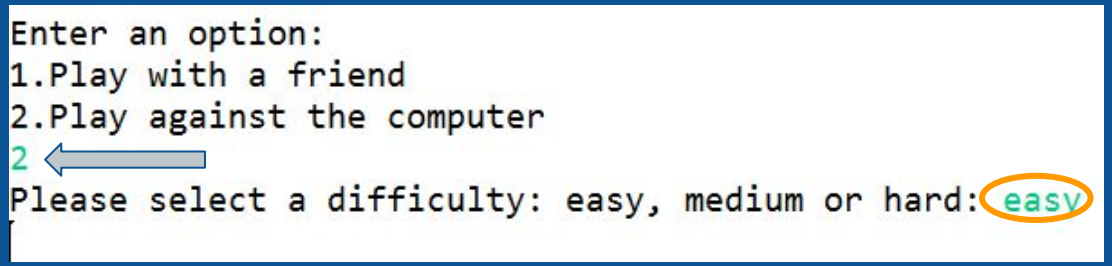
```
Game Rules:
WELCOME TO HANGMAN.
-THE GOAL OF THE GAME IS TO GUESS A WORD, ONE LETTER AT A TIME.
-EVERY INCORRECT LETTER YOU GUESS ADDS A PART TO THE HANGMAN.
-AFTER 10 INCORRECT GUESSES, THE HANGMAN IS COMPLETE MAKING IT A GAME OVER.
-IF AT ANY POINT YOU WOULD LIKE TO GUESS THE ENTIRE WORD, ENTER "0".
```

The player has an option to have another person enter a word, or the computer to choose a word for him.

MULTI PLAYER



SINGLE PLAYER



MULTI PLAYER	SINGLE PLAYER
In Main: `hangman.usePlayersWord(playersWord);`	In Main: `hangman.createRandomWord(input.nextLine());`

Calls the Hangman class:
`public void usePlayersWord(String word)`
`{`
`guessWord.playerWord(word);`
`}`
	Calls the Hangman class:
	`public void createRandomWord(String level)`
	`{`
	`guessWord.randomWord(level);`
	`}`

Which calls the GuessWord class:
`public void playerWord(String wordX)`
`{`
`word.setWord(wordX.toUpperCase());`
`createArray(word);`
`}`
	Which calls the GuessWord class:
	`public void randomWord(String level)`
	`{`
	`word.createRandomWord(level);`
	`createArray(word);`
	`}`

Which sets the word in the GenerateWord class	Which generates a random word in the GenerateWord class

SINGLE PLAYER continued...

GenerateWord Class

```
if(level.equalsIgnoreCase("easy"))
{
    EasyWords w[] = EasyWords.values();
    id = num.nextInt((w.length));
    this.word = w[id].toString();
}
```

- Read in the easy words into an array.
- Generate a random number with the random field "num" that accesses a word at that subscript.
- Locate the enumerated word at that subscript and assign it to the hangman word.

The word for the hangman game is now set

1. Either by receiving a word from the player

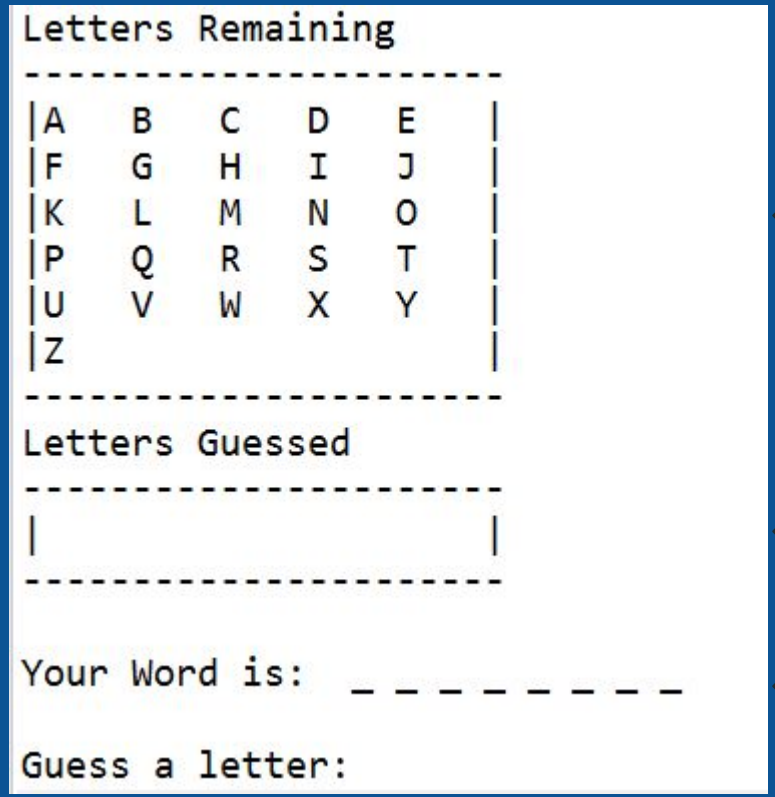
```
public void playerWord(String wordX)
{
    word.setWord(wordX.toUpperCase());
    createArray(word);

}
```

2. Or by generating a random word from an enumerated class based on the choice of level

```
public void randomWord(String level)
{
    word.createRandomWord(level);
    createArray(word);

}
```

An array the size of the word is created and each subscript is assigned an underscore

The game is on!!



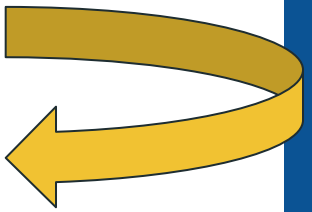
Each round a word box displays all the available letters the player can guess.

Along with a wordbox of all the letters that were guessed already.

The array of underscores is displayed. Every time the player guesses correctly, the underscore is replaced by the correct letter.

```
Guess a letter: d
The letter you guessed is incorrect.

+---------+
|         |
|         |
|         |
|         |
|         |
+----------------+
Letters Remaining
----------------------
|C   F   G   H   J   |
|K   L   N   O   P   |
|R   S   T   X   Y   |
----------------------
Letters Guessed
----------------------
|A   E   I   B   U   |
|M   Q   W   Z   V   |
|D                   |
----------------------

Your Word is:  V A _ A _ I _ _
```

Every incorrect guess adds a part to the hangman



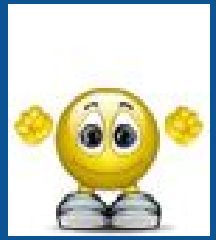
Every letter guessed is removed from the letters remaining and added to the letters guessed Word box

If a letter was already guessed, this error message is displayed, but no turns are lost

```
Guess a letter: a
Invalid guess
```

If at any point in the game the player is ready to guess the entire word, he can enter "0" and is prompted for the word

Since we all know the word is vacation…. (10 more days!!!



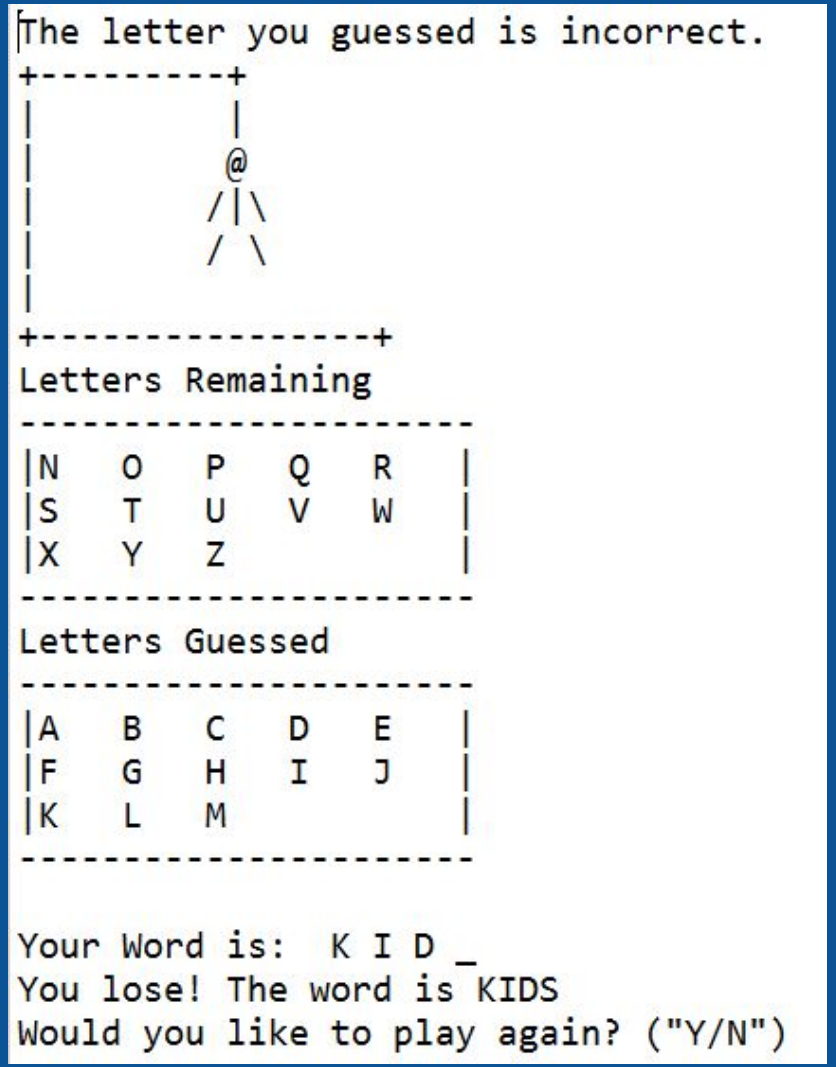
)

```
Your Word is:  V A _ A _ I _ _

Guess a letter: 0
Guess the word:vacation
```

```
Your Word is:  V A C A T I O N
Congratulations you guessed the word!
```

The player has a game over if he guesses 10 incorrect letters, or if he does not guess the word correctly when he enters "0"

At the end of each game, the player has an option to play another game or exit



Challenges

Unit Testing:

- Private fields with encapsulation - hard to have access to test
  For example: we were not able to access the word that the computer randomly generated, so could not test methods based on the word.
- Solution: After creating an option for the user to enter a word, we were able to test our methods based on our own word.

GitHub:

- In the beginning, we made mistakes in the committing/pushing process that made our repository disorganized.
- Solution: We ended up creating a new repository and just pushed our first group of code as one commit.
- There were a number of times that one of us forgot to pull from gitHub before making changes.
  Solution: The easiest/cleanest solution we found was to delete our local project and clone it.

Challenges

Hangman:

- After completing the game, we realized our code was not running properly after the player chose to play another game.
  We quickly realized we forgot to reset our hangman, word banks, and boolean variables that declares if the player gusssed the word or not!
- Solution: In each class, we created methods that reset the correct fields, and called them all from a method in the hangman class.

```
public void resetGame()
{
    wordBank.resetWordBank();
    guessWord.setWordGuessed(false);
    guessWord.setLettersGuessed(0);
    hmperson.setIncorrectGuess(0);
}
```

```
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   src/HangmanGame/HangmanMain.java

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Gittie Klein\OneDrive\Documents\MCO152 - Computer Methodology\Klein_G-mco152-2016F_hangman> git add .\src\
PS C:\Users\Gittie Klein\OneDrive\Documents\MCO152 - Computer Methodology\Klein_G-mco152-2016F_hangman> git commit -m "v
iew hangman after lose"
[master 889d4a9] view hangman after lose
 1 file changed, 1 insertion(+)
PS C:\Users\Gittie Klein\OneDrive\Documents\MCO152 - Computer Methodology\Klein_G-mco152-2016F_hangman> git push
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 524 bytes | 0 bytes/s, done.
Total 5 (delta 0), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/gittieklein/Klein_G-mco152-2016F_hangman.git
   b81aaaf..889d4a9  master -> master
PS C:\Users\Gittie Klein\OneDrive\Documents\MCO152 - Computer Methodology\Klein_G-mco152-2016F_hangman> git add .\src\
PS C:\Users\Gittie Klein\OneDrive\Documents\MCO152 - Computer Methodology\Klein_G-mco152-2016F_hangman> git commit -m "a
lways display to string after end of game"
[master a202d34] always display to string after end of game
 1 file changed, 3 insertions(+), 4 deletions(-)
PS C:\Users\Gittie Klein\OneDrive\Documents\MCO152 - Computer Methodology\Klein_G-mco152-2016F_hangman> git push
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 524 bytes | 0 bytes/s, done.
Total 5 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/gittieklein/Klein_G-mco152-2016F_hangman.git
   889d4a9..a202d34  master -> master
PS C:\Users\Gittie Klein\OneDrive\Documents\MCO152 - Computer Methodology\Klein_G-mco152-2016F_hangman> git pull
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 5 (delta 3), reused 5 (delta 3), pack-reused 0
Unpacking objects: 100% (5/5), done.
From https://github.com/gittieklein/Klein_G-mco152-2016F_hangman
```

HANGMAN

Gittie Klein

Tamar Neumann