MEETING MINUTES (Draft provided by Hasan | Revision by Keming He | Own by The Ohio State University)

Project Name:	Save Brutus! (Hangman)		
Date and Time: (MM/DD/YY HH:MM)	11/07/18 6:15pm	**Location:**	HI 308
Meeting Facilitated by:	All Members	**Documented by:**	Keming

1. Meeting Objective

Previous Meeting Date:	11/05/18 6:15pm
Previous Meeing Objective:	Write algorithm draft for the Hangman game, sign and upload teamworking agreement
Current Meeting Objective:	Finalize pseudo code for the Hangman game and create individual responsibility agreement

2. Attendance at Meeting

Name and OSU Email	Phone	Previous Responsibilities	Completion Status
Keming He.1537	6145586658	(This is a group responsibility for every team member to fulfill) Given the character vector word = [‘b’, ’e’, ’e’, ’n’], create a detailed algorithm/psuedo code that lets player input a guess letter, and determine whether the guess is correct; if correct the corresponding blanks will turn into letters, if not, health value should minus one	Done
Kayla Huff.879	9376942315		Done
Nathan Weltle.2	4193576461		Done
Tristan Langley.67	8583360273		Done

3. Agenda and Notes, Decisions, Issues

Topic	Owner	Time Needed
Complete the pseudo code for the game	All Members	45min
Format and upload the pseudo code to team website, create individual responsibility	Tristan	15min
Create and import all images (and sound) for the game,	Kayla	N/A
Research on useful functions for the game	Kayla and Keming	15min
Test pseudo code and prototype	Nathan	30min

4. Work Distribution

New Responsibilities	Owner	Due Date

5. Next Meeting Schedule

Date and Time: (MM/DD/YY HH:MM)	11/08/18 5:30pm	**Location:**	HI 224
Next Meeting Facilitates by:	All Members	**Documents by:**	Keming
Future Objectives:	Turn the pseudo code into matlab script file and test it		

Appendix A: Additional Meeting Notes

Attached below is the latest version of the Save Brutus! (Hangman) game:

```
PSEUDOCODE (most recent)
11/07/2018

print(welcome to save brutus!)
Pic1 = imread('hanging post.jpeg','jpeg');
>>imshow(pic1,'InitialMagnification','fit')
health = 6
Wordlist = [.......];
Word = random(wordlist);
status = [0, 0, 0, 0]
statusSumBefore = 0
statusSumAfter = 0
//end of initializing variables

while (statusSumBefore <4&&(health > 0) )
{
//Print the word for guessing with already guessed letter displayed and yet-to-be-guessed letters //replaced with
underscore
For (i : length(wordslist))
{
If (statis(i) == 1)
        Print (wordsList(i))
Else
        Print '_ '
}
Check player guess and update word status or health
print(-----instructions-----)
Guess = input(enter a number: )

For i = 1:4
//Use a for loop to check if the player's letter matches with the word array element
If guess == element
        Status(i) = 1
End
statusSumAfter = sum(status)
If statusSumAfter == statusSumBefore
//means that the player's letter did not match any letter
health=health-1
print (lose 1 turn" message)
Else
print (your guess was correct)
statusSumBefore =  statusSumAfter
Time lapse for 3 seconds then clc
} //End of while loop

End Message
If statusSumBefore == 4
        print (congrats, brutus lives to see another football game!)
Else
        print(you are out of guesses. You killed brutus!!!!!!)
End
```

Useful
Rand(): for pulling a random 4-letter word from the list
convertStringsToArray(): self-explanatory