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

Project Name:	AU18 Engr1181 SDP: Hangman Game		
Date and Time: (MM/DD/YY HH:MM)	11/05/18 6:30pm	**Location:**	HI 308
Meeting Facilitated by:	All Team Members	**Documented by:**	Keming

1. Meeting Objective

Previous Meeting Date:	11/01/18 5:30pm at HI 224
Previous Meeing Objective:	Create Teamwork Agreement, create team website, document three consumer interview, determine first game development project
Current Meeting Objective:	Sign Teamwork Agreement, upload agreement and interview document onto team website, write general algorithm for the Hangman Game

2. Attendance at Meeting

Name and OSU Email	Phone	Previous Responsibilities	Completion Status
Keming He.1537	6145586658	Document 11.01 meeting, print Teamwork Agreement for group signature	Done
Kayla Huff.879	9376942315	Document three interviews	Done
Nathan Weltle.2	4193576461	Prepare to lead the Hangman algorithm meeting	Done
Tristan Langley.67	8583360273	Build team website	Done

3. Agenda and Notes, Decisions, Issues

Topic	Owner	Time Needed
Sign the Teamwork Agreement, upload it to website, and take team photo	Keming	10min
Upload three interview recoreds to u.osu.edu	Kayla	10min
Upload Teamwork Agreement to u.osu.edu	Tristan	10min
Create general Hangman Game algorithm	All members	30min

4. Work Distribution

New Responsibilities	Owner	Due Date
(This is a group responsibility for every team member to fulfill)	Keming	11/07/18 6:15pm
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	Kayla	11/07/18 6:15pm
	Nathan	11/07/18 6:15pm
	Tristan	11/07/18 6:15pm

5. Next Meeting Schedule

Date and Time: (MM/DD/YY HH:MM)	11/07/18 6:15pm	**Location:**	HI308
Next Meeting Facilitates by:	All Members	**Documents by:**	Keming or Nathan
Future Objectives:	The team members will each share their own solutions to Step 12 to Step 15; and together the team will write detail Hangman Game algorithms that can be turned into code in MATLAB.		

Appendix A: Additional Meeting Notes

Current Algorithm for Hangman from Google Docs:

One word, one time hangman game, resulting in either a win or a lose
Rules, Known, and Constraints:

1. Health system
2. Word list
3. One letter at a time is guessed

Algorithm:

1. Create vector for alphabet [a, b, c, d, …, y, z]
2. Create word list
3. Set hangman health to full (usually 6)
4. A (4-letter) word is chosen by the computer from the word list- creating a new vector from alphabet vector values
5. Display 4 “fill in the blank” spaces for letters // The same number of spaces as the word length
//While loop
6. Check if word is complete
7. If not complete move on to check health
8. If complete show success message // Go to Ending 1

9. Check if health = 0
10. If health > 0, move on to prompt another input
11. If health=0 , end game (lose) // Go to Ending 2

12. Prompt user to input a letter
13. Check to see if letter is a value in word vector
14. If letter is in word vector, replace its corresponding blank space with the letter // Can replace multiple spaces if the same letter appears more than once. Then go back to check if word is complete
15. If not, deduct 1 health, display “guess again,” and go back to check if word is complete
//end
16. Ending 1: Word is complete while health > 0 - Player wins!
17. Ending 2: Word is incomplete while health == 0 - Player loses!