Weekly Meeting

3/6 Tue - Project Proposal
3/27 Tue - Specifications, Task Breakdown and System Design
4/13 Fri - Test Results and Project Status
4/26 Thu - Final Product

Date	To do	Finished	Prepare for next meeting
2/28	• First meeting • Communicating	• Deciding to use Python 3.6 • Installing GitHub desktop and creating a repository • Writing Deliverable I (Project Proposal) • Registering a study room for next meeting	• Bring the real game • Study the rule • Learn Python Graphics

3/7	• Play the game
	• Gather requirements
	• Design system
	• Task breakdown
		• Played the game
		• File structure
		• Decide to use Pygame
		• Assigned Task
		  ○ Chia-Chun: board
		  ○ Peter: deck
		  ○ Gavin: instruction
		  ○ Joe: main menu window
			• Do the work!

3/21	• 3pm-5pm in room 201
	• Database schema
	• Deliverable 2
		• Discussed what everyone did
		• Wrote deliverable 2
		• Assigned Task
		  ○ Chia-Chun: Board and pawns
		  ○ Peter: Work on deck and start database and database schema
		  ○ Gavin: Breakdown Instruction (GUI), buttons and output
		  ○ Joe: main menu
		• Decided to use top/bottom/left/right instead of colors to represent players
		• Decided to use one array for the track and four arrays for safety zones
			• Main menu GUI
			• Diagrams of major components
			• Database schema
			• Class hierarchy
			• Picture of Sorry!

3/26 Deliverable 2			

3/28	• 3-5 in room 201	• Discussed the position dictionary
		• Describe the files
		• Put game logic in game.py and call Game in setUpBoard
		• Assigned Task
		  ○ Chia-Chun: move()
		  ○ Joe: menu, game pieces be movable
		  ○ Gavin: scanning in images for game, output for game console
		  ○ Peter: saving and DB	

4/4	• database->file
	• Discuss “7” card
	• I quit
	• # computers
	• Keep relaxation start? (instruction)
		• Fixed the git issue
		• Changed deck to be called
		• Fixed the board thing
		• Assigned Task
		  ○ Chia-Chun:force players to follow turns, move backward
		  ○ Joe: finish the menu, get movement working
		  ○ Gavin: put card images
		  ○ Peter: have all information be able to be saved, have database
			• Add current problems in deliverable 3

4/12	• Finish deliverable 3	• Discussed deliverable 3	
4/13 Friday 6pm -- Deliverable 3			

4/18		• Assigned Task:
		  ○ Chia-Chun: move automatically
		  ○ Gavin: Take photo of card
		  ○ Peter: save/resume
		  ○ Joe:
			Quit button
			Save button
			End game
			Draw deck always faces down

4.19	6pm meeting	• Assigned Task:
		  ○ Chia-Chun: Check move, computers move
		  ○ Gavin: Calculate score
		  ○ Peter: Save/resume
		  ○ Joe: End game, stats
		• playing.possibleList = [dic, dic, …]
		• {'option': 1 or 2, 'pawnIndex': 0 to 3, 'move': number, 'destination': positionDict, 'type': track/safetyZone/start/home, 'bumpOther': number, 'bumpSelf': number}
			• save/resume
			• Card 7
			• Check if any pawn can move
			• Make computer move
			• Calculate score
			• End game

4/23	3-5 meeting	• Started testing and found bugs
		• Fixed several bugs
		• Assigned task:
		  ○ Chia-Chun: fix card 7
		  ○ Gavin: testing
		  ○ Peter: fix resume
		  ○ Joe: stats	

4/25	3-5 meeting
	• Quit button
	• Install mysql?
	• Appendix -> whose part
	• presentation
		• Fixed several bugs
		• Keep testing
		• Assigned task:
		  ○ All: finish slides
		  ○ Chia-Chun: instruction to install
		  ○ Gavin: testing
			• Write name in comments
			• Slides
			• Instruction to install
			• Prepare for the presentation!!

4/26	4-6 meeting in Perkins • Practice for the presentation		
4/26 Thursday 6pm -- Final product			

playing.possibleList = [dict, dict, …, dict]

Dict = {

#You might need these

'destination': position #position is a dictionary {'type': xxx, 'side': xxx, 'index': xxx}

'distanceFromHome': integer #current distance from pawn to home

'forward': integer #how many steps it will actually move (forward = final distanceFromHome - original distanceFromHome)

'bumpSelf': integer #how many your pawns will be bumped

'bumpOther': integer #how many opponent's pawns will be bumped

#I keep these for computer moves

'option': 1 or 2 #based on which option

'move': integer #the number it attempts to move

'firstPawn': Pawn #the first selected pawn

'secondPawn': Pawn #the second selected pawn

'Index': integer #Index of this element

}