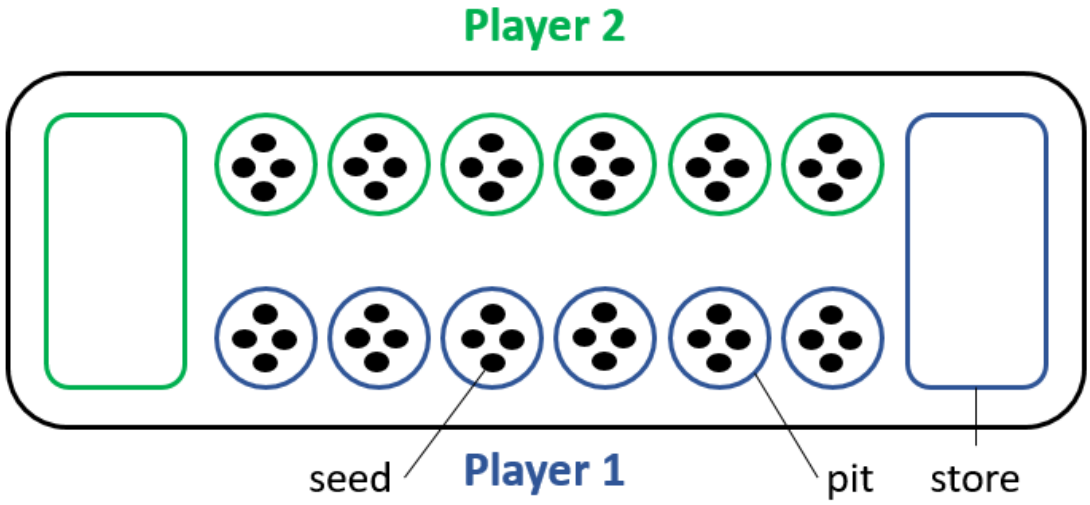
EGME 205 Group Project

Fall 2020: Mancala

Due: Friday, December 4, 2020

Your goal for this project is to program a text-based version of Mancala in Matlab. Although there are many different versions of Mancala around the world, we will be adhering to the most common version in the United States (sometimes called Kalah to distinguish it from the other versions). The rules for this version of Mancala are as follows.



- There are two players. Each player has six (6) *pits* and one (1) *store*. The game starts with four (4) *seeds* in each pit, as shown above.
- On a player's turn, that player picks up all the seeds in one of their pits and moves counterclockwise around the board, dropping one (1) seed in each pit (including their opponent's pits) along the way until they run out of seeds in their hand.

- If a player passes their own store, they place one (1) seed in it. However, if a player passes their opponent's store, they do not place a seed in it.
- If the last seed in a player's hand lands in that player's store, that player takes another turn.
- If the last seed in a player's hand lands in one of that player's pits, and that pit was empty immediately beforehand, that player places the seed in that pit, as well as all seeds in the opposite pit, in that player's store.
- The game is over when either player's pits are empty. At that point, the other player takes all of the seeds remaining in their pits and places those seeds in their store.
- The player with the most seeds in their store at the end of the game is the winner.
- The best way to learn the rules is to play the game. You can play Mancala online at https://www.mathplayground.com/mancala.html.

There are two components to the project:

(a) **(60 points) Basic mechanics** A text-based, two-player version of the game, with all of the necessary game mechanics, is enough to receive the points for this part. Note that, for this part, it is not necessary to include graphics in the figure window (see "Enhancements" below). The code must not allow invalid selections or illegal moves at any point during the game.

(b) **(20 points) Enhancements** In addition to the requirements listed above, you must include at least three enhancements to the game. These enhancements should add an element of originality to your code, making it different from the rest of the groups. Ideally, these will make the game more enjoyable for the player. Some examples of possible enhancements are listed below:

- static graphics in the figure window
- animations
- a clickable game board
- sound effects
- a one-player version of the game with an artificially intelligent computer player
- various difficulty levels for the computer player
- alternative versions of mancala

You must describe your enhancements in a comment at the beginning of the code.

(c) **(20 points) Participation** Each group must give a brief (approximately 5-minute) update on their progress once per week upon request by the instructor. These progress checks will occur at an unspecified time during lecture each week until the project is due. Group members are responsible for being present and ready for each progress check. Each group's participation score will be commensurate with the amount of progress made and the number of group members present at each progress check.

NOTE: All code must be your own, and you may only use the following commands and structures that were covered in this course:

- arrays/matrices
- basic math
- `plot()` and related functions
- Boolean logic and logical operators
- `isequal()`
- IF-statements
- WHILE-loops
- FOR-loops
- characters and strings
- `input()`
- `fprintf()` and `disp()`
- `rand`
- `round()`, `ceil()`, and `floor()`
- `mod()`
- user-defined functions

Use of any other commands or structures will result in a zero grade for the project. If one of your enhancements requires a command or structure that is not listed above, you may request an exception for your group. All requests must be made to the instructor in writing via email at least one week prior to the due date. The instructor reserves the right to refuse such a request for any reason.

Your final submission must be a single M-file named "mancala_groupX.m," where "X" is the number corresponding to your assigned group. Your code will be graded according to the rubric in Table 1.

Table 1: Grading rubric.

Quality	**Score**	**Description**
High	80%	Meets all of the requirements; no bugs/errors; code runs as intended; three or more highly original enhancements
Medium	60%	Meets all of the requirements; no bugs/errors; code runs as intended; fewer than three enhancements or three or more unoriginal enhancements
Low	30%	Meets some, but not all, of the requirements; at least one bug/error that prevents the code from running as intended
Very Low	15%	Does not meet the requirements, but an honest attempt was made
Poor	0%	Does not meet the requirements; very little, if any, attempt was made

Your group's participation score (up to 20%) will be added to this. By default, every member of a group will receive the same score on the project. However, the instructor reserves the right to adjust an individual group member's score to reflect lack of participation. A Peer Evaluation Form is enclosed for students who wish to report dysfunctional group dynamics. Completing this form is optional but may affect an individual group member's score.

EGME 205 Group Project

Peer Evaluation Form

Your Name: _____ Your Group: _____

Please write the names of all of your group members, including yourself, and rate the degree to which each member fulfilled his or her responsibilities in completing the group project on the following scale:

1 2 3 4 5
Did little or no work *Did their fair share of work*

These ratings should reflect each individual's level of participation, effort, and sense of responsibility, not their academic ability.

Name of group member	*Rating*				
_____	1	2	3	4	5
_____	1	2	3	4	5
_____	1	2	3	4	5
_____	1	2	3	4	5
_____	1	2	3	4	5