

Group Project

Mancala

[Github Repo](#)

CSC-17B

Date: 06/11/2023

Josue Carlito

Andrew Guzman

Paul Ingram

Chris Kimoliatis

Rules of Mancala

Set Up:

- 1 Mancala board, made up of two rows of six pits(holes) each.
- 48 colored stones, 4 stones are placed in each of the 12 pits.
- Each player has a designated pit(called a “Mancala”) to the right side of the Mancala board where they collect stones.

Game Rules:

1. The game starts with one player picking up all of the stones in any one of the pits on their side.
2. Moving counter-clockwise, the player places one of the stones in each of the pits until the stones run out.
3. If you run into your own Mancala, deposit one stone in it. If you run into your opponent's Mancala, skip it and continue moving to the next pits.
4. If the last stone you drop is in your own Mancala, you take another turn.
5. If the last stone you drop is in an empty pit on your side, you capture that stone and any stones in the pit directly opposite.
6. The player with the most marbles in their score bank when the game ends wins.

Implementation:

Admin and Log In (Chris) : Developed the login and signup functions with the use of PHP, HTML, Javascript and MySQL. The login and signup pages use queries to access the back-end database to either modify or rewrite the data in each column. Each user is given a unique primary key that is later passed into each page through sessions to allow for modification of the database.

Leaderboard (Josh) : The leaderboard was created using PHP and querying the MySQL server's player data. The leaderboard is then displayed in order of most wins with the player's high score as well.

Game (Andrew and Paul) : The front end interface for the game was made using Javascript. The game board is stored in an HTML table and the player makes their play by clicking one of the buttons below the board. The message indicating who's turn it is then switches to say it is the computer's turn. Then, the computer randomly plays a cell and the computer's played cell is highlighted. Once the game is finished, the user's game stats are updated.

Instructions on How to Run

1. Download project folder
2. Import database dump files in MySQL Workbench
3. Place folder in XAMPP/htdocs folder
4. Run MySQL database and Apache web server

5. Open project in Netbeans
6. Run index.php file

Login Credentials

Admin username: admin **password:** admin

User username: test **password:** 123

Sample Input/Output

Login/Registration

Welcome to Mancala!

[Log In](#) [Sign Up](#)

Log In

[Log In](#)

[Don't have an account? Sign up](#)

Sign Up

[Sign up](#)

[Already have an account? Log in](#)

Mancala Game Board and Leaderboard

3	7	9	3	0	7	0	3
	2	0	0	9	1	8	
1	2	3	4	5	6		

Computer selected cell : 4

LEADERBOARD

USERNAME	WINS
Pingram	4
Jcar	3
Ckimo	2
Aguz	1

12	0	0	0	0	0	0	39
	0	0	0	0	0	0	
1	2	3	4	5	6		

Game Over: jcarlito won

LEADERBOARD

USERNAME	WINS
Pingram	4
Jcar	3
Ckimo	2
Aguz	1

Leaderboard

Rank	Name	Score	Wins
1	test	4	12
2	josh	0	5
3	kimo	4	4
4	jcarlito	41	3
5	paul	2	2
6	andrew	0	0
7	tester	0	0

Username	Password	Score	Wins	Admin Status	Action	Delete
paul	123	2	2	False	Update	Remove
kimo	123	39	8	False	Update	Remove
admin	admin	0	0	True	Update	Remove
mlahr	mlahr	0	0	True	Update	Remove
test	test	37	1	False	Update	Remove


Files Layout

- adminPage.php
- adminStyle.css
- dbh.inc.php
- functions.inc.php
- increment.php
- increment.php
- index.php
- leaderboard.php
- login-inc.php
- login.php
- mancala.php
- mancalaDB.sql
- profile.php
- signup-inc.php
- signup.php
- style.css
- styles.css
- update_session.php

Files Layout

1. Objects - Javascript/PHP - Serialization
 - a. Passing info with Javascript Objects to PHP Objects with Cookies
 - b. \$_SESSION in PHP is an object
2. Reading/Writing Files/Local Storage/JSON - Javascript
 - a. Stringify username
3. SQL Mancala Database

Users SQL Table

	Field Name	Field Type
	users_id	Integer [INT]
	users_name	Text [VARCHAR]
	users_pwd	Text [VARCHAR]
	users_score	Integer [INT]
	users_status	Integer [INT]
	users_win	Integer [INT]

4. Form Validation - Regular Expressions
 - a. Login.php uses regular expressions to validate user input for login credentials
5. User-Admin-Login
 - a. Admin interface implemented. Can edit users' status, remove users from the database, and create other admins.
6. Cookies - Sessions - Securing Pages
 - a. Login and logout functions use PHP sessions.

Flowcharts

