CZ1003 MINI PROJECT: BATTLESHIP

Group members:

Tay Quan Rui
Teh Siang Wen
Uday Sharma

DECOMPOSITION FOR GAME DEVELOPMENT

1. Account creation

- Check if username and DOB are valid
- Check if password is strong

2. Account login

- Check if username and password are correct
- Lockdown mode after 3 attempts

3. Pre-game configuration: placement of ships

- Randomly deploy ships for computer
- Ask user for ship placement

4. Gameplay

- User's turn: ask user for attack
- Computer's turn: randomly generate attack
- Game ends when all ships are sunk

DATA ABSTRACTION

Gameplay board

- Stored as a three dimensional array accessed by array[depth][row][col]
- User enters coordinates as (depth, row, column) with simplified index

Implementation of button

- Not what we think of a button intuitively
- Changes color when mouse is in a certain range of coordinates, cannot “detect” mouseover

DATA STRUCTURES

1. Three dimensional array (list)

- Store game boards
- Accessed by array[depth][row][col]
- False if no ship, True if ship, "hit" if ship is tagged

2. Tuple

- Used to store color RGB values
- Format: (red, green, blue)

3. Dictionary

- Used while reading data from file
- Key: username
- Value: (password, date of birth)

PATTERN RECOGNITION

1. Account creation and login

- Options are mutually exclusive
- User data file is shared
- Incorrect input will require re-entering

2. Pre-game configuration

- Carrier should not be placed in subsea
- Ship body should not exceed playing board
- Ships should not overlap

3. Gameplay

- Turn is swapped between player and computer until the game ends
- Gameplay flow almost the same, except computer randomly generates coordinates

PATTERN RECOGNITION

Certain Patterns can be determined:

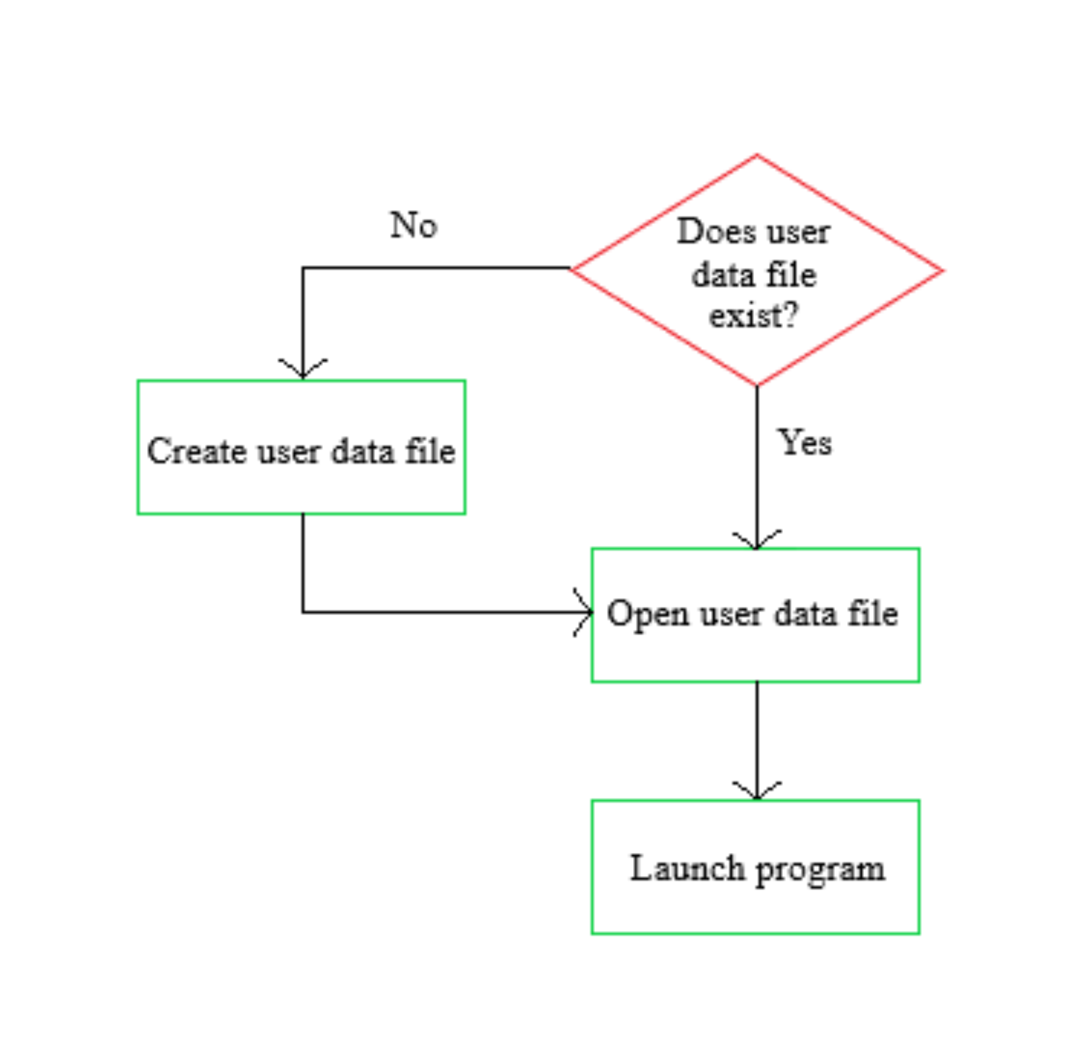
1) For incorrect inputs, an if-elif-else loop is used to prompt for resubmission of input
2) Certain data structures are shared across different sub-components
   - File management can be used to handle the shared data
   - Will work similarly to a dictionary
3) Due to identical mechanisms in the gameplay, it can be coded into functions that can be called upon for similar cases
4) Gameplay process is repeated until one player wins
   - A sentinel controlled loop can be used

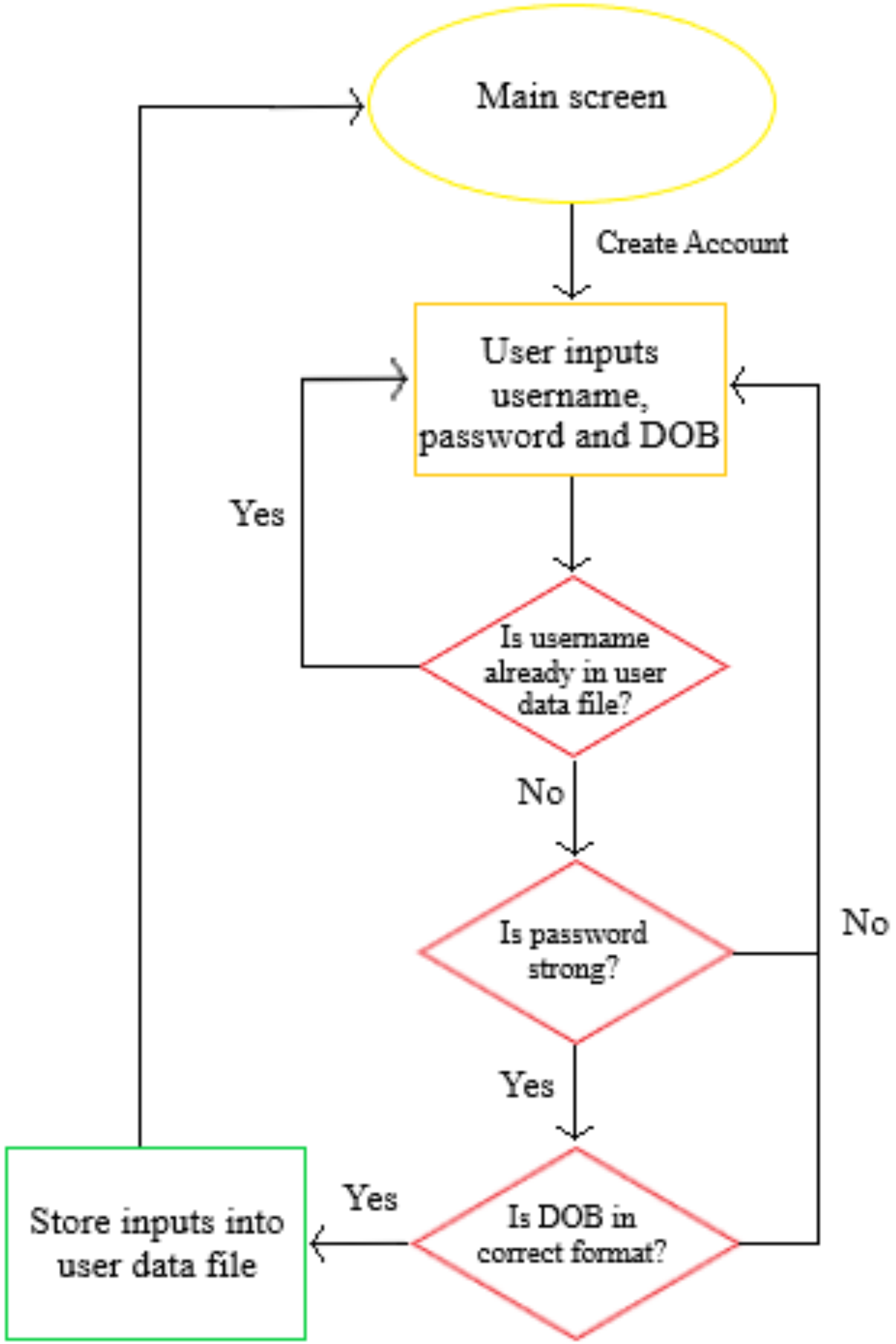
ALGORITHM DESIGN

Use the strategy of "Divide and Conquer"

- Each sub-component will have its own conceptual design to complete its primary task
- Completion of a sub-component's task will result in continuation to the next sub-component

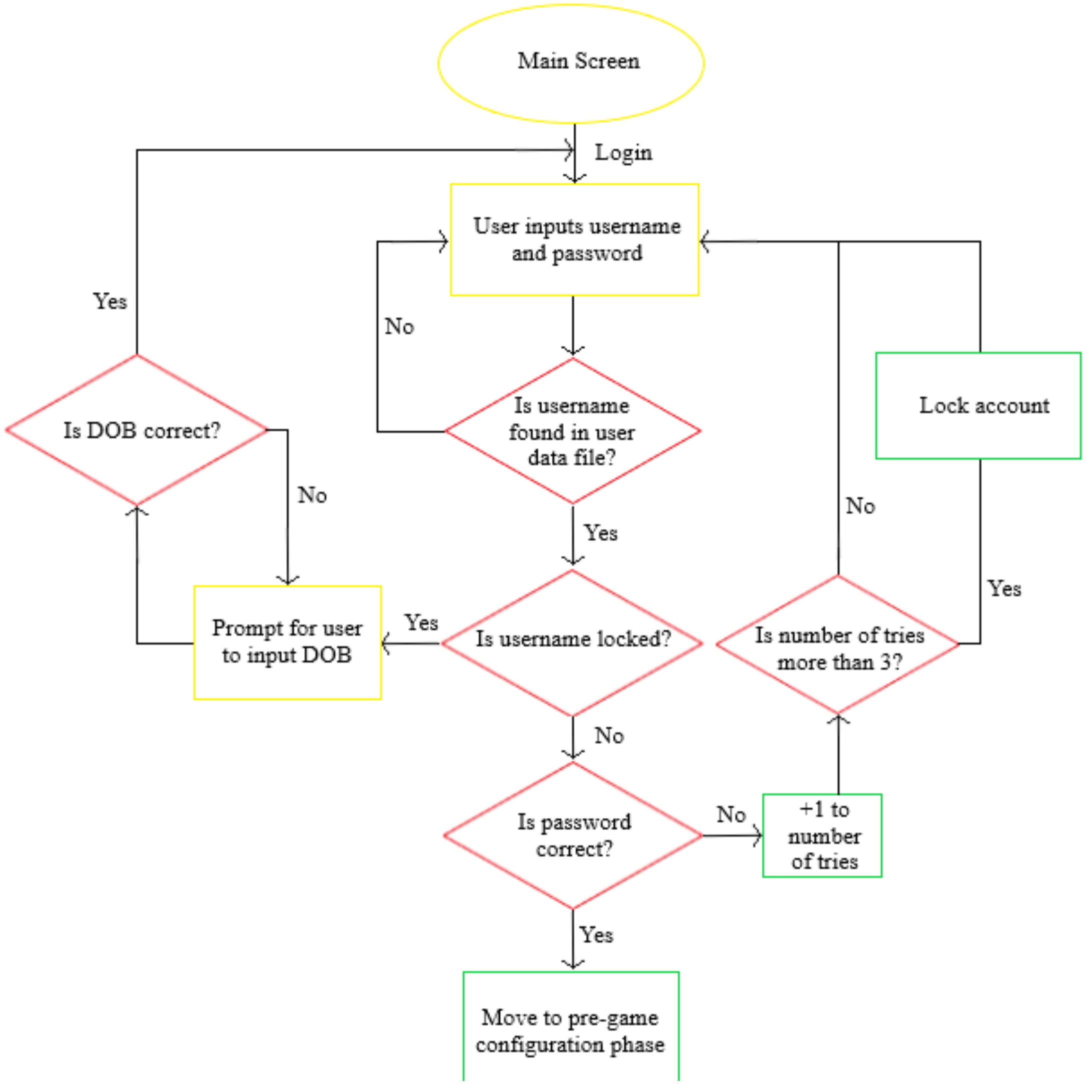
PRE-PROGRAM LAUNCH

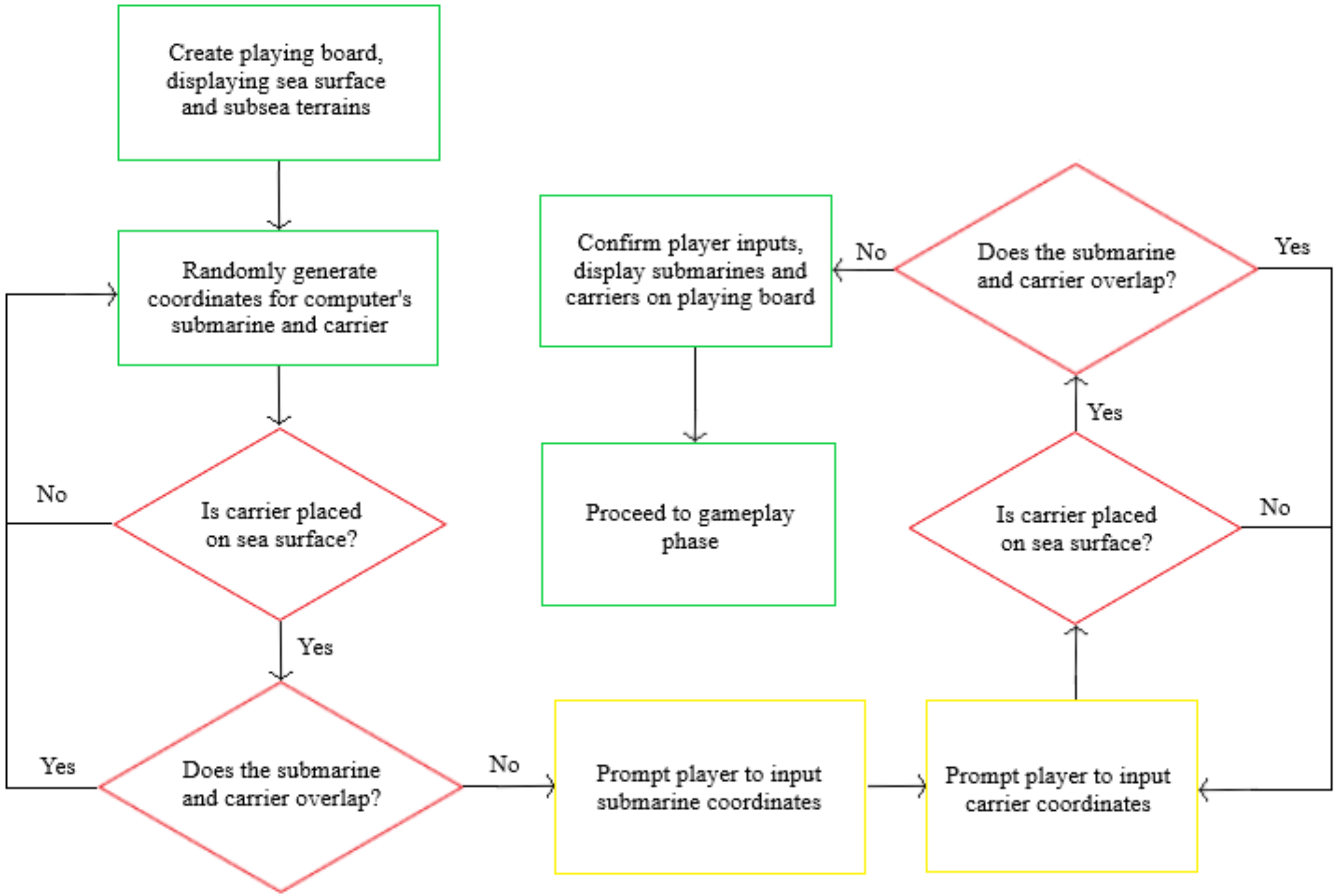



Main screen
Create Account
User inputs username, password and DOB
Yes
Is username already in user data file?
No
Is password strong?
No
Yes
Yes
Store inputs into user data file
Is DOB in correct format?

ACCOUNT CREATION

ACCOUNT LOGIN


Main Screen
Login
User inputs username and password
Is username found in user data file?
No
Yes
Is username locked?
Yes
Prompt for user to input DOB
Is DOB correct?
Yes
No
No
Is password correct?
No
+1 to number of tries
Is number of tries more than 3?
No
Yes
Lock account
Yes
Move to pre-game configuration phase


Create playing board, displaying sea surface and subsea terrains

Randomly generate coordinates for computer's submarine and carrier

Is carrier placed on sea surface?

No

Yes

Does the submarine and carrier overlap?

Yes

No

Prompt player to input submarine coordinates

Prompt player to input carrier coordinates

Is carrier placed on sea surface?

No

Yes

Does the submarine and carrier overlap?

Yes

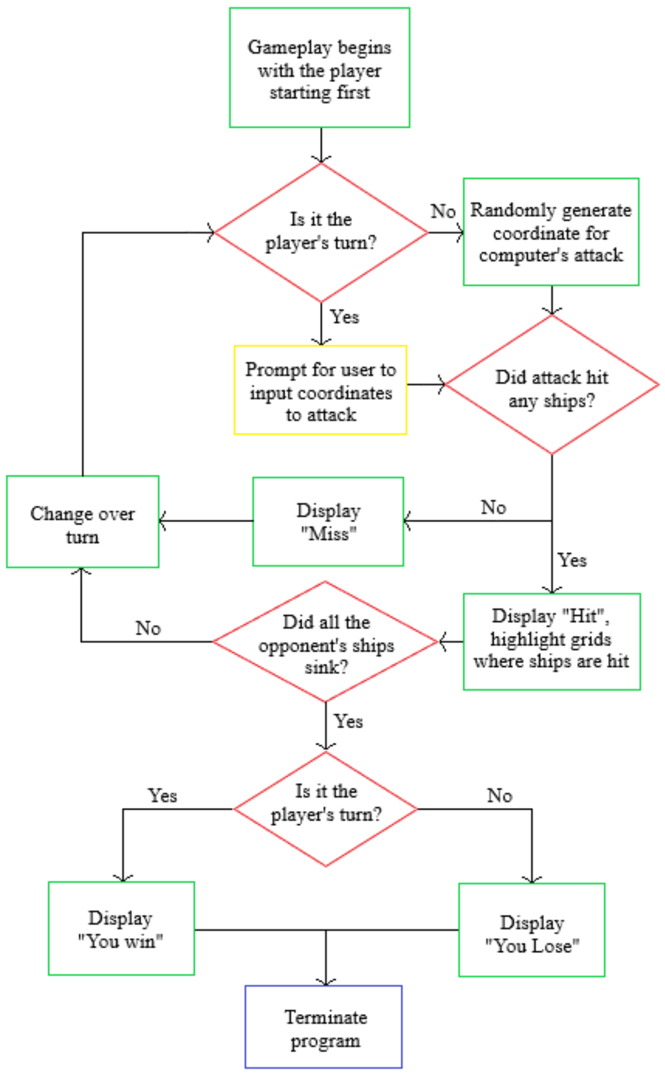
No

Confirm player inputs, display submarines and carriers on playing board

Proceed to gameplay phase

PRE-GAME CONFIGURATION

GAMEPLAY


Gameplay begins with the player starting first
Is it the player's turn?
No
Randomly generate coordinate for computer's attack
Yes
Prompt for user to input coordinates to attack
Did attack hit any ships?
No
Display "Miss"
Change over turn
Yes
Display "Hit", highlight grids where ships are hit
Did all the opponent's ships sink?
No
Yes
Is it the player's turn?
Yes
No
Display "You win"
Display "You Lose"
Terminate program

SYSTEM FEATURES

1. Account creation

- User can create a login account
- Password must meet minimum criteria
- Extra security with date of birth

2. Login and account safety

- Login with username and password
- Lockdown mode if password entered incorrectly three times
- Must enter date of birth to exit lockdown

3. Ship placement

- Two ships: 1x carrier (4 units), 1x submarine (3 units)
- Computer randomly places its ships
- User places ships by entering coordinates
- Check for ship overlap or incorrect coordinates

4. Gameplay

- User attacks by entering coordinates
- Check for incorrect coordinates or already attacked
- Randomly generate computer's attack
- Game ends when one side's ships have sunk

GROUP CONTRIBUTIONS

UDAY SHARMA

- Gameplay
- User interface design

TEH SIANG WEN

- Login and user creation
- Framework for the code

TAY QUAN RUI

- Program flow design
- Report design

THANK YOU