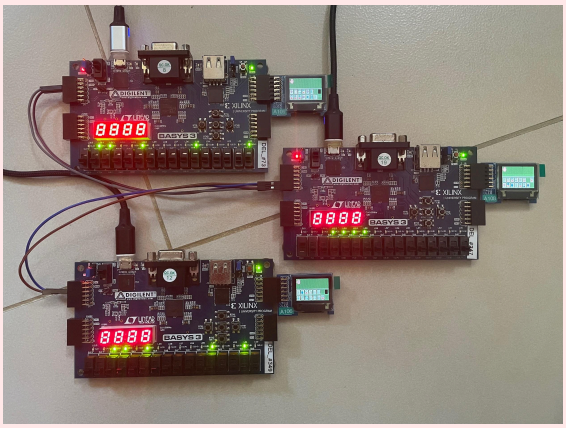
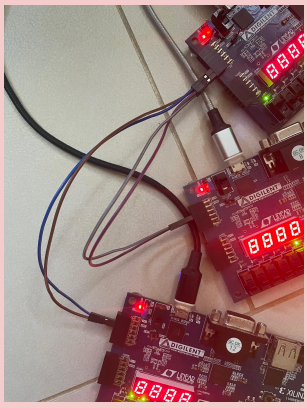
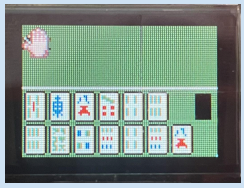
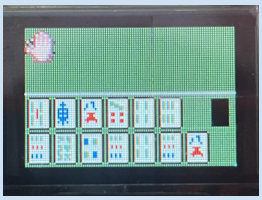
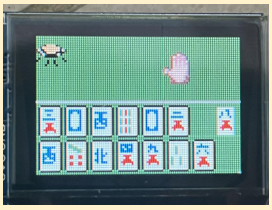
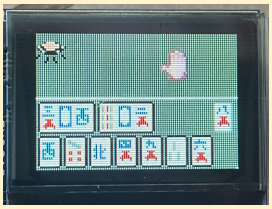
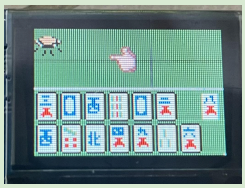
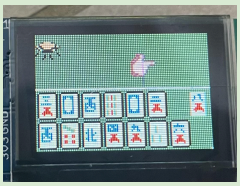
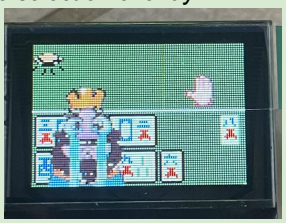
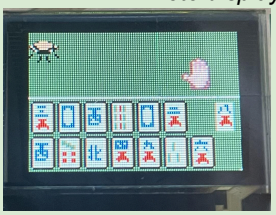
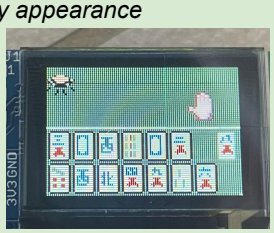
Group ID: S1_03
Student A: Brandon Kang (A0272866N)
Student B: Roderick Kong Zhang (A0286550Y)
Student C: Chua Feng Yuan (A0272146H)
Student D: Tong Jing Yen (A0288445N)

PERSONAL AND TEAM IMPROVEMENTS		
Student and Improvement Name	**Improvement Description**	**Images/Photos**
Team Chinese Mahjong	This project involves creating a multiplayer Mahjong game where each of the 4 players are controlling their hand using their own Basys 3 board. The rules of the game are the same as in traditional Chinese Mahjong, with the exception that *flower* and *animal* tiles will be excluded from the game. The OLED screen on each of the player's Basys 3 boards will display their own hand along with the drawn tile if it is their turn. Players will wait for their turn to make their move, and they can select which tiles to discard by turning on the switch corresponding to the desired tile.	
Student A: **Brandon Kang** Serial Communication via UART, Integration with Game Logic	Serial communication is achieved via UART using a baud rate of 9600 bps and a frame format of 8N1. Data is sent in the form of packets, either through PACKET_TYPE_TILE, PACKET_TYPE_MOVE, or PACKET_TYPE_TURN, depending on what type of data is being transmitted. Depending on the packet type of an incoming packet, the basys3 board would know how to deserialize the packet to obtain the relevant information. When the program is loaded, one board would have to have SW15 turned on. That board would be the master board while the other boards are the slave board. By pressing BTNC on the master board, it will shuffle all the mahjong tiles and distribute each player's hand respectively. This action consists of sending multiple packets of type PACKET_TYPE_TILE to the slaves with each packet serialised in one tile. These packets are sent until all players have 13 tiles in their hand. The game automatically starts and the master board starts first. For each player's turn, they will automatically draw a tile from the master who holds the deck, and discard a tile to the master. The player chooses the tile to discard and press BTNC to remove that tile and send it to the master to discard.	  Wire connections between master and slave boards using port JA.
Student B: **Roderick Kong** Game Logic, Integration with UART	The game logic provides the backend functions and data management for the Mahjong game to be played. Each unique tile is assigned a unique index from 1-36. Upon initialisation of the program, when btnC is pressed, all tiles in the 136-tile array will be shuffled using the Fisher-Yates shuffle algorithm, and 13 tiles will be dealt to each board. As a result, the OLED display on each board will display the 13 tiles dealt to each player. During each player's turn, an additional tile is added to their	  *Initial hand* *Hand after receiving tile*

	hand and shown on their board. After selecting the tile they wish to discard by turning on one switch from sw[13:0] and pressing btnC to confirm their choice, the tile will be discarded and disappear from their OLED screen. After the current player discards a tile to end their turn, there will be a 5-second window for other players to *pon/chi/kan*. If a player does so, the discarded tile will be added to their hand, and it becomes their turn. Otherwise, the program switches to the next player in order. The current player is indicated on led[15:14].	 *Hand after discarding tile boxed in red*
Student C: **Chua Feng Yuan** Implementation of Visuals and design	Manipulation of the running index counter of the PMOLED to acquire the corresponding pixels to display on the screen for each tile type. This design is shown within tiles that are arranged by the types of sets of tiles made through the move actions done after another player discards a tile. When the user clicks on the left mouse button, the layout will change, and utilising the arrangement mechanism implemented by Jing Yen, the player would then be able to group specific tiles together after obtaining them, and be able to manage their hand in this way. Also, when the user 'selects' their tile in this way, the selected tile will also blink at a rate of 1Hz.	  *Layout with no groupings* *Layout with 1 grouping*  *Layout with 2 groupings*
Student D: **Tong Jing Yen** Implementation of UI/UX and additional features	After the initialization of the program, the user is able to perform 'gestures' by triggering certain conditions. For these, the user would be using an attached mouse and holding right click before carrying out one of three actions, dragging the cursor to the left, right or in the path of a tick (a short diagonal downwards to the right, then a longer diagonal upwards to the right). The cursor image rendered is dynamically selected based on the current state of user interaction. The left gesture corresponds to the left pointing cursor, the right gesture to the right pointing cursor. On holding the right click, it turns into a clenched fist and otherwise, an outstretched palm. Furthermore, upon clicking the middle mouse button, a translucent selection overlay will pop up such that the user will be able to choose between a 'laughing' or 'crying' emote, by pressing the middle mouse button again on the highlighted area, a short animation will play at where the image will move up and down at a section of the screen. The user is able to manage their hand by moving each individual tile left or right across the rows. After selecting the tile to move, where sw[0] corresponds to the first tile or the bottom-rightmost tile, and sw[12] corresponds to the top-rightmost tile, the user can press btnL or btnR to shift the selected tile to the left or right respectively. Other inputs include selecting tiles to discard during the player's turn using the switch corresponding to the tile in hand. Finally, there is an eye icon at the top left of the screen. When the user moves the cursor over it, it will start blinking.	  *Cursor during left and right gesture*  *Translucent emote selection overlay*   *Emote display appearance*   *Tile swapping using switches (bottom left tiles)*

References

Shuffle a given array using Fisher–Yates shuffle algorithm. GeeksforGeeks. (2022, December 19). https://www.geeksforgeeks.org/shuffle-a-given-array-using-fisher-yates-shuffle-algorithm/

AI Declaration

Our group, S1_03, consisting of group members, Brandon Kang, Roderick Kong Zhang, Chua Feng Yuan, and Tong Jing Yen, declares that we have not used AI for the creation of our project.