PROJECT IMPROVEMENT PROPOSAL DOCUMENT	
Group ID: S1_03	**Description**
Group Theme: 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. Basic Rules of Traditional Chinese Mahjong: To start the game, 144 downward facing tiles are shuffled. From these tiles, each player draws their hand of 13 tiles. At each turn, the player draws a tile and discards a tile from their hand. The primary objective of Mahjong is to build a complete hand of 14 tiles. A player wins by forming a hand of four sets of three, plus a pair of identical tiles. 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 most recently discarded tiles on the table. Players will wait for their turn to make their move, and they can *pon*, *chi*, *kan*, and discard tiles via either 1) turning on the switch corresponding to the desired tile, 2) using the scroll-wheel on a mouse to highlight and select a tile. Throughout the game, there will also be machine-assisted decision making, such as the ability to take a discarded tile based on conditions of matching a three-of-a-kind, a four-of-a-kind, or a set of consecutive numbers being formed (that may be toggled on or off but will still be used for validation). The game concludes when one player has a hand that fulfils the winning condition, similar to that of regular Mahjong.
Group Member A: **Brandon Kang** Serial Communication via UART	Since Mahjong is a 4 player game, it would require 4 Basys 3 boards to be able to communicate with one another to transmit the game state and a common display to all 4 OLED screens. As such, serial communication is required to send data from one board to another board. **<u>Milestone 1: Basic UART Communication (Week 9)</u>** Set up UART communication between 2 Basys 3 boards with a baud rate of 115200 bps and 8N1 frame. The 8 bit data contains the status of LEDs 0 to 7. Establish a master-slave configuration between the boards, where turning on switches 0 to 7 on the master will turn on the respective LEDs on the slave. **<u>Milestone 2: Data Packet Design and Transmission (Week 10)</u>** Design a structure for the data packets, including the start bit, data payload such as OLED data and game data, and parity bit. Implement the encoding and decoding of these packets within the UART modules. Test sending and receiving game data between two boards.

	<u>Milestone 3: Multi-board Communication (Week 11)</u> Modify the UART modules to handle communication with multiple boards using a master-slave configuration. Implement a protocol that ensures that data is sent to the correct board. Test communication between all 4 boards, ensuring that all data is transmitted correctly. **<u>Milestone 4: Optimisation and Final Testing (Week 12)</u>** Integrating with other game functions and optimising transmission to ensure smooth real-time performance.
Group Member B: Roderick Kong Zhang Game Logic	The game logic is crucial for game-related projects such as Chinese Mahjong, and provides the backend functions and data management for the game to be played. **<u>Milestone 1: Proof of Concept (Week 9)</u>** Devise a system for labelling and identifying each unique tile in Mahjong, and create an algorithm for shuffling (randomising) the tiles before handing them out to each player. Begin working on the logic for machine-assisted decision making, specifically the ability to identify whether a discarded tile from other players can be taken based on conditions of matching a three-of-a-kind, a four-of-a-kind, or a set of consecutive numbers being formed. The proof of concept should showcase a demo of a randomised set of tiles in the hand of a player as well as a random discarded tile from another player. The game logic should identify whether the tile can be taken using *pon* (3 same tiles), *chi* (3 consecutive tiles) or *kan* (4 same tiles). **<u>Milestone 2: Core Features (Week 10)</u>** Implement the following core features that are required for a working version for gameplay: ● Turn-based system ● Database to store tiles that are to be handed out, tiles to be discarded, and each player's hand ○ Ability to add or remove tiles for each array ● Logic for game setup ○ Correctly identifying the *wind* and starting player ○ Shuffling (randomising) all the tiles and allocating the required number of tiles for each player based on the *wind*, and the rest to the drawing pile ● Functions for each player during their turn ○ Drawing of a tile from the drawing pile/taking a discarded tile from other players by *pon*, *chi*, *kan*, or if it is the winning tile ○ Manually declare if the player has won ○ Discard a tile from the hand at the end of the turn if the player has not won **<u>Milestone 3: Integration with UI/UX and Extension Features (Week 11)</u>** Integrate the core logic with UI/UX features to begin preliminary

	testing. Implement the following extension features that enhance the gameplay by introducing innovative and useful additions in order of priority: ● Sorting of tiles in the player's hand ● CPU to simulate real player(s) if there are insufficient players ● Implementing the winning condition to check if a player has won ● Adding *flower* and *animal* tiles ● Counting the score for each player for the round **Milestone 4: Optimisation and Final Testing (Week 12)** Debug any errors, integrate with other game functions and optimise algorithms if needed to ensure smooth real-time performance.
Group Member C: Chua Feng Yuan Designing of tiles and interfaces	The tiles are to be designed fit for the 96 x 64 OLED screen, which would also include various regions for which information is provided for the player such as the current tile to be handled, the area for which tiles involved in "matching" are stored. The overall look presented on the screen should also provide the user with an experience as close to authentic as possible. **Milestone 1: Designing of Each Tile for Fitting** The identity of each tile must be close enough to the actual design to ensure that the user is able to correctly identify what tiles are present in their possession. We are opting for a way to have a certain section of tiles be magnified or miniaturised as the user scrolls through their hand. To that end, a 'magnified, 'normal' and a 'small' version of each tile will have to be made. **Milestone 2: Arrangement of Necessary Verilog Code** Each tile will have its own set of pixel colouring to be formed, and these are to be set-up on Verilog. The adjustment of the hand such that the tiles are correctly represented based on current game state and user inputs **Milestone 3: Integration of Design with Game Logic** The display will at this point be updated to follow the flow of the game, and other guiding designs might be added here to assist players. **Milestone 4: Optimisation and Final Testing** Further work on ensuring accurate representation of user inputs are to be done here.
Group Member D: Tong Jing Yen Input methods for gameplay and user interface	In addition to the buttons and switches on the Basys 3, we are provided a mouse for use as an input device. Leveraging the variety of actions that can be taken by the player, each button and switch will be assigned to different functions in-game. The scroll wheel and gesture input methods will be integrated as well.

	Milestone 1: Switches and Buttons of Basys 3 (Week 9) Basys 3 inputs will be mainly used for rearranging the mahjong tiles in the player's deck. ● Each switch (sw[15:0]) corresponds to one of the player's tiles on the screen. A tile is selected when its switch is toggled HIGH. ● The left and right push buttons are used to rearrange the mahjong tiles selected through the switches. ● The centre push button is used to finish the rearrangement of selected mahjong, or execute other functions during the player's turn. A system to track the position of each mahjong tile in the player's deck will also be created to keep track of what needs to be displayed. **Milestone 2: Mouse Buttons and Scroll Wheel (Week 10)** The mouse is the main input device when it is the player's turn to play. We aim for all input methods of the mouse to be used. For Milestone 2, we aim to set up the left and right buttons, and the scroll wheel. The latter input method was not taught in class and will be researched on and implemented. ● Left button is used to receive a mahjong tile into the player's deck. ● Right button is used to discard a tile from the player's deck. ● The scroll wheel is used to browse through the player's deck when it is not their turn, or to pick the tile to receive or discard during their turn. **Milestone 3: Handling Mouse Pointer and Gestures (Week 11)** Implement the mouse pointer input method to obtain the movements of the mouse. The coordinates can be used to control the position of an on-screen pointer, or used for gesture detection. This input method will be implemented as an enhancement to the existing mahjong game input system, or as part of the user interface. **Milestone 4: Testing and Optimisation (Week 12)** Integration with other game functions, especially the game logic and display function, to ensure expected behaviour and smooth operation.