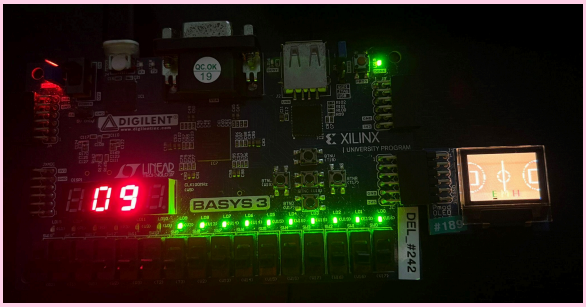
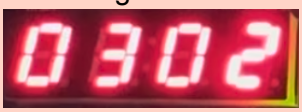
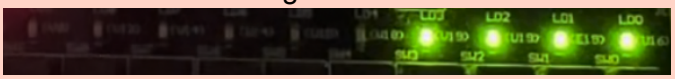
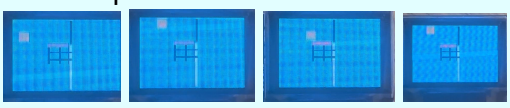
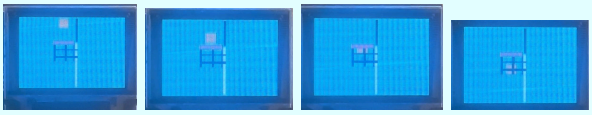
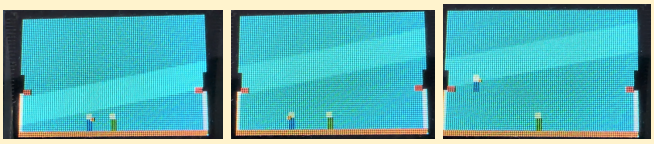
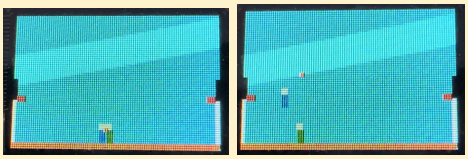
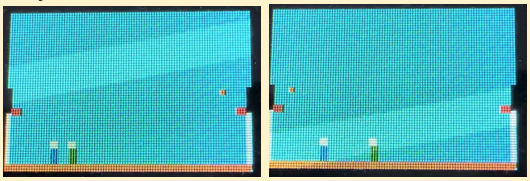
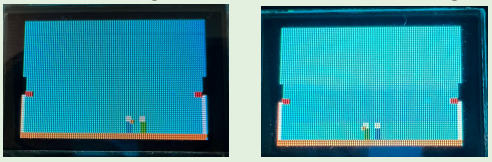
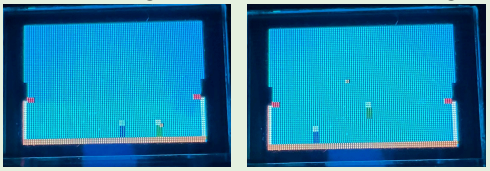
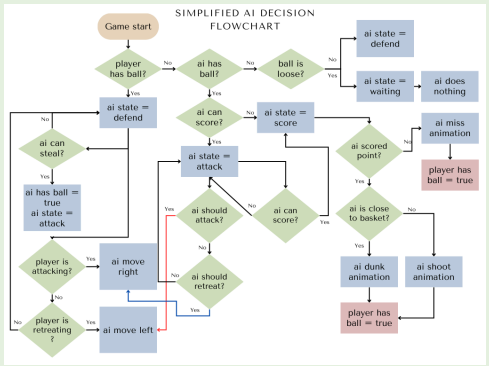
Group ID: 13
Lavan Aiman Sumanan (A0291113R), Andre Villanueva (A0290504L),
Neo Min Wei (A0252595W), Chua Qing Wei Ryan (A0254458W)

PERSONAL AND TEAM IMPROVEMENTS		
Student and improvement name	**Improvement description**	**Images/photos**
Team "Basketball Jam"	Basketball Jam is a 2D basketball singleplayer game which involves a player versus a bot. The difficulty level of the bot includes easy, medium and hard. The player has the ability to move left and right, jump, shoot, and steal the ball from the bot. The score is tracked on the 7-segment display during the game, and highscores are shown on the display on the start and end screen. The led will turn off one at a time to indicate that the game is coming to an end. Each game will last 60 seconds. The bot has the same capabilities as the player. Whenever the player/bot attempts to score, an animation will be played to indicate a score/miss.	
Student A: Lavan Aidan Sumanan "Score tracking, timer, randomisation of scoring"	**Score tracking:** High score is displayed on the 7-segment display at the start and end screen. During the game, player's score is tracked on the two leftmost anodes, and AI's score is tracked on the two rightmost anodes. **Game timer:** During the game, LED[9] to LED[0] are on. In the final 10 seconds, LEDs turn off one by one each second to signal the game's end. **Randomisation of scoring:** The likelihood of scoring depends on the player's distance from the basket, using a 12-bit LFSR for randomness. Shots closer to the net have a higher chance of success.	High score  Score in game  Countdown timer using led 
Student B: Andre Villanueva "Animation, start and end menu"	**Start Screen:** The start screen shows a basketball court from the top view. The AI difficulty level can be selected using **btnL** and **btnR**. Press **btnD** to finalise decision. **End Screen:** Game Over displayed on end screen. Either a W or L will appear to indicate whether the player has won or lost the game. Press **btnD** to go back to the start screen. **Animations:** 3 animations sequences are available in the game, a **miss**, **score** or **dunk** sequence, to indicate the player/AI's scoring result.	 Scoring Sequence  Miss Sequence  Dunk Sequence

	Court Model & Player Model: The court and players are displayed on the screen after the game has started. (see images below)	
Student C: Neo Min Wei "Player movement, ball movement, gravity"	The player moves based on the **btn** inputs on the basys3 board. The controls are as follows: **btnL**: player moves **left** **btnR**: player moves **right** **btnU**: player **jumps** **btnD**: player **steals** the ball from the AI if the player model is in contact with the ball **btnC**: player **shoots** the ball towards the basket Based on the player's or AI's position when shooting the ball, the **arc** of the ball movement will **adjust** accordingly such that it always **reaches their respective basket**. **Gravity effects** can be observed when the player is jumping and in the movement of the basketball when it is shot	btnR: btnL: btnU:  btnD: btnC:  Player ball arc: AI ball arc:  (reaches right basket) (reaches left basket)
Student D: Chua Qing Wei Ryan "AI opponent"	The AI has 3 difficulty levels: easy, medium, hard. Based on the **difficulty** level, the following parameters are adjusted: Shooting range, attack range, defend range, aggressiveness, steal chance, score chance, max jump height, speed. The **aggressiveness**, probability of AI making a successful **steal**, and probability of AI **scoring** is determined using a **32-bit LFSR**. The AI also has 4 main states: **defending**, **attacking**, **shooting**, and **waiting**. The actions and decision making of the AI can be seen in the simplified AI Decision Flowchart.	AI defending AI attacking/steal  AI retreating AI shooting  Simplified AI Decision Flowchart 

References:

Feedback bits for maximum length of 32-bit LFSR -
https://www.ijera.com/papers/Vol4_issue6/Version%206/P0460699102.pdf