

Brief Design Report Group 12

The Design Specification:

The objective of this mini project is to design a simple game console implementing the Flappy Bird game on the DE0-CV board. The game will be controlled using a PS/2 mouse for bird movement (left-click to flap), while DIP switches and push buttons will handle mode selection and game controls. The visual output will be displayed on a VGA screen with a resolution of 640×480 pixels.

The game features two distinct modes: Training Mode and Game Mode. When the system powers on, players are greeted with a title page displaying both mode options. Mode selection is performed using DIP switch SW0, with confirmation through push-button KEY0. Training Mode offers players unlimited practice time at the easiest difficulty setting, allowing them to hone their skills without level progression. Game Mode presents a more challenging experience with five progressively difficult levels, each introducing new gameplay elements.

The core gameplay revolves around navigating the bird through obstacles while managing three available lives. Lives are displayed prominently on-screen, decreasing by one with each obstacle collision. Ground impacts result in an immediate game over. The five levels feature distinct characteristics: Level 1 serves as an introductory stage with constant speed; Levels 2 and 3 introduce gifts alongside moderate speed increases; Level 4 removes gifts while further increasing speed; and Level 5 presents the ultimate challenge with maximum speed, no gifts, and additional obstacle types.

During gameplay, KEY1 provides the pause and resume functionality. Upon game over, the screen displays bold "GAME OVER" text and options to restart the current game or return to the main menu. These selections are made using dedicated switches and push buttons, ensuring intuitive navigation. The visual interface maintains clear indicators for lives remaining, current level, and score, providing players with constant feedback on their performance.

The core gameplay mechanics include a scrolling screen that moves from right to left at a constant speed, with the speed increasing with each level. Obstacles such as pipes appear at random intervals, and power-ups/gifts will be present (i.e, medicine) to give the bird a life/give it an extra advantage. The bird is controlled through the PS/2 mouse, with a left-click making it flap upward; otherwise, gravity would pull the bird down, making the bird drop and end the game. Collision detection ensures the bird loses a life upon hitting obstacles, and the game ends when all lives are depleted.

The system is divided into three main modules: Input Control (mouse, switches, buttons), Game Logic (physics, collision, level progression), and Display Output (VGA display). A finite state machine (FSM) governs the game flow, transitioning between the main menu, active gameplay, and game-over states.

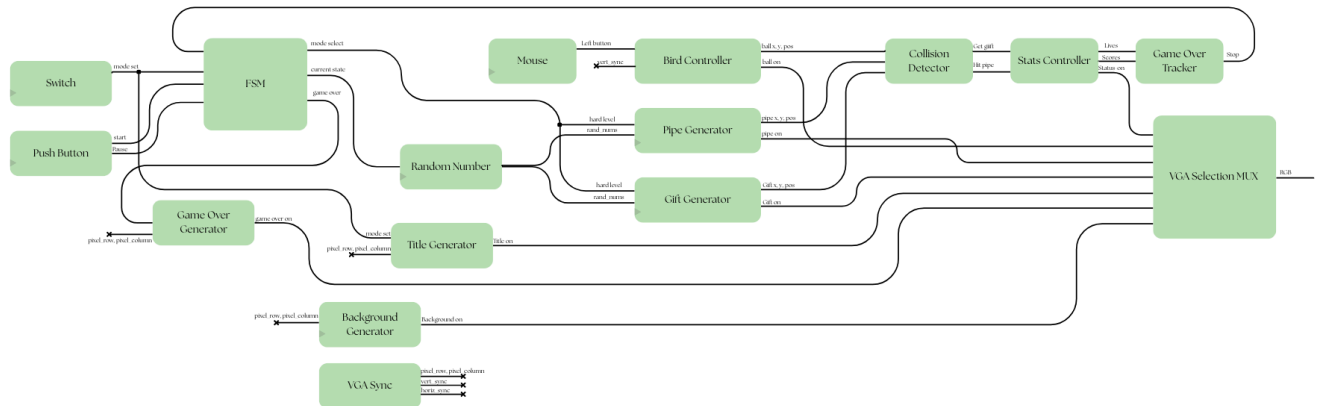
Game Strategy and Plans:

As of now, we as a group have finished the first section of the project, the Interim Interview, and are now beginning the next phase of the game's production alongside the Final Interview. After the interim interview has been completed, we plan to continue working on the assigned programming tasks (shown in the Gantt chart) alongside the preparation of the final interview, which would help each of us individually to understand the overall system in complete depth.

[Link to Gantt Chart](#)

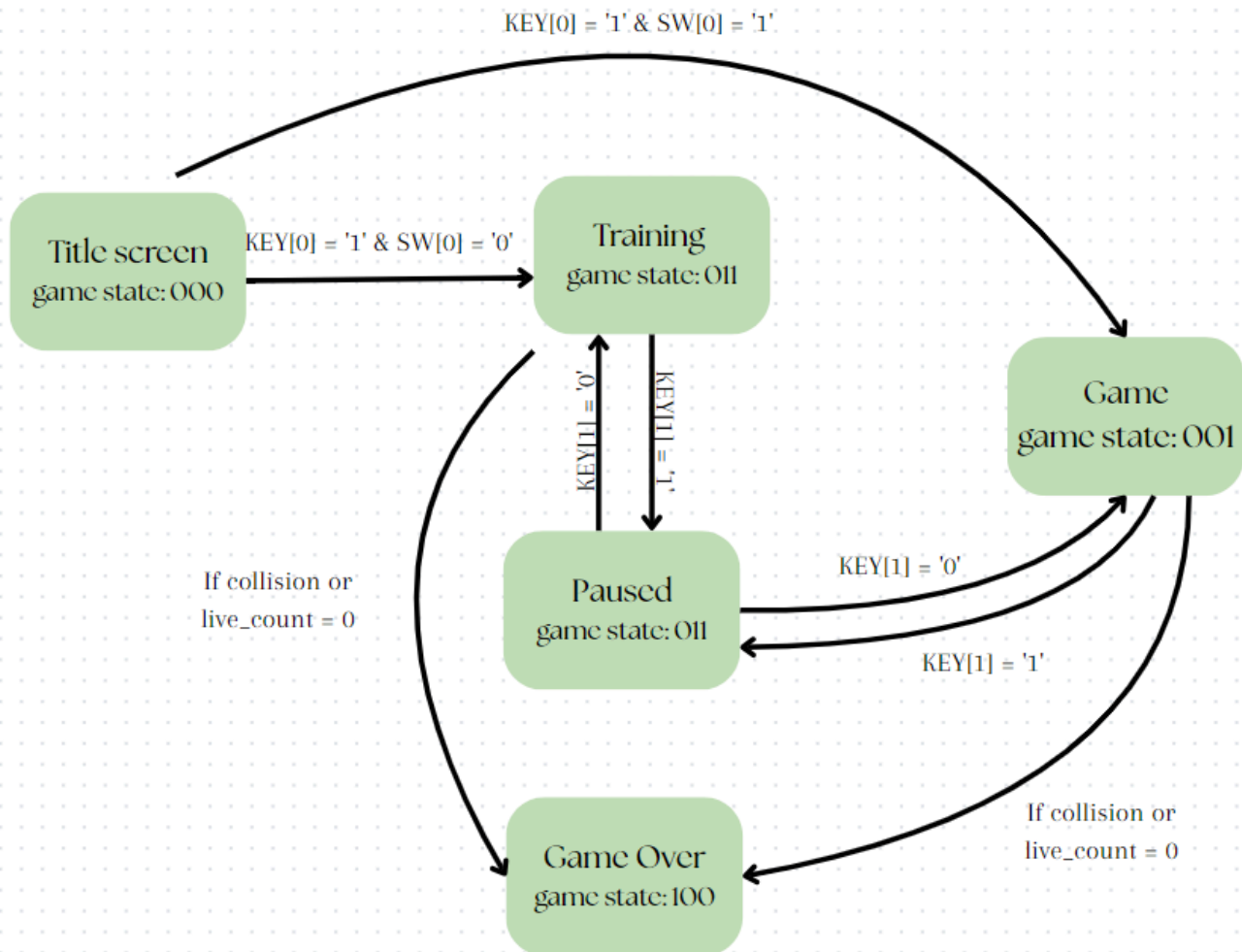
WBS	Tasks to do	Task Owner	Start Date	Due Date	Duration	Complete?
1 Interim Interview (5%)						
1.1	Brief Design Report	All	2025.05.12	2025.05.14	3	☑
1.1.1	Game strategy and design specifications, and plans	Chaeun	2025.05.12	2025.05.14	3	☑
1.1.2	A block diagram	Lucy	2025.05.12	2025.05.14	3	☑
1.1.3	The high-level state machine of the game. (all the flow, transitions, pages)	Salomia	2025.05.12	2025.05.14	3	☑
1.2	Demo	All	2025.05.09	2025.05.12	4	☑
1.2.1	Working Mouse	Lucy & Salomia	2025.05.09	2025.05.12	4	☑
1.2.2	DIP Switches & buttons	Chaeun	2025.05.09	2025.05.12	4	☑
1.2.3	VGA display(Text) and 7 segments display (mouse XY)	Lucy	2025.05.09	2025.05.12	4	☑
1.2.4	Moving ball on the VGA display & some control	Lucy	2025.05.09	2025.05.12	4	☑
1.3	Interview Preparation	All	2025.05.14	2025.05.16	3	☑
2 Final Interview (15%)						
2.1	Final Demo	All	2025.05.19	2025.05.28	10	☐
2.1.1	Pipes : Function Coding	Chaeun	2025.05.19	2025.05.22	4	☐
2.1.2	Background : Function Coding	Salomia	2025.05.19	2025.05.22	4	☐
2.1.3	Obstacles : Function Coding	Lucy	2025.05.19	2025.05.23	5	☐
2.1.4	Items : Function Coding	Lucy	2025.05.19	2025.05.23	5	☐
2.1.5	Game level - increase the speed (Need 3 levels)	Chaeun	2025.05.22	2025.05.23	2	☐
2.1.6	Game Over Requirement	Salomia	2025.05.22	2025.05.24	2	☐
2.1.7	Score & Level	Lucy	2025.05.23	2025.05.24	2	☐
2.1.8	Implement all the design components	Chaeun	2025.05.24	2025.05.26	3	☐
2.1.9	FSM implementation and connection	Salomia	2025.05.24	2025.05.26	3	☐
2.1.9	LSFR for obstacle/item generation	Lucy	2025.05.24	2025.05.26	3	☐
2.1.10	System Integration	All	2025.05.26	2025.05.28	3	☐
2.1.11	Full Gameplay Testing	All	2025.05.26	2025.05.28	3	☐
2.2	Presentation Preparation	All	2025.05.28	2025.05.29	2	☐
2.3	Interview Preparation	Individual	2025.05.19	2025.05.30	12	☐
3 Final Report (10%)						
3.1	Final System design and implementation	Chaeun	2025.05.28	2025.05.30	3	☐
3.2	Information about the FSM	Salomia	2025.05.28	2025.05.30	3	☐
3.3	Design decisions/trade-offs explanation	Lucy	2025.05.28	2025.05.30	3	☐
3.4	Resource usage and performance of the implementation	Chaeun	2025.05.28	2025.05.30	3	☐

Block Diagram:



[Link to Block Diagram](#)

The high-level state machine of the game:



[Link to High Level State Machine](#)