AI 시스템반도체 설계(2기)

1차 프로젝트 보고서

[게임 설계 프로젝트_v2]

담당강사 : 류경식

훈련기관 : 대한상공회의소 서울기술교육센터

분 반 : 502

교 육 명 : AI 시스템반도체 설계(2기)

제 출 자 : 최지우

제 출 일 : 2025. 05. 06

목 차

__(추가)시연 설명서__

1. 단계 1 : 게임 시작 >> 준비_1 >> 난이도_1

처음 게임에 진입하면, '**PLAY! / Press any key to start**'라는 문구와 함께 배경음이 재생된다. 이때 아무 키나 누르면 다음 상태인 '**준비 1**'로 전환되며, 배경음은 종료되고 화면에는 '**Stage 1 / Press any key to start**'라는 문구가 출력된다. 여기서 다시 아무 키나 누르면 게임이 시작된다.

2. 게임 방법 : 키 조작

게임 조작은 왼쪽의 '**Jog_key**'와 오른쪽의 '**SW1**'을 이용한다. 처음에 플레이어의 비행기는 화면 우측 하단에 노란색으로 표시되며, '**Jog_key'**를 조작해 상/하/좌/우로 이동할 수 있고, '**SW1**'을 누르면 공격을 발사한다.

3. 게임 규칙 : '점수'와 'HP'

LCD 화면 좌측 상단에는 '**점수**', 하단에는 '**HP**' 정보가 출력된다. 플레이어는 키를 조합해 적의 공격을 피하고 자신의 공격을 명중시켜야 하며, 적에게 공격을 10회 명중시켜 총 10점에 도달하면 다음 난이도로 진입하게 된다. 반대로 적의 공격에 3번 피격되면 HP를 모두 잃고 **'Game Over'** 상태가 된다.

4. 단계 2 : 난이도_1 >> 준비_2 >> 난이도_2 >> 준비_3 >> 난이도_3

'**준비 2**', '**준비 3**' 단계에서도 **'Stage@/Press any key to start'**라는 문구가 출력되며, 아무 키나 누르면 게임이 시작된다. 다음 난이도로 넘어가는 방식은 이전과 동일하며,

- **난이도 2** 에서는 20 점,
- **난이도 3** 에서는 30 점을 달성하면 된다.

즉, 각 단계에서는 기존 점수에 추가로 10 점을 더 쌓으면 된다.

5. 아이템 : 'HP 회복'

플레이어의 체력이 최대치(3)가 아닐 경우, 일정 확률로 맵 내 무작위 위치에 '**체력을 회복할 수 있는 오브젝트**'가 등장한다. 이 오브젝트와 충돌하면 체력이 회복되며, 이는 좌측 하단의 HP 변화 또는

획득 시 재생되는 효과음으로 확인할 수 있다. 제한 시간 내에 획득하지 못하면 해당 오브젝트는 사라지며, 이후 일정 확률로 다른 위치에 다시 등장한다.

6. 단계 3 : 난이도_3 >> GameClear

난이도 3에서 30점을 달성하면, 화면에 '**GAME Clear! / Congratulation!**'이라는 문구가 출력되며 클리어 배경음이 1회 재생된다. 이때 다시 아무 키나 누르면 게임은 처음 단계로 돌아가며, 게임 시작 배경음이 다시 재생된다.

7. 특수 상황 : GameOver

플레이어가 적의 공격에 맞아 HP가 0이 되면, 화면에 '**GAME OVER! Press any key to restart**'라는 문구가 출력되고, 게임오버 배경음이 1회 재생된다. 이후 아무 키나 누르면 처음 단계로 되돌아가며, 게임 시작 배경음이 다시 재생된다.

Ⅰ. 서론

1. Who : 누가 프로젝트를 진행하는가?

'서울상공회의소 기술교육센터'에서 운영하는 'AI 시스템반도체 설계(2기)' 교육 과정 중,『시스템 프로그래밍』및『Arm Architecture Processing』과목을 담당하시는 류경식 강사님의 지도하에, C 언어를 활용하여 STM32 보드 위에 게임을 구현하는 프로젝트를 진행한다.

이 프로젝트는 해당 교육 과정의 첫 번째 실습 과제로, 이후 진행될 팀 프로젝트와 달리 약 2주간 개인이 주도적으로 수행하는 개인 프로젝트이다. 따라서 프로젝트의 모든 과정(보고서 작성, 발표 자료 제작, 소스 코드 작성 및 결과물 구현 등)은 본인이 전담하여 수행한다.

2. When : 프로젝트의 기한은 언제까지인가?

프로젝트에 대한 기본적인 안내는 교육 시작 이후 계속해서 제공되었으나, 구체적인 채점 기준과 필요 결과물에 대한 상세 공지는 2025년 4월 25일(금요일) 오후에 전달되었다.

최종 발표는 2025년 5월 6일(화요일) 오전 11시로 예정되어 있으며, 실질적인 제작 기간은 약 10일이다. 다만 해당 기간 중 7일(주말 포함)은 교육이 휴강이므로, 강사님의 의도한 실질적인 프로젝트 수행 기간은 약 7일로 추정된다.

3. Where : 어디에서 프로젝트를 진행하는가?

STM32 보드와 이를 PC에 연결하는 케이블은 모두 강사님의 개인 소유이므로, 강의가 끝난 후에는 반드시 반납해야 한다. 보드를 집으로 가져가 작업할 수 있도록 보관용 상자가 지급되긴 하지만, 보드 파손 우려로 인해 실제로 보드에 코드를 업로드하거나 테스트하는 작업은 대부분 교육장(강의실)에서 진행한다. 다만, 프로젝트의 기간이 4월 말부터 5월 초 연휴를 포함하기 때문에, 보고서 작성이나 발표 자료 준비는 본가나 고시원 등 교육장 외부에서도 병행하여 진행된다.

4. What : 프로젝트의 주제는 무엇인가?

앞서 언급했듯이, C 언어로 STM32 보드에서 동작하는 게임을 만드는 것이 프로젝트의 기본 주제이다. 채점의 핵심 기준은 '독립적으로 이동하는 객체(Object)의 개수'이며, 그 외에도 키 입력을 통한 객체 제어, 배경 음악(BGM) 및 효과음 삽입 여부 등도 가산점 항목이다.

따라서 구현 가능성과 확장성을 고려하여, 다양한 기능을 포함할 수 있는 주제를 선정하였다. 그 결과, 사격 기능이 포함된 "Shooting Game"을 프로젝트 주제로 선택하였다.

5. Why : 왜 해당 주제를 하기로 결정했는가?

채점 기준인 객체 수와 창의성을 고려할 때, 주제의 난이도와 확장성이 중요하다. 처음에는 'Snake Game'(뱀이 이동하면서 먹이를 먹는 게임)과 'Shooting Game'(유닛을 조작하여 적을 격추하는 게임) 중에서 고민하였다. 그러나 'Snake Game'은 상대적으로 난이도가 낮아 설계가 일찍 종료될 가능성이 있었고, 반면 'Shooting Game'은 제한된 시간 안에서도 지속적인 기능 확장이 가능하다고 판단되었다.

이러한 판단을 바탕으로 최종적으로 'Shooting Game'을 프로젝트 주제로 결정하였다.

6. How : 어떻게 설계할 것인가?

게임 구현에는 기존 강의에서 배운 UART, Timer, SysTick, LCD, LED 등의 모듈을 그대로 활용하며, 추가적으로 제공된 jog_key와 graphics 관련 헤더 파일도 함께 사용된다.

게임에서 효과음을 출력하고 키 입력을 처리하기 위해 인터럽트(Interrupt)의 사용은 필수적이다. 따라서 사전에 제공된 GAME_TEMPLATE 파일 내에 포함된 인터럽트 활성/비활성 관련 코드를 분석하고 적절히 활용하는 것이 핵심적인 설계 전략이 될 것이다.

Ⅱ. 설계

1. 목표 설정

· 기본적인 비행기 슈팅 게임을 구현한다.

· 적 기체는 무작위로 이동하도록 설계하여 예측 불가능한 움직임을 유도한다.

· TIM3, TIM2는 게임의 배경음악과 효과음 재생에 사용하고, TIM4는 각각 적 기체의 움직임을 담당하는 타이머로 활용한다.

· 주어진 기간 내에 게임의 기본 틀을 성공적으로 구현한 후, 추가적으로 다양한 요소를 도입하여 완성도를 높일 계획이다. ·

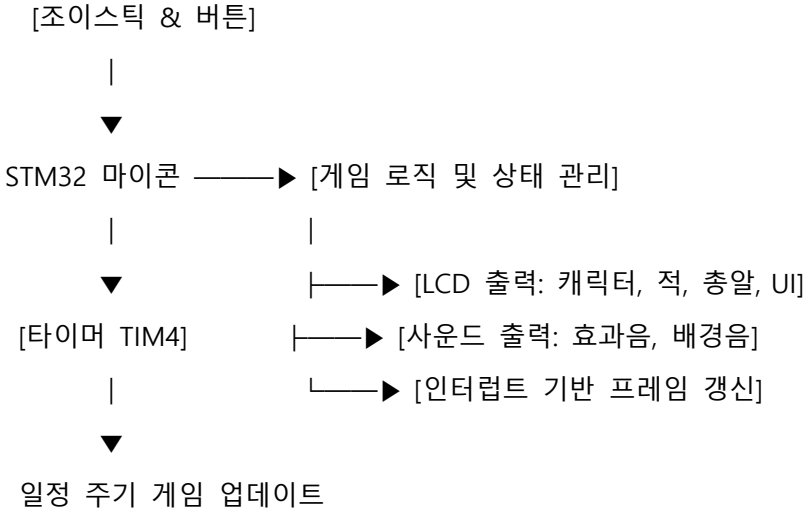
· **(추가)**아군 기체에 체력(HP) 시스템을 도입하고, 체력이 일정 이하로 감소했을 경우 무작위 위치에 회복 아이템이 생성되도록 한다. 해당 아이템을 획득하면 체력이 회복되도록 설계한다.

· **(추가)** 아이템을 획득하거나 적의 공격에 피격되는 등 상호작용이 발생할 때, 효과음이 재생되도록 구현한다.

· **(추가)** 게임 클리어 또는 게임 오버 시, 해당 상황에 맞는 배경 음악이 재생되도록 설정한다.

2. 설계 구성 요소

(1) 전체 시스템 구조도(System Overview)



(2) 하드웨어 구성(Hardware Configuration)

- **STM32 보드:** 게임 전체 로직 및 제어 수행
- **조이스틱:** 아날로그 입력으로 플레이어 이동

- **버튼:** 게임 시작 및 재시작 등 입력
- **LCD 디스플레이:** 게임 그래픽 출력 (비행기, 적, 총알, 하트, 점수 등)
- **부저(PWM 기반):** 효과음 및 배경음 출력
- **타이머 TIM4:** 주기적 인터럽트로 게임 속도 조절 및 상태 업데이트
- **타이머 TIM2 & TIM3:** 게임 내에서 발생하는 배경음 및 효과음 조절

(3) 소프트웨어 구성(Software Components)

- **메인 루프 (main())** – 게임 상태 흐름 제어 (START, PLAY1~3, GAMEOVER)
- **입력 처리 (Jog_key(), Jog_key_in())**
- **그래픽 출력 (Lcd_Draw_Box(), Lcd_Clr_Screen(), Lcd_Printf(), Lcd_Put_Pixel(), Lcd_Init())**
- **객체 관리 구조체 (Player, Enemy, Bullet, Item)**
- **충돌 판정 및 업데이트 (Check_Plane_Collision(), Check_Bullet_Collision())**
- **사운드 출력 (Buzzer_Beep(tone, duration))**
- **인터럽트 루틴 (HardFault_Handler(), EXTI3_IRQHandler(), EXTI9_5_IRQHandler(), TIM4_IRQHandler(), USART1_IRQHandler(), EXTI15_10_IRQHandler())**

(4) 게임 로직 설계(Game Logic Design)

구조 요약:

- 게임은 START → PLAY1 → PLAY2 → PLAY3 → GAMEOVER 상태를 순차적으로 진행
- 각 상태는 난이도(속도)로 구분되며, **TIM4->ARR** 값을 줄여 탄막 속도 증가
- 적군은 정해진 위치에서 나타나며, 좌우 랜덤 방향으로 이동
- 적군과 플레이어는 총알을 발사하며, 충돌 시 상태 변화(점수 증가, 생명 감소 등)
- Jog_Key()에 들어오는 값이 상/하/좌/우 로 이동시키는 입력이고, **EXTI3**이랑 **EXTI15_10**의 Interrupt를 통해서 신호를 처리.

핵심 조건:

- 적에 총알 명중 → 점수 증가
- 적 총알에 맞음 → 생명 감소
- 생명 0 → GAMEOVER
- 점수 일정 이상 → 다음 라운드로 자동 진입

(5) 그래픽 렌더링 방색(LCD Rendering)

- **Lcd_Draw_Box() :** 테두리 및 오브젝트 그리기 (사각형 모양)

- **Lcd_Clr_Screen() :** 화면 전체를 초기화하기 (검정색)
- **Lcd_Printf() :** LCD의 원하는 위치에 원하는 크기와 색깔의 글씨를 출력하기.
- **Lcd_Put_Pixel() :** 비행기의 체력을 나타내는 하트 모양을 그리기.
- **Lcd_Init(3) :** LCD의 초기값을 불러오기. (해당 함수 내에 3을 입력해야 하는 것에 주의)

(6) 입력 처리 설계(Input Handling)

조이스틱:

- **Jog_key_in()**을 통해 입력 신호를 감지.
- **Jog_key()**의 값을 통해서 입력 방향을 감지하고, UP/DOWN/LEFT/RIGHT 방향에 따라 플레이어 위치 변경.

버튼:

- **Jog_key_in()**을 통해 버튼 눌림 감지.
- START/READY 상태에서 버튼을 누르면 게임 시작.

보완 요소:

- 조이스틱 이동 범위 제한. (Object가 테두리의 영향을 침범하지 않도록 제한을 설정.)

(7) 사운드 처리(Sound Handling)

- 배경음 및 효과음은 **Buzzer_Beep()** 함수로 구현.
- 내부적으로 TIM3과 TIM2을 활용해서 톤을 생성.

(8) 인터럽트 처리 구조(Interrupt Design)

- Jog_key 신호를 처리하는 Interrupt 함수는 사전에 제공됨.
- 아래와 같은 Interrupt 함수로 처리.

```
////////////////////////////////////////////////////////////////////////////////
 volatile int Jog_key_in = 0;
 volatile int Jog_key = 0;

 void EXTI3_IRQHandler(void)
 {
   // UP
   Jog_key_in = 1;
   Jog_key = 0;
```

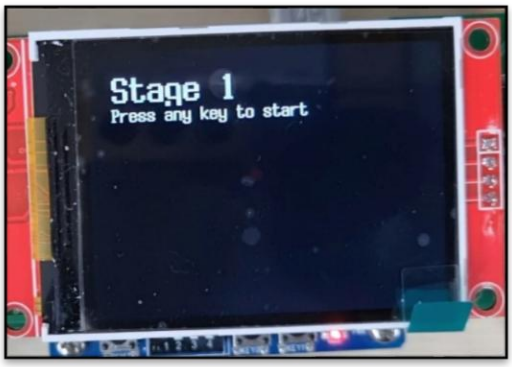
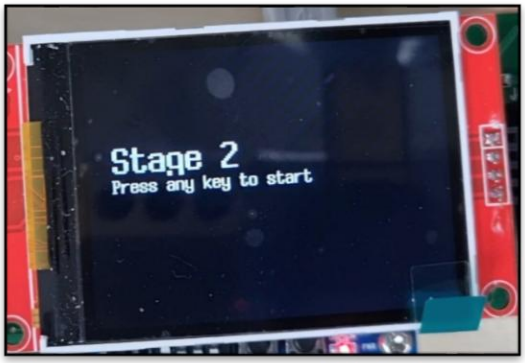
```
  EXTI->PR = 0x1<<3;
  NVIC_ClearPendingIRQ(EXTI3_IRQn);
}

void EXTI9_5_IRQHandler(void)
{
  // RLD
  static int EXTI9_5_LUT[8] = {0,1,2,0,3,0,0,0};
  Jog_key = EXTI9_5_LUT[Macro_Extract_Area(EXTI->PR,0x7,5)];
  Jog_key_in = 1;
  EXTI->PR = 0x7<<5;
  NVIC_ClearPendingIRQ(EXTI9_5_IRQn);
}

////////////////////////////////////////////////////////////////////////////////
```

Ⅲ. 일정

날짜	주요 작업 내용
4/25 (금)	- 프로젝트 및 평가 기준 소개
4/26 (토) **~ 4/27 (일)**	- 주제 선정 - 1차 계획서 작성
4/28 (월)	- 아군/적군 비행기 이동 설계 - 총알 움직임 설계 - TIM4 타이머를 활용한 움직임 구현
4/29 (화)	- 충돌 판정 함수 설계 (총알 vs 적, 총알 vs 아군 등) - 아군 비행기 세모 출력 시도 (→ 실패)
4/30 (수)	- LCD 출력 구성: 테두리, 점수판, 시작/종료 화면, 난이도 알림 등 - 시작 배경음 설계 - 충돌 시 점수 증가 구현
5/1 (목)	- 체력(HP) 시스템 구현 - 회복 아이템 설계 - 라운드 및 난이도 설계 (1~3단계) - 공격 및 아이템 효과음 추가 - 라운드 증가 시 발생하는 버벅임/멈춤 오류 해결 - 적군의 삼방향 공격 구상 (→ 실패)
5/2 (금)	- Clear 및 Game Over 시 배경음 설정 - 보고서 작성
5/3 (토)	- 보고서 및 발표자료 작성. - 소스코드 정리.
5/4(일) **~ 5/6(화)**	- 보고서 및 발표자료 작성. - 시연 영상 촬용.

Ⅳ. 결과

1. 결과물

State : Start	State : Play1
	
State : Ready1	**State : Clear**
	
State : Ready2	**State : Gameover**
	

2. 결론

기본적인 형태로 동작하는 게임을 설계하는 데에는 큰 문제가 없었지만, 구현하지 못한 아이디어가 남아 있다는 점이 아쉽게 느껴진다.

가장 아쉬운 점은 '이동하는 객채를 삼각형으로 구현'하는 부분이다. 이 아이디어를 프로젝트의 가장 초반에 시도했더라면 시간이 좀 걸리더라도 구현할 수 있었겠지만, 이미 객체의 이동과 충돌을 모두 구현한 뒤에 떠올랐기 때문에 결국 포기하게 되었다. 객체를 기존의 사각형에서 삼각형으로 바꾸려면 이동과 충돌 처리도 전면 수정해야 했기 때문이다. 이를 통해 코드의 초기 설계가 얼마나 중요한지 직접 체감하게 되었다.

또한, 설계 중에 '공격을 세 갈래(3-way)로 쏠 수 있도록 구현할 수 있을까?'에 대해 고민했고, 실제로 이를 시도했으나 여러 차례 실패하였다. 그 과정에서 설계한 코드에 빈틈과 오류가 많다는 점을 인지하였고, 제출 자료를 준비할 시간도 고려했을 때 더 이상 시도할 여유가 없다는 판단으로 결국 이 아이디어 역시 포기했다.
(당시에 떠올린 문제점으로는 '3 방향으로 동시에 쏘게 되면 총알의 개수도 3 배로 필요하다', '총알 하나하나에 충돌 처리를 따로 구현해야 한다'는 점 등이 있었다.)

또 하나 기억에 남는 문제는 '객체 이동'을 구현하던 중 발생했던 에러이다. 내 비행기와 적 비행기를 제외한 모든 LCD 화면이 노랗게 출력되는 현상이 발생했다. 당시에는 다양한 .bin 파일을 보드에 업로드하며 이미지 출력을 실험하고 있었기 때문에, 정확한 원인을 파악하기 어려웠다. 이를 해결하기 위해 이미지 출력과 관련된 코드들을 모두 점검했지만, 1 시간이 지나도록 해결할 수 없었다. 결국 LCD 소자를 보드에서 분리한 후 재조립하는 과정을 거쳐 문제를 해결할 수 있었다.

3. 고찰

프로젝트의 설계를 모두 마친 뒤, 가장 먼저 떠오른 생각은 **"아직 갈 길이 멀다"**는 것이었다.

요즘 시대에는 화려한 그래픽과 정교한 인터페이스를 갖춘, 재미있고 완성도 높은 게임들이 수없이 많다. 그런 게임들을 접하며 살아오다 보니, 내가 직접 설계하고 구현한 게임을 바라보는 순간, 현직에서 활동 중인 전문가들과의 격차를 더 깊이 실감하게 되었다. 단순히 재미있는 게임을 만드는 것만으로는 부족하다는 사실, 그리고 높은 사양의 게임의 뒤에는 정교한 설계와 많은 고민이 있다는 점을 다시금 깨달을 수 있는 계기였다.

이번 프로젝트는 단순히 게임을 구현하는 데 그치지 않고, 설계부터 구현까지의 전 과정을 스스로 책임지며 직접 부딪혀 본 값진 경험이었다. 비록 아직 부족한 점이 많고, 상업용 게임들과 비교하면 단순한 수준이지만, 이러한 차이가 오히려 나에게 더 큰 동기를 부여했다. '나도 저런 게임을 만들 수 있을까?'라는 의문에서 시작해서, '어떻게 하면 점점 더 나아질 수 있을까?'와 같은 고민을 하게 되었다.

앞으로는 이번 경험을 바탕으로, 프로그래밍 실력뿐만 아니라 구조적인 설계 능력에 대한 능력을 계속해서 쌓을 것이다. 실패와 부족함을 두려워하기보다, 그것들을 토대로 더 나은 결과물을 만들어내는 과정을 즐길 수 있는 개발자가 되고 싶다. 작은 프로젝트 하나하나가 모여 결국 나를 성장시킬 것이라는 믿음을 가지고, 앞으로도 꾸준히 배우고 도전하겠다.