
Design Specification Document

과 목	종합설계과제
담당교수	박창현 교수님
제 출 일	2022.05.13.
학 과	컴퓨터공학과 박지연(팀장)
이 름	오은아, 나유경, 전인서, 이경주



개 발 요 약(예시)

● 목 적

본 문서는 haptic을 활용한 메타버스 실감교육 플랫폼 '메타블록'의 요구사항을 기록한 DSD문서로서 앞으로 개발할 '메타블록'의 표준이 된다.

● 관련 기술

◇ Leap Motion

- CPU: intel Core i5 이상
- RAM: 2GB RAM
- OS: Window7 이상, Mac10.7 이상
- USB: 2.0 Port 이상

◇ 플렉서 센서

- 전압 범위: 0~5V
- 작동 온도: -45°C ~ + 80°C
- 저항: 25KΩ (오차: ± 30 %)
- 굽힘 저항 범위: 45K -125K Ohms

◇ 서보 모터(MG90S)

- 동작 전압: 4.8 ~ 6.0 V
- 토크 :1.8 ~ 2.2 (kg-cm)
- 회전 각도: 0~180°
- 무게: 13.4g
- 핀 기능

Red: 5V 전압 인가

Orange: PWM 제어

Brown: 접지 처리

◇ 진동 모듈

- 무게: 1g
- 전류: 80mA
- 전압: DC 3.7

◇ 우노 r3(DM70)

- 마이크로컨트롤러: ATmega 328
- 동작전압: 5V
- 입력전압(어답터등 외부전원): 7-12V
- SRAM: 2KB(ATmega328)
- EEPROM: 1KB (ATmega328)
- 클럭속도: 16MHz

● 개 발 환 경

◇ Unity3D

- 유니티 에셋스토어 Oculus Integration에셋

◇ Visual Studio 2022

- 객체 지향 프로그래밍 언어 C#

◇ Windows 11

● 제 품 요 약

메타버스와 오쿨러스 퀘스트, 햅틱 디바이스로 구성된 시스템에서 사용자가 햅틱 디바이스를 이용하여 메타버스 상에서 실제로 블록을 쌓는 듯한 촉각을 느끼며 사용자가 원하는 모양대로 블록을 쌓는다.

● 개 발 비 용

◇ 환경구축

Unity3D Part

Haptic Device Part

◇ 임 금

10,000원/시간

● 목 표 시 장

◇ 주 시장

사고력 발달 교육 플랫폼

◇ 2차 시장

메타버스를 활용한 게임시장

□ Contents

1. 서론

- 1.1 프로젝트 개요
- 1.2 블록 다이어그램

2. DSD

- 2.1 HW DSD
 - 2.1.1 사용 부품
 - 2.1.2 부품 기능
 - 2.1.3 핀의 기능
 - 2.1.4 보드 스펙 비교
 - 2.1.5 회로도
- 2.2 SW DSD
 - 2.2.1 Flow Chart Diagram
 - 2.2.2 Metaverse 함수
 - 2.2.3 Haptic Device 함수

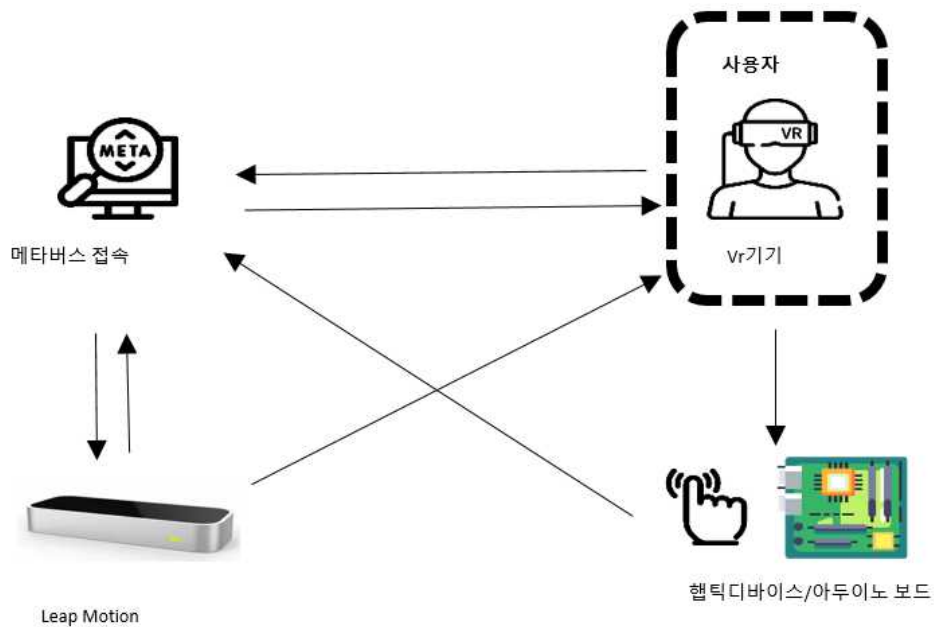
3. 결론

- 3.1 수행 방법 및 사용 도구
- 3.2 테스트 방안

1. 서론

1.1 프로젝트 개요

본 프로젝트는 메타버스, 오쿨러스 퀘스트, 햅틱 디바이스로 구성된 시스템을 개발하고자 한다. 사용자가 직접 두 손을 사용하여 가상의 블록을 쌓을 수 있는 메타버스 상 가상공간을 제공, 사용자가 햅틱 디바이스를 이용하여 메타버스 상에서 본인이 원하는 모양대로 블록을 쌓아나가 사용자만의 특별한 블록작품을 제작, 사용자가 햅틱 디바이스를 사용하여 실제로 블록을 쌓는 듯한 경험을 제공할 수 있게 하는 실감 콘텐츠 프로그램을 개발하는 것이 목표이다. 프로젝트 개념도는 그림 1과 같다.

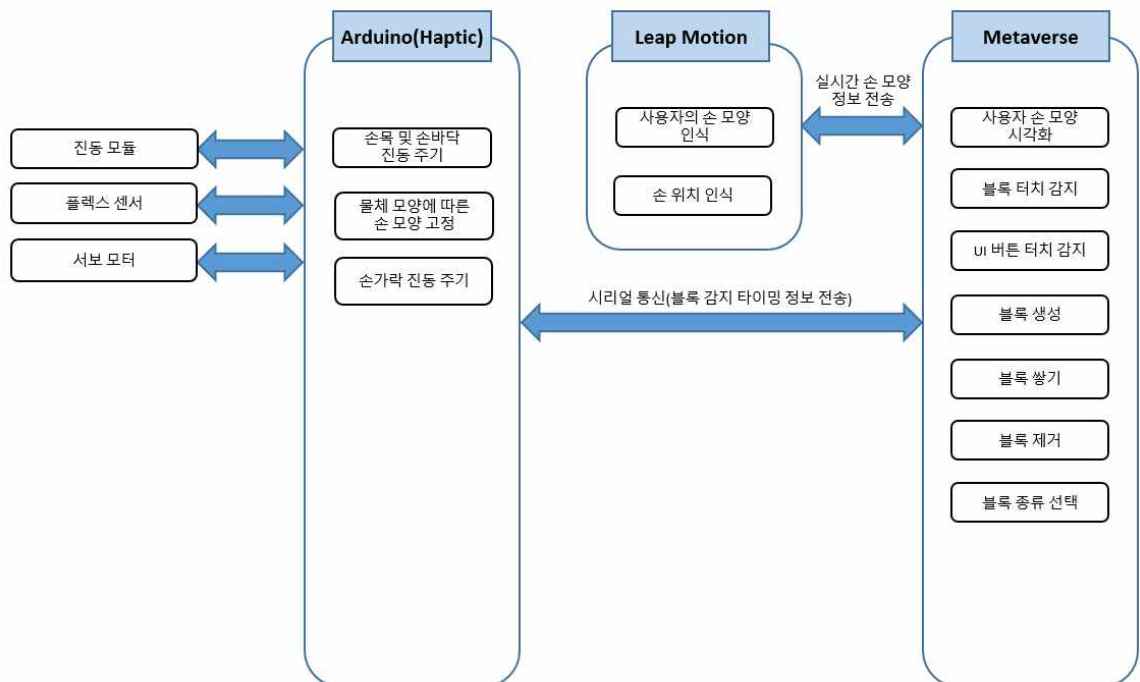


[그림 1] 프로그램 전체 흐름도

1.2 블록 다이어그램

1.2 블록다이어그램

개발에 있어 전체 시스템의 구조를 도식화한 Block diagram은 그림 2와 같다. 본 시스템은 크게 Metaverse와 Arduino(Haptic), Leap Motion으로 구분된다. 아두이노는 Unity3D와 시리얼통신(블록 감지 타이밍 정보 전송)을 한다. Leap Motion과 Metaverse는 실시간 손 모양 정보 전송을 한다. Leap Motion은 사용자의 손 모양을 인식하고 손 위치를 인식한다. Metaverse는 사용자 손 모양을 시각화 하고 블록 터치를 감지하며 UI 버튼 터치 감지를 한다. 또 블록을 생성, 쌓기, 제거를 할 수 있으며 블록 종류를 선택할 수 있다. Arduino(Haptic)은 진동 모듈, 플렉스 센서, 서보 모터를 통해 손목 및 손바닥 진동을 주고 물체 모양에 따른 손 모양을 고정하고, 손가락에 진동을 준다.



[그림 2] Block Diagram

2. DSD

2.1 HW DSD

2.1.1 사용 부품

- 진동 모듈
- 플렉서 센서
- 서보모터
- Leap Motion
- 오쿨러스 퀘스트

2.1.2 부품 기능

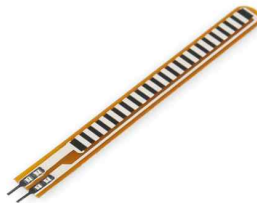
- 진동 모듈



[그림 3] 진동 모듈

손목, 손바닥, 손가락에 진동을 주는 모듈이다.
특징으로는 무게는 1g, 전류는 80mA, 전압은 DC 3.7이다.

- 플렉서 센서



[그림 4] 플렉서 센서

손가락이 기울어진 정도를 수치로 계산해주는 모듈이다.
특징으로 전압 범위는 0~5V, 작동 온도는 -45°C ~ + 80°C, 저항은 25KΩ (오차: ± 30 %), 굽힘 저항 범위는 45K -125K Ohms 이다.

- 서보모터



[그림 5] 서보 모터

물체 모양에 따른 손 모양을 고정하기 위한 모듈이다. 특징으로는 동작 전압: 4.8 ~ 6.0 V, 토크 :1.8 ~ 2.2 (kg-cm), 회전 각도: 0~180°, 무게: 13.4g, 핀 기능으로는 Red: 5V 전압 인가, Orange: PWM 제어, Brown: 접지 처리를 한다.

- Leap Motion



[그림 6] Leap Motion

사용자의 손 모양을 인식하고 손 위치를 인식하기 위한 것이다. 특징으로는 CPU: intel Core i5이상, RAM: 2GB RAM, OS: Window7이상, Mac 10.7이상, USB: 2.0 Port 이상을 요구한다.

- 오쿨러스 퀘스트

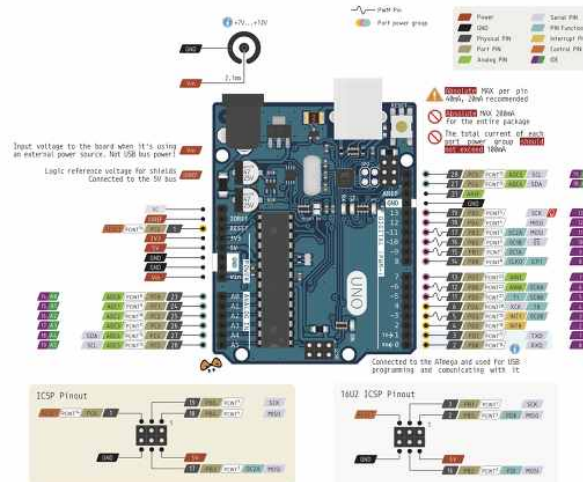


[그림 7] 오쿨러스 퀘스트

사용자가 Metaverse를 시각적으로 좀 더 실감나게 사용할 수 있도록 해주는 것이다. 특징으로는 USB3.0 케이블로 컴퓨터와 연결해 PC 기반의 VR 기반 콘텐츠를 사용할 수 있게 한다. 작동 방식은 PC에서 렌더링 된 VR 콘텐츠를 인코딩 시킨 후 전송하여 Quest에서 디코딩 한 후 화면에서 보여주는 방식이다. 권장 사양은 CPU: Intel i5-4590 / AMD Ryzen 5 1500X 이상, 그래픽 카드는 NVIDIA GeForce GTX 970이상, AMD Radeon RX 400 Series 이상, RAM은 8GB이상, 운영체제는 Windows10 이상, USB는 3.0 이상을 요구한다.

2.1.3 핀의 기능

핀 및 기능 설명은 그림 9와 같다.



[그림 8] 아두이노 보드

핀 번호	역 할
3V3	보드로부터 3.3V의 전압을 출력할 수 있는 핀. 센서 중 3.3V를 사용하는 센서의 전원을 이 핀에서 끌어낼 수 있다.
5V	보드로부터 5V의 전압을 출력할 수 있는 핀이다. 마찬가지로 센서 중 5V를 사용하는 센서의 전원을 이 핀에서 끌어낼 수 있다.
Vin	아두이노에 전원을 공급할 때 사용할 수 있는 핀이다. 기본적으로 아두이노 IDE를 사용하여 보드에 업로드할 때는 컴퓨터에 연결된 데이터 케이블로부터 전원을 공급받지만, 업로드 이후에 보드를 단독으로 사용할 시에는 별도의 외부 전원이 필요하다.
GND	보드의 접지 핀이다. 모든 회로는 하나의 접지선에 맞물려야 제대로 구동된다.
A0~A5	아날로그 신호를 입,출력할 수 있는 핀이다.
D0~D13	디지털 신호를 입,출력할 수 있는 핀이다.
D0	RX 핀의 기능을 가지고 있다. 'Receive 핀'으로, 통신 수행 시 수신 핀으로 사용될 수 있는 핀이다.
D1	TX핀의 기능을 가지고 있다. 'Transmit 핀'으로, 통신 수행 시 발신 핀으로 사용될 수 있는 핀이다.
D2/D3	인터럽트 기능을 수행할 수 있는 핀이다. 각각 INTO 와 INT1 으로 정의되어 있다.
D3/D5/D6/D9/D10/D11	PWM 신호를 출력할 수 있는 핀이다. 보드 상에서는 핀 번호 앞에 '~' 무늬가 붙어 있는 것으로 구별이 가능하다.

[그림 9] 핀의 기능

2.1.4 보드 스펙 비교

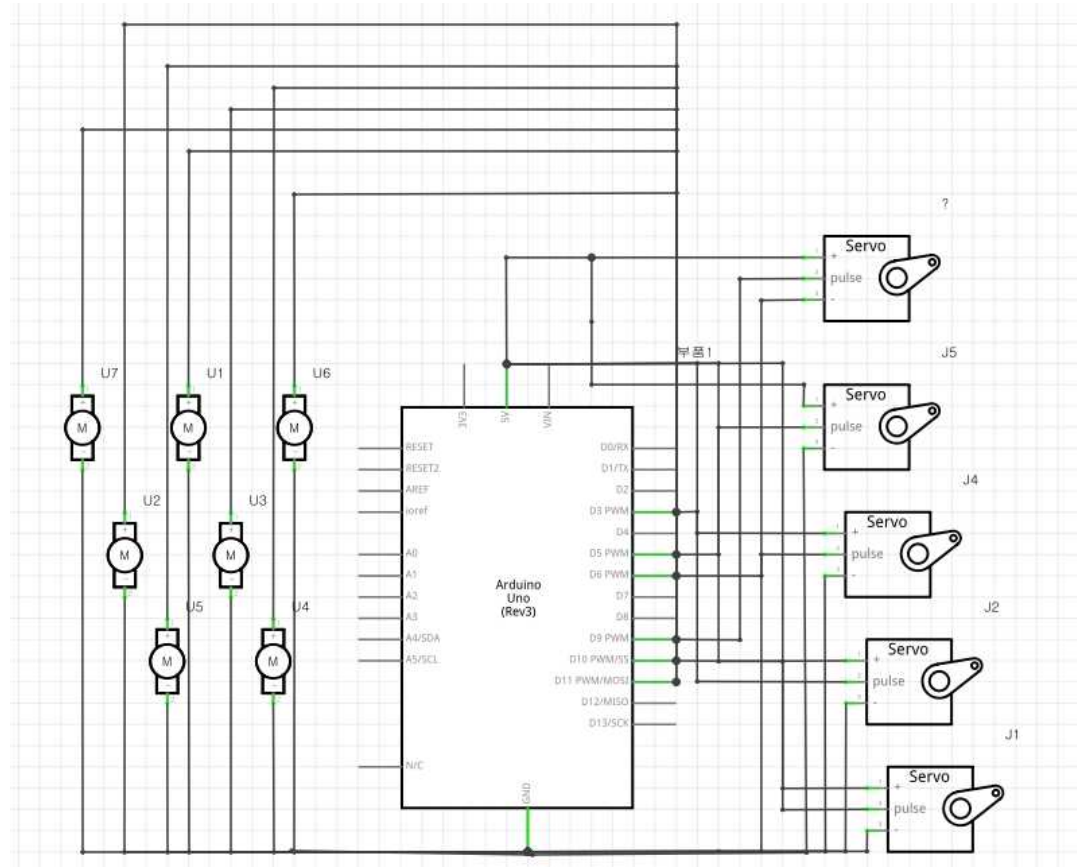
1. 개발 난이도가 쉬워야 하며 다양한 기능을 동시 구현할 수 있어야 한다.
2. 설치 형태의 제품 개발을 위해 작고 가벼워 이동에 용이해야 하며 가격은 다량 구매시 2~300,000원 이하여야 한다.
3. Unity 3D와의 연결에 용이해야 한다.
4. 사용하는 센서의 원활한 작동을 위해 5V 이상의 전압 사용이 가능해야 한다.

위의 기준을 두고 시중에 판매하는 각 임베디드 보드 모델들을 조사하였다. 아래 표는 조사한 모델들의 스펙이다. [Raspberry Pi]는 아두이노 모델보다는 훨씬 비싸지만 확장성과 성능이 아두이노에 비해 훨씬 뛰어나다는 강점이 있다. 하지만 본 프로젝트에 진행하기에는 난이도가 있고 아두이노로도 충분하다. 그리고 Unity 3D에는 아두이노 에셋이 있어 [Raspberry Pi]보다 아두이노를 사용하여 통신하는 것이 훨씬 유리하다. 따라서 해당 프로젝트에는 아두이노를 사용하는 것으로 결정하였다.

	Arduino Uno	Arduino Nano	Arduino Leonardo	Arduino Yun	Raspberry Pi 3
Price(원)	7,200	6,900	30,000	87,000	46,200
Micro Controller	ATmega 328P	ATmega 328	ATmega 32u4	ATmega 32u4	ARM Cortex-A53
Operating Voltage	5V	5 V	5V	5V	
Input Voltage	7-12V	7-12 V	7-12V	7-12V	
Digital I/O Pins	14개	6-20개	20개	20개	40개
PWM Channels	6개	14개	7개	7개	
Analogue Input Channel	6개	8개	12개	12개	
DC Current per I/O Pin	20 mA	40 mA	40 mA	40 mA	
Flash Memory	32 KB	16 KB	32 KB	32 KB	최대 256GB
Clock Speed	16 MHz	16 MHz	16 MHz	16 MHz	1.2 GHz
Length	68.6 mm	45 mm	69.6mm	73mm	85mm
Width	53.4 mm	18 mm	53.3mm	53mm	49mm
Weight	25g	5g	20g	32g	41.2g
Wifi	X	X	X	IEEE 802.11.b/g/n	X

[그림 10] 보드 스펙 비교

2.1.5 회로도

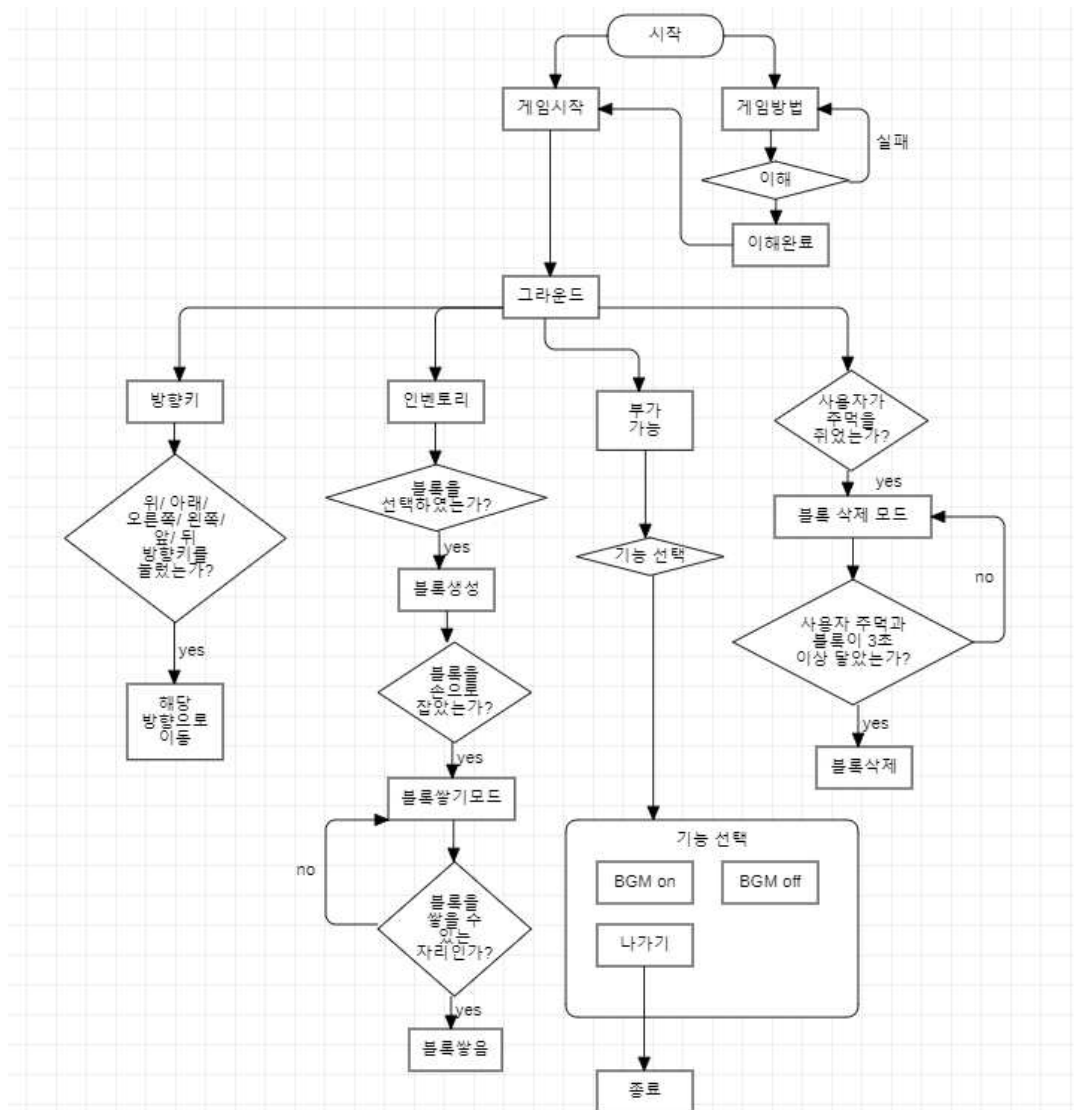


[그림 11] 회로도

2.2 SW DSD

2.2.1 Flow Chart Diagram

Flow Chart Diagram은 문제나 작업의 범위를 결정하고 분석하며, 그 해석 방법을 명확히 하기 위해 필요한 작업과 처리의 순서를 통일된 기호와 도형을 사용하여 도식적으로 표시한 것이다. 다음 [그림 12]는 '메타 블록'의 작업과 처리하는 과정을 순서대로 간단한 기호와 도형으로 도식화하여 Flow Chart Diagram으로 나타낸 것이다.



[그림 12] Flow Chart Diagram

프로그램을 시작하면 게임 시작과 게임 방법으로 나뉜다. 게임 방법에서 사용자가 이해를 완료하면 게임 시작 모드로 전환된다. 게임을 시작하면 그라운드 배경이 나타난다. 사용자는 그라운드에서 인벤토리와 방향키, 부가기능 UI를 확인할 수 있다. 메타버스 상 방향키를 햅틱 장갑을 낀 손을 이용해 터치하면 사용자의 위치를 해당 방향으로 이동할 수 있다. 메타버스 상 인벤토리에서 사용자가 쌓고싶은 블록을 선택하면 해당 블록이 사용자 시선 가운데에 생성된다. 블록을 사용자의 손을 잡으면 블록쌓기 모드로 전환된다. 메타버스 상에서 사용자 손에 쥐어진 블록의 위치가 쌓을 수 있는 자리라면 블록을 놓아 그라운드에 쌓는다. 만약 사용자가 주먹을 쥐었다면 블록 삭제 모드로 전환된다. 그리고 사용자의 주먹과 블록이 3초 이상 닿아져 있으면 해당 블록이 삭제된다. 부가기능 UI를 손가락으로 터치하면 기능 선택 창이 나타나며 BGM ON OFF 기능을 수행할 수 있다. 나가기를 터치하면 게임을 종료한다.

2.2.2 Metaverse 함수

2.2.2.1 select_arrow()

Name	select_arrow()
Input	없음
Process	사용자가 방향키를 눌렀을 때 해당 방향으로 시점을 이동하는 함수이다. 위/아래/오른쪽/왼쪽/앞/뒤 중에서 하나의 방향키를 골라서 누를 수 있다.
Output	1, 2, 3, 4, 5, 6

2.2.2.2 select_block()

Name	select_block()
Input	없음
Process	사용자가 블록의 종류를 선택할 수 있는 함수이다. 인벤토리에는 각자 색깔이 다른 9가지 종류의 블록이 존재한다. 사용자가 인벤토리에서 원하는 블록을 선택했을 때 블록쌓기 모드가 시작된다. 출력값으로는 1(블록쌓기모드 ON)이 있다.
Output	1

2.2.2.3 build_block()

Name	build_block()
Input	1(select_block())의 결과값
Process	사용자가 블록을 쌓는 함수이다. select_block()의 결과값이 1인 경우 시작된다. 사용자가 블록을 쌓길 원하는 자리를 선택하면 그 자리가

	블록을 쌓을 수 있는 자리인지 판단한다. 가능한 경우 블록을 쌓는다. 만약 블록을 쌓을 수 없는 자리면 블록이 쌓이지 않는다.
Output	성공적으로 블록을 쌓으면 success 값인 1 을 반환한다.

2.2.2.4 delete_block()

Name	delete_block()
Input	blockName
Process	사용자가 블록을 삭제하는 함수이다. 사용자와 3초이상 접촉하여 선택된 블록의 이름명을 입력받는다. 해당 블록객체를 find하여 객체를 destroy한다. 성공적으로 삭제를 완료하면 success값인 1을 반환한다.
Output	1, 0

2.2.2.5 show_preview()

Name	show_preview()
Input	없음
Process	사용자가 쌓을 블록의 위치를 큐브의 프레임으로 미리 보여주는 함수이다. 사용자가 쌓을 블록의 위치를 미리 보고 정확한 판단을 한 시점에 build_block()을 호출한다.
Output	없음

2.2.2.6 communicateUA()

Name	communicateUA()
Input	립모션 제스처, -1, 0, 1(get_gesture())함수의 Output)
Process	유니티와 아두이노 사이의 시리얼 통신을 담당하는 함수이다. 아두이노에서 플렉스센서의 출력값을 지속적으로 유니티로 보낸다. 유니티는 립모션과 플렉스센서의 출력값을 이용하여 특정 제스처를 인식한다. 인식한 제스처를 바탕으로 아두이노에 신호를 보내 서보모터 혹은 전동모터를 작동시킨다.
Output	서보모터실행(0, 1), 전동모터 실행(0, 1, 2, 3, 4, 5, 6, 7)

2.2.3 Haptic Device 함수

2.2.3.1 control_servo()

Name	control_servo()
Input	0, 1
Process	손가락 5개에 대한 서보모터를 제어할 수 있는 함수이다. 입력값이 0일 경우 블록을 손으로 잡았을 때의 신호이며 서보모터를 180도로 회전하게 하고, 입력값이 1일 경우 블록을 잡지 않았을 때의 신호이며 서보모터를 원래의 위치로 회전하게 한다.
Output	없음

2.2.3.2 control_vibrate()

Name	control_vibrate()
Input	1, 2, 3, 4, 5, 6, 7
Process	다섯 손가락과 손바닥, 손목의 진동 모듈을 제어할 수 있는 함수이다. 입력값에 따라 해당 진동 모듈을 동작시킨다. 사용자가 제어하는 상황에 따라 진동의 세기와 작동 시간을 다르게 동작시킨다.
Output	없음

2.2.3.3 get_gesture()

Name	get_gesture()
Input	flexVal
Process	flex 센서로부터 보내오는 값을 입력받아 손의 제스처를 주먹을 쥔 상태, 버튼을 클릭하기 위해 검지를 뺀 상태를 인식하는 함수이다. 출력값으로는 -1(제스처가 없는 상태), 0(주먹을 쥔 상태), 1(버튼을 클릭하기 위해 검지를 뺀 상태)가 있다.
Output	-1, 0, 1

3. 결론

3.1 수행 계획

5월 이후의 수행 계획을 다음 [그림 13] Gantt Chart로 나타내었다. 계획은 크게 설계, 개발, 팀 미팅 계획으로 나누었고 5월부터 11월까지 나타냈다. 설계로는 시작화면, 게임화면, UI 설계, 그라운드(게임 배경) UI 상세 설계, 블록 생성 및 쌓기 알고리즘 설계, 블록 삭제 알고리즘 설계, 햅틱 디바이스 부품 회로도 설계, 햅틱 디바이스 내부 알고리즘 설계로 나누었다. 개발은 UI 개발, 블록 시스템 개발, 햅틱 디바이스 제작, 유니티와 leap motion 연동, 유니티와 vr 기기 연동, 유니티와 햅틱 디바이스 연동으로 나누었다. 팀 미팅 계획은 매주 월요일 12시 온라인 미팅으로 팀원 전체가 참여하게 되고, 매주 수, 금 12시 오프라인 미팅은 팀장 박지연, 팀원 오은아가 진행 하고 매주 화, 목 12시 오프라인 미팅으로 팀원 나유경, 전인서, 이경주가 진행한다.

구분	추진내용	추진일정									
		5월	6월	7월	8월	9월	10월	11월			
설계	시작화면, 게임화면 UI 설계	■									
	그라운드(게임 배경) UI 상세 설계	■	■								
	블록 디자인 및 옵션 설계		■								
	블록 생성 및 쌓기 알고리즘 설계	■	■								
	블록 삭제 알고리즘 설계			■							
	햅틱 디바이스 부품 회로도 설계			■							
	햅틱 디바이스 내부 알고리즘 설계				■						
개발	UI 개발				■	■					
	블록 시스템 개발				■	■	■				
	햅틱 디바이스 제작				■	■	■	■			
	유니티와 leap motion 연동				■	■					
	유니티와 vr기기 연동					■	■				
	유니티와 햅틱 디바이스 연동							■	■	■	
팀 미팅 계획	매주 월요일 12시 온라인 미팅(팀 전체), 매주 수, 금 12시 오프라인 미팅(은아, 지연), 매주 화, 목 12시 오프라인 미팅(유경, 인서, 경주)	■	■	■	■	■	■	■	■	■	■

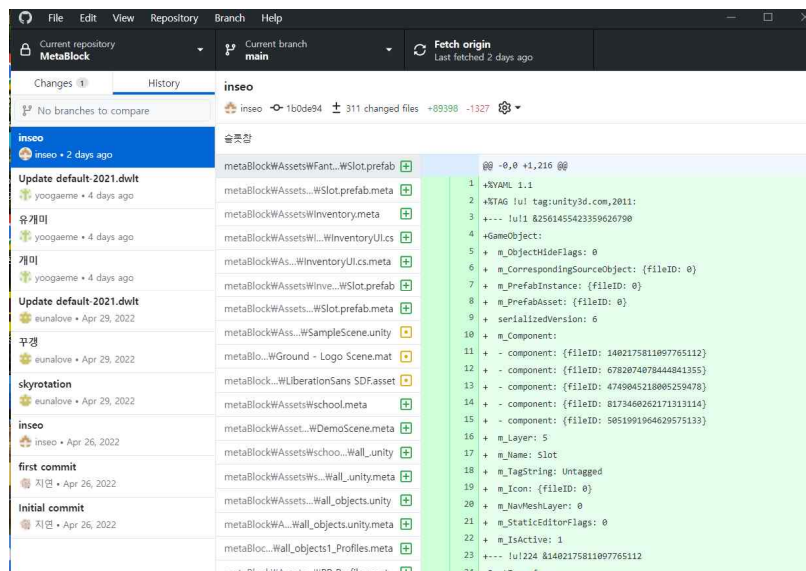
[그림 13] Gantt Chart

3.2 수행 방법 및 사용 도구

수행 방법은 Notion([그림 14] Notion 화면 참조)을 사용하여 팀 전체 일정을 조율하고 Github([그림 15] Github 화면 참조)를 통해 Unity3D 개발을 공유한다.



[그림 14] Notion 화면



[그림 15] Github 화면

3.3 테스트 방안

테스트 방안을 [그림 16] 테스트 Gantt Chart로 나타내었다. 테스트 할 항목들에는 블록 생성, 쌓기, 삭제 테스트, 슬롯 기능 동작 테스트, Leap Motion 동작 테스트, vr 기기 연동 테스트, 햅틱 디바이스 센서 및 모듈 동작 테스트, 햅틱 디바이스 피드백 테스트, Unity 부가 기능 동작 테스트, 전체 기능 최종 테스트가 있다. 전체적으로 설계 및 개발이 끝난 후 즉각 테스트를 할 계획이다. 11월에는 종합설계 프로젝트 완료 및 발표를 한다.

구분	추진내용	추진일정											
		5월	6월	7월	8월	9월	10월	11월					
테스트	블록 생성 테스트												
	블록 쌓기 테스트												
	블록 삭제 테스트												
	슬롯 기능 동작 테스트												
	Leap Motion 동작 테스트												
	vr기기 연동 테스트												
	햅틱 디바이스 연동 테스트												
	햅틱 디바이스 센서 및 모듈 동작 테스트												
	햅틱 디바이스 피드백 테스트												
	Unity 부가 기능 동작 테스트												
	전체 기능 최종 테스트												
종료	종합설계 프로젝트 완료 및 발표												

[그림 16] 테스트 Gantt Chart