



**컴퓨터 비전 시를 활용한
손 재활/발달 플랫폼**

CONTENTS

- 01 개발동기
- 02 프로젝트 소개
- 03 시연영상
- 04 사용된 기술
- 05 사업성 및 계획

01

손 발달 - 도구 활용을 통한 소근육 발달

01 개발동기

02 프로젝트 소개

03 시연영상

04 사용된 기술

05 사업성 및 계획

스마트폰 가지고 논 아이, 연필 못 칠 수도 있어요

휴대폰-태블릿이 전통 장난감 대체한 기술 세대 영 소아과 전문의들 "손가락 근육 발달 저하" 우려

기자 전정윤



전통적인 장난감 대신 스마트폰과 태블릿을 가지고 노는 요즘 어린이들이 발달 저하로 연필을 제때 제대로 잡지 못한다는 경고가 나왔다.

[이호분의 아이들 세상] 소근육 발달이 늦어 글씨 못 쓰는 아이

입력 : 2017-11-21 10:46

👍 좋아요 0개



이호분 연세누리정신과 원장

활동적이고 산만하여 움직임이 많고, 운동을 좋아하는 아이들 중 예외로 미세 소근육의 기능이 부족해 손사용이 어려운 아이가 많다. 또 대근육의 움직임도 어색해 걸음걸이가 자연스럽지 않은 아이들도 또한 많다.

P는 초등학교 3학년 남자 아이다. 어려서부터 걸기 시작하면서 바로 뛰어 다닐 만큼 활동적인 아이였다. 학교에 가서도 공부 시간에 딴 짓을 하거나 장난을 쳐 지적받는 일이 많았다. 심지어 선생님에게 대들기도 하고, 학교 규칙을 어기고, 화가 나면 참지 못해 친구들과 다툼도 잦았다. 겉디다 못한 P의 부모는 아이를 병원에 데리고 왔다.

안동포 마을 치매율 전국 1/5수준...비밀은 '수작업'

입력 | 2017-07-0606:51 수정 | 2017-07-06 07:45



정치 국정위, 내년부터 초등생 독감 예방 무료 접종

◀ 앵커 ▶

손을 많이 움직이는 수작업이 치매 예방에도 도움이 된다고 하죠.

실제로 안동포를 손수 만드는 마을의 노인 치매 유병률이 전국 평균의 1/5 수준인 것으로 확인됐습니다.



게임 기반
소근육 발달 운동

01

손 재활 - 재활 보조장비 착용

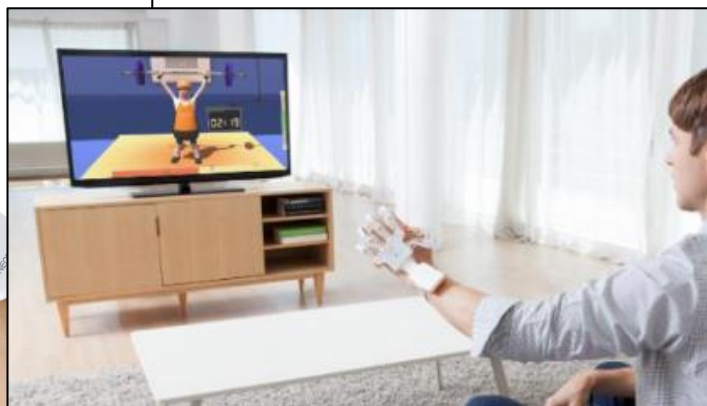
01 개발동기

02 프로젝트 소개

03 시연영상

04 사용된 기술

05 사업성 및 계획



재활장비 기반



보급형

편의성



영상인식 게임 기반
손 관절 재활 운동

02

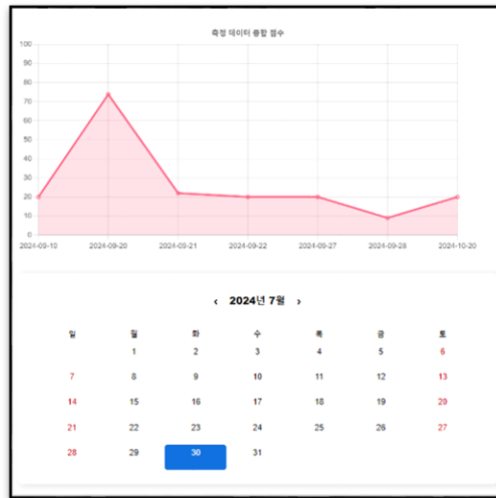
게임 기반 손동작 영상 인식을 통한 손 발달 및 재활 솔루션

- 01. 개발동기
- 02. 프로젝트 소개
- 03. 시연영상
- 04. 사용된 기술
- 05. 사업성 및 계획

유형별 손 발달 Game



손 발달 분석 Dashboard



02

화면샷 1: 유형별 손 발달 Game

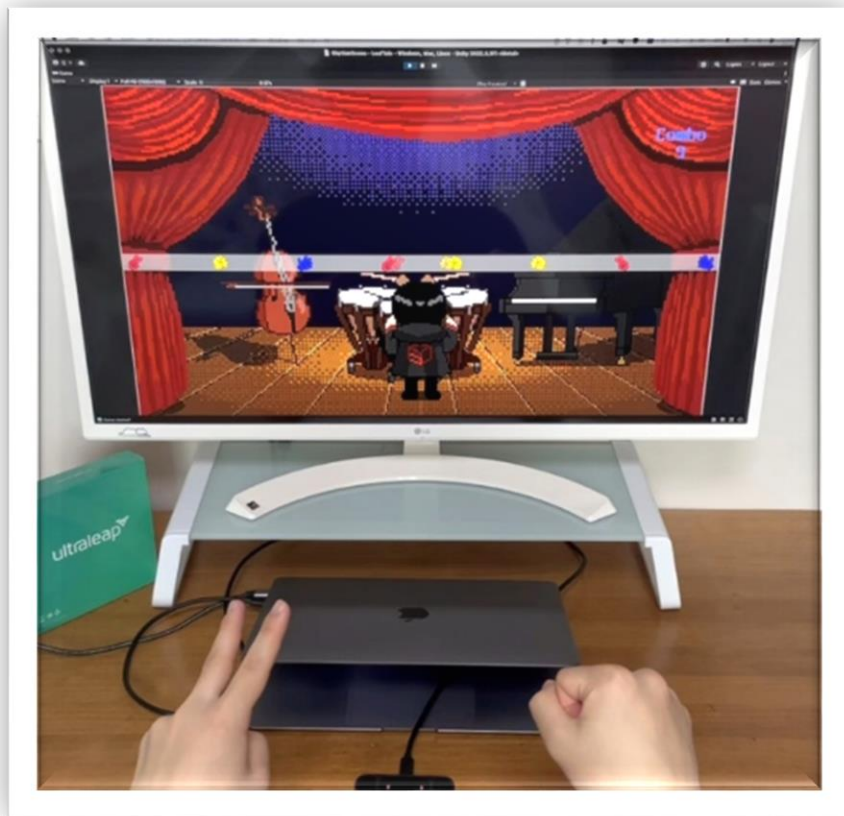
01. 개발동기

02. 프로젝트 소개

03. 시연영상

04. 사용된 기술

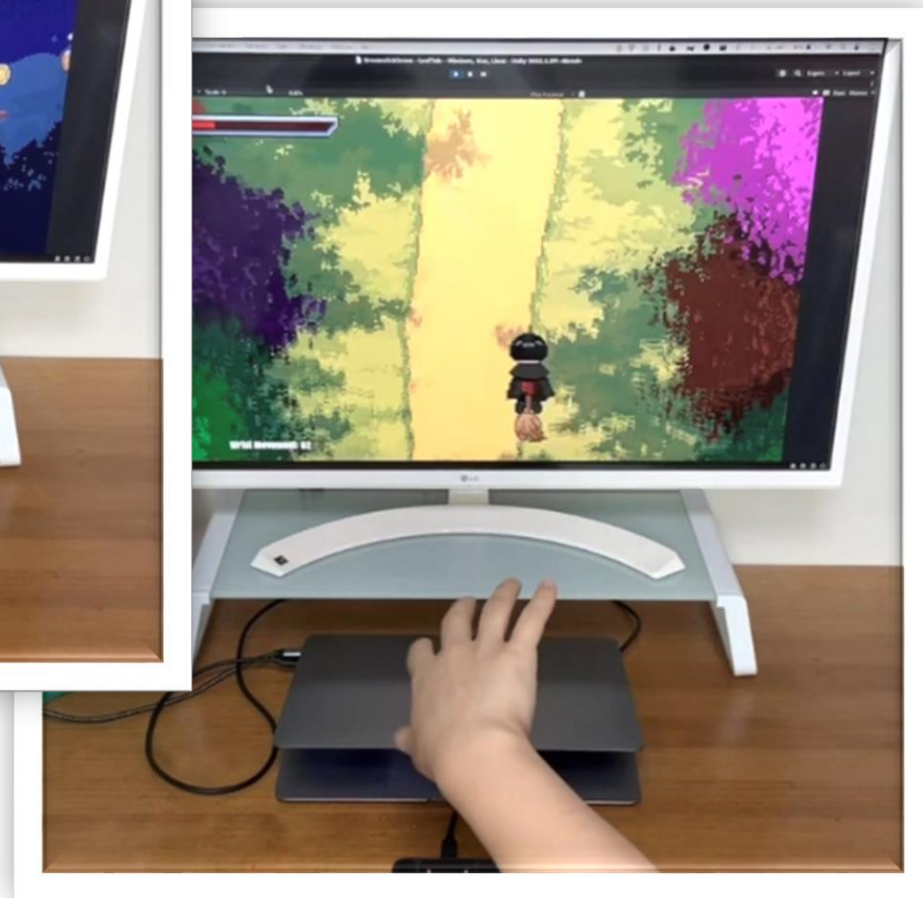
05. 사업성 및 계획



Game Type 1 :
손가락 관절 발달



Game Type 2 :
손 민첩성 발달



Game Type 3 :
손목 가동성 발달

02

화면샷 2: 손 발달 분석 Dash Board

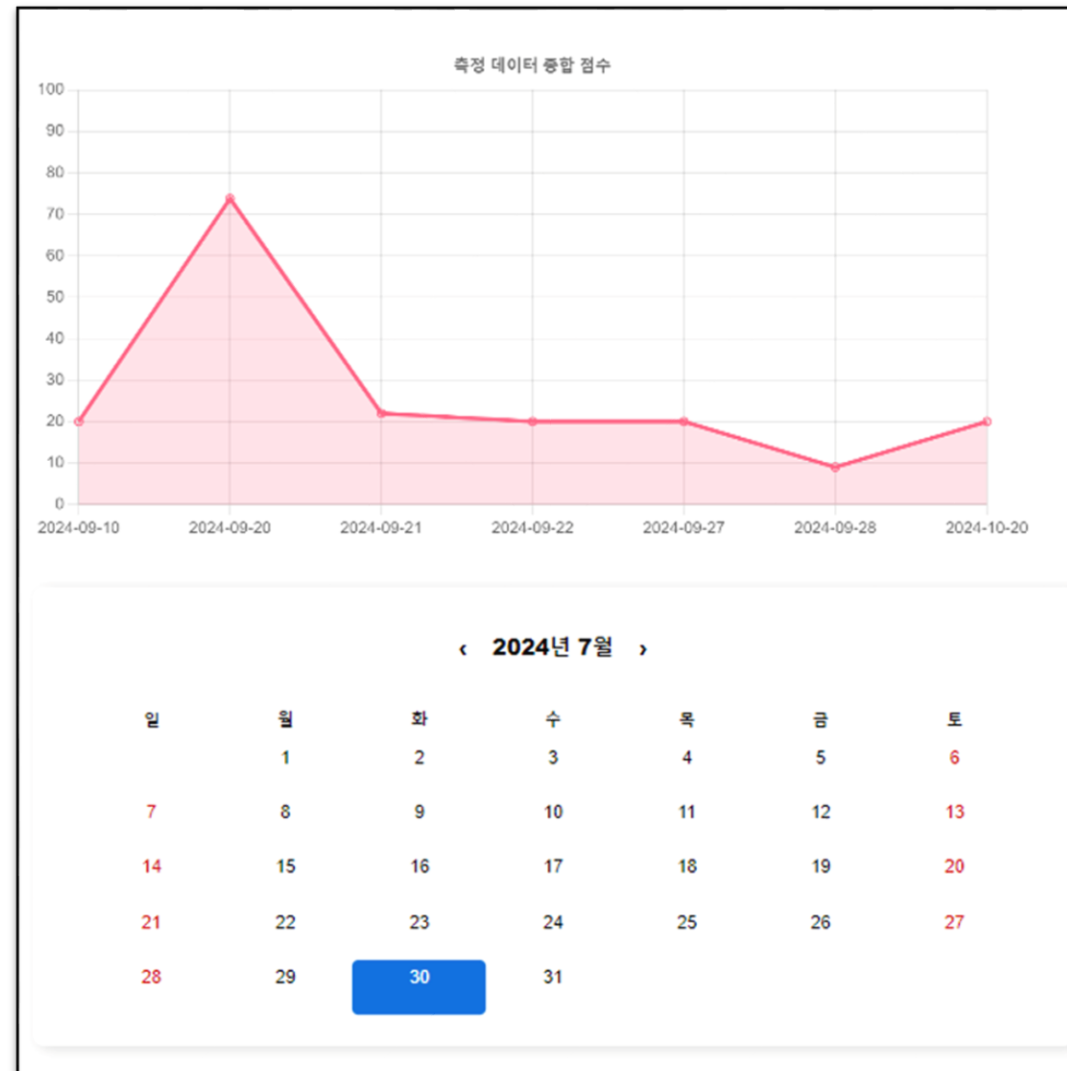
01.개발동기

02.프로젝트 소개

03.시연영상

04.사용된 기술

05.사업성 및 계획



02

화면샷 3: 핸드모션인식

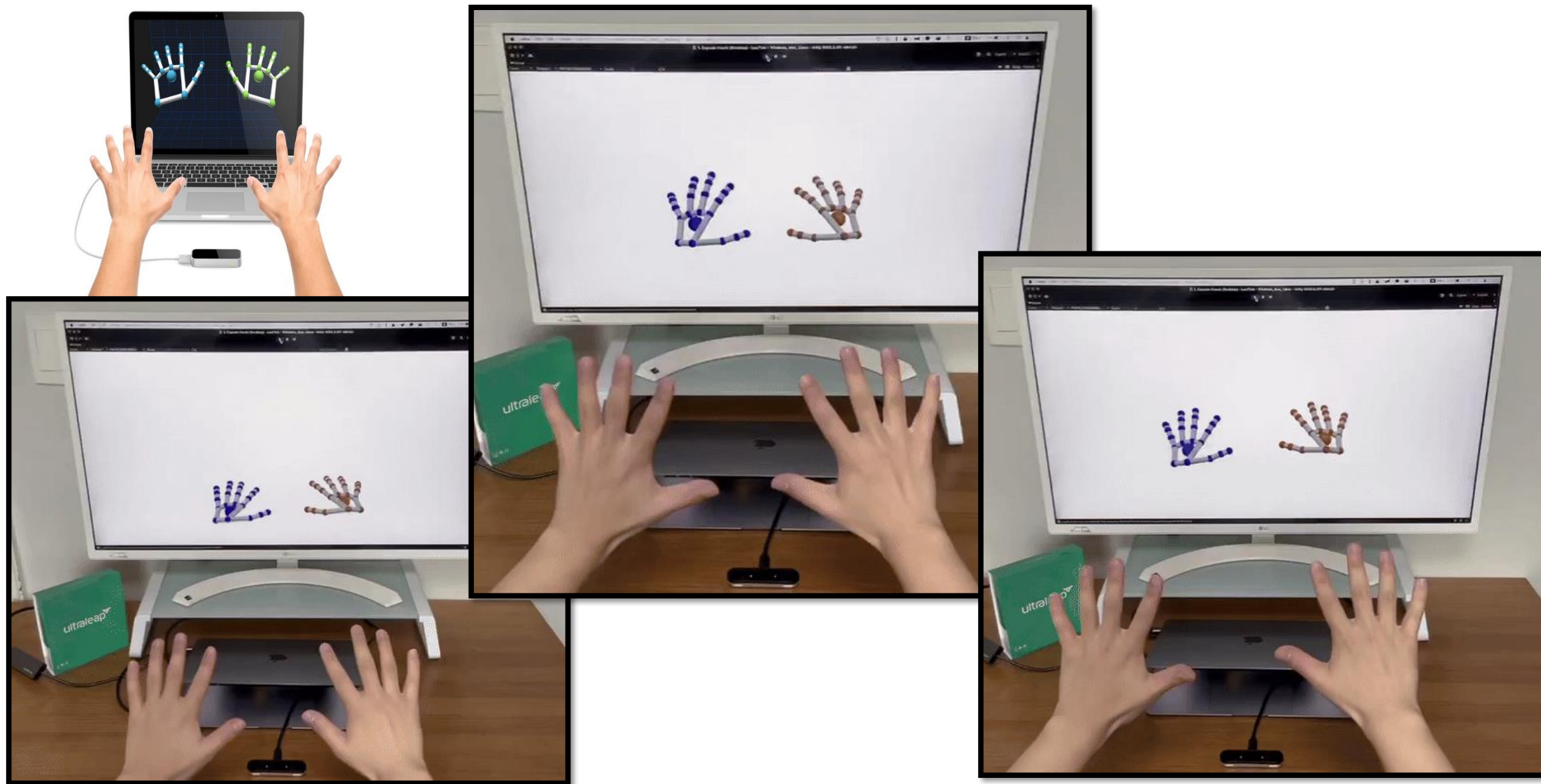
01. 개발동기

02. 프로젝트 소개

03. 시연영상

04. 사용된 기술

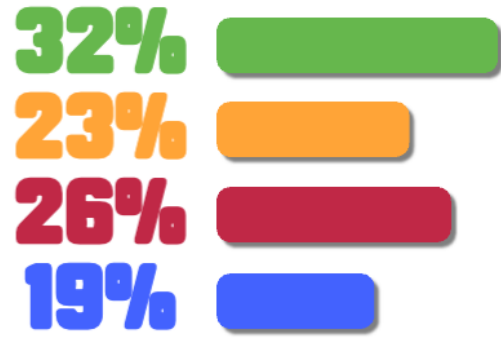
05. 사업성 및 계획



03

시연영상

[영상 보러가기](#)



프론트엔드 영역

서버에서 전송된 게임 데이터를 처리하며, 웹 대시보드를 통해 시각적으로 제공하여 사용자가 자신의 진행상황을 확인할 수 있습니다.



게임 영역

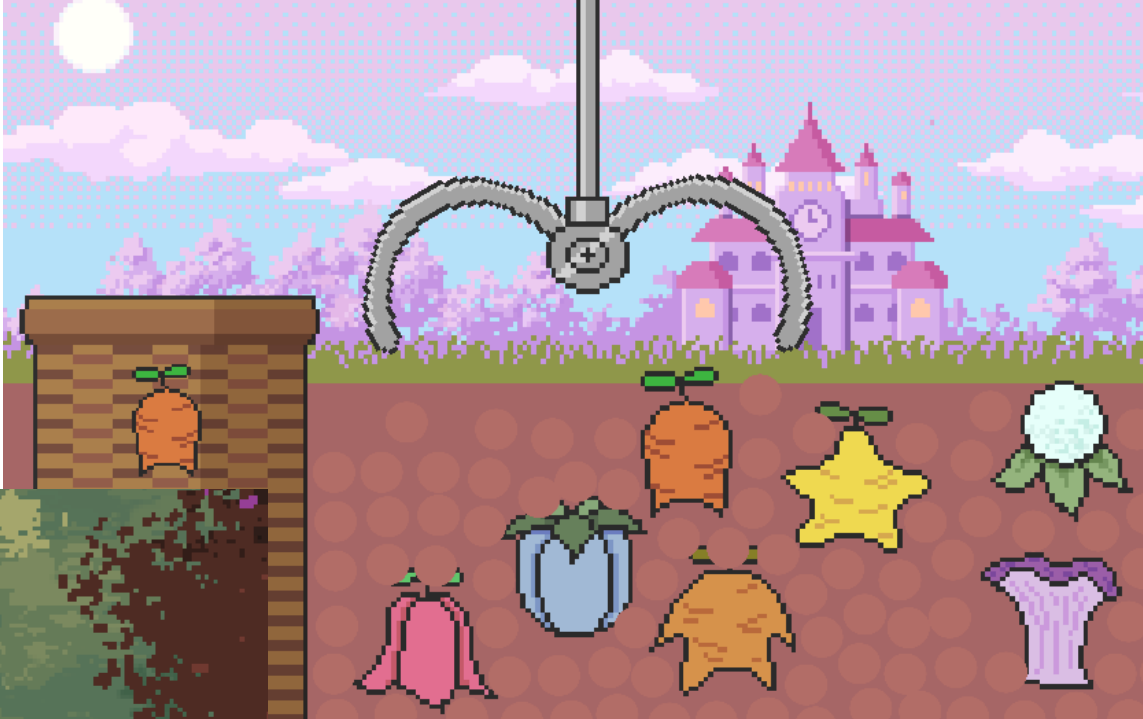
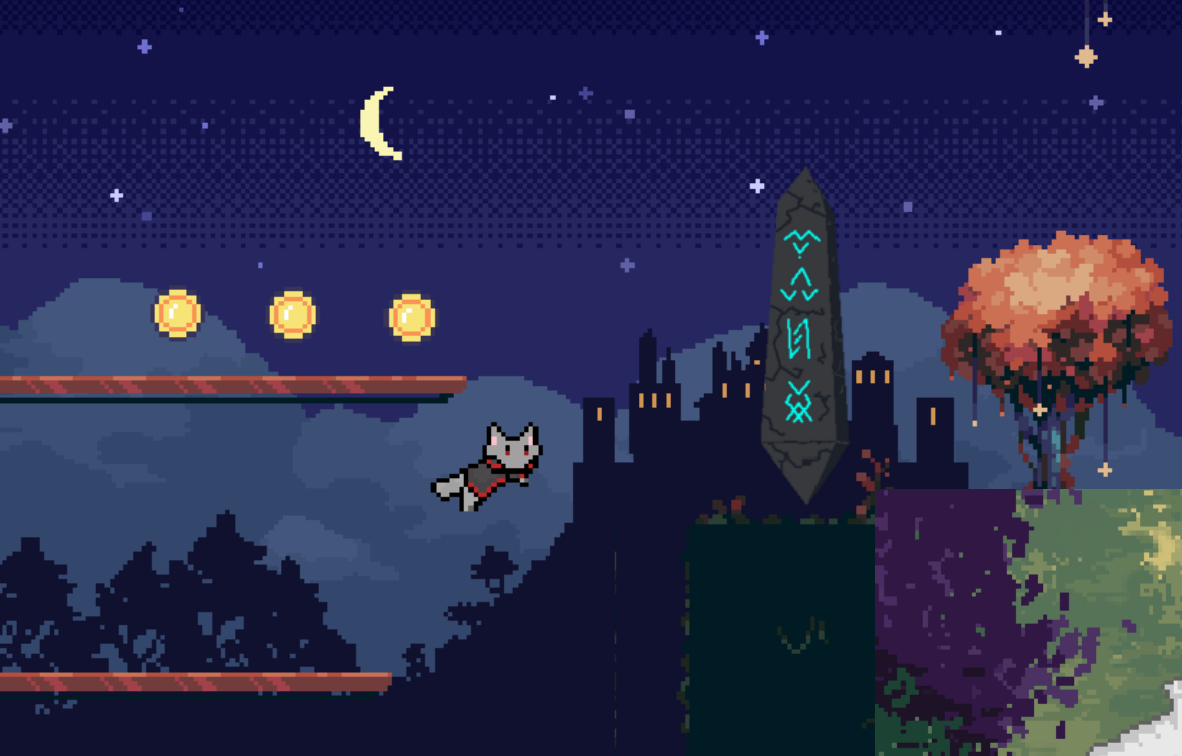
다양한 손 재활 및 발달 게임들로 이루어져 있으며, 게임 데이터를 서버로 전송하여 사용자가 게임을 통해 손 기능을 자연스럽게 향상시킬 수 있습니다.

하드웨어 영역

컴퓨터 비전 AI 기술을 이용해 사용자의 손 움직임을 실시간으로 추적하고 분석합니다.

백엔드 영역

서버로 전송된 게임 데이터를 처리하며, 웹 대시보드와 연동하여 실시간으로 데이터를 제공하는 역할을 합니다.





Hong gildong님의 대시보드입니다.

[이전 게임 플레이 기록 보기](#)

[선택하세요](#) ▾

[새로그침](#)

신체 측정 데이터

손가락 움직임 평가 점수

100점 10 ▲

양호

손목 관절 가동 범위 점수

91점 11 ▲

양호

손가락 정밀 작업 능력 점수

81점 11 ▲

양호

손 장시간 사용 지구력 평가 점수

90점 10 ▼

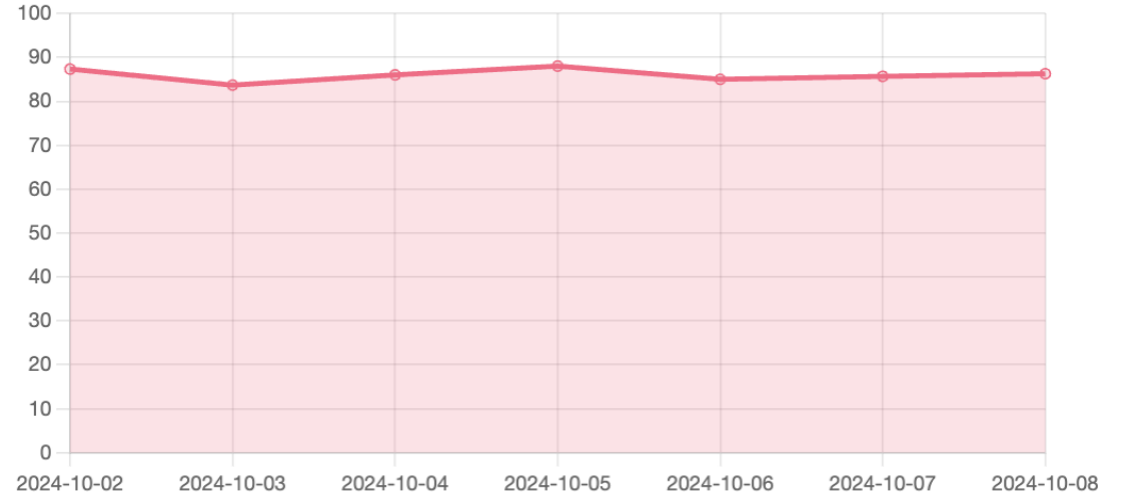
양호

손목 안정성 테스트 점수

92점 8 ▼

양호

최근 일주일 데이터 종합 점수



재활 진척도

Today (Pink) Before (Yellow)



< 2024년 10월 >

일	월	화	수	목	금	토
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

04

활용기술 스택

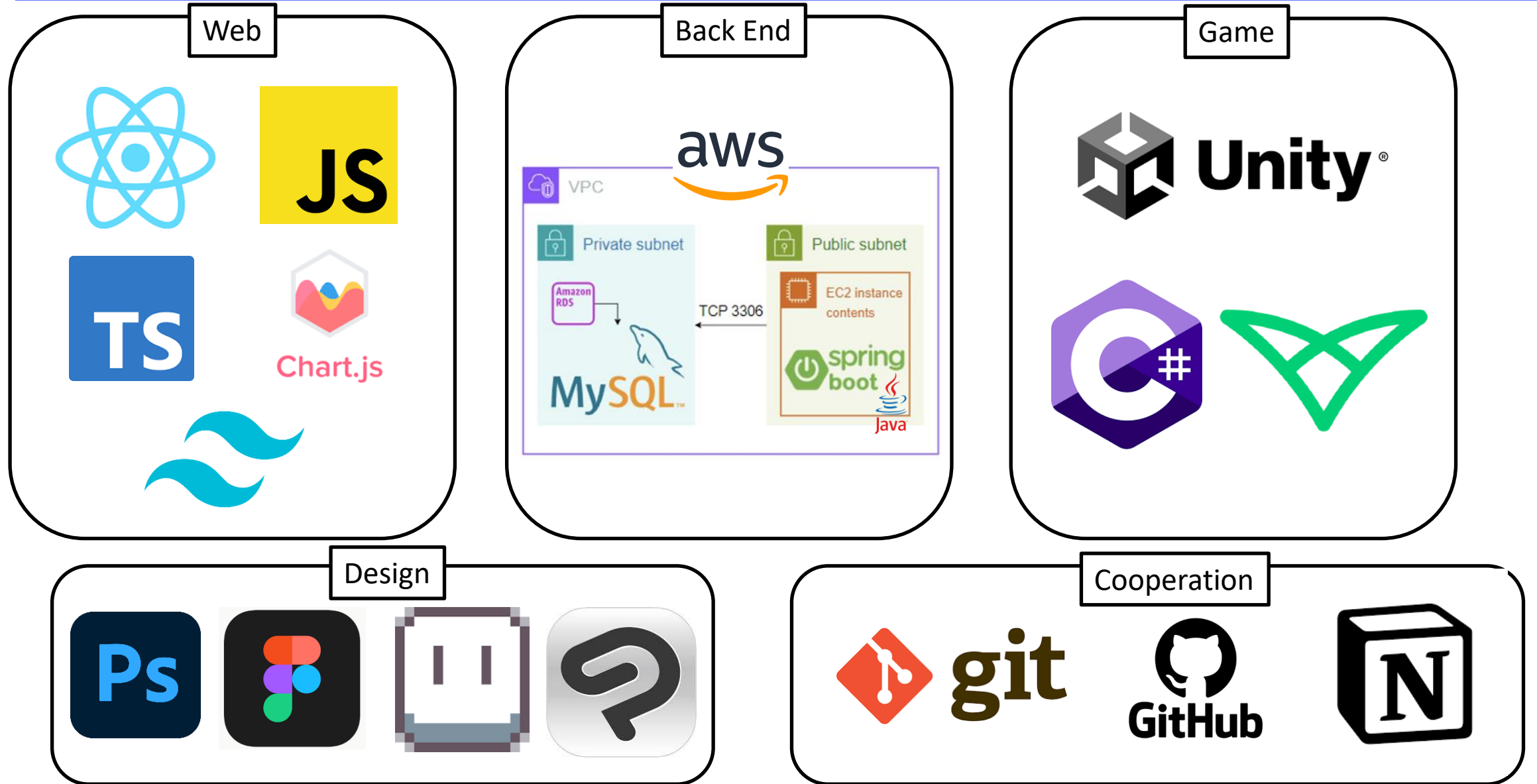
01.개발동기

02.프로젝트 소개

03.시연영상

04.사용된 기술

05.사업성 및 계획



04

Code View

01. 개발 동기

02. 프로젝트 소개

03. 시연 영상

04. 사용된 기술

05. 사업성 및 계획

DetectHandTilt

```
2 if (isLeapOn && isMoveAllow && isHit) { //림오선에 손이 인식되어 있는지 확인, 게임 속 플레이어가 움직일 수 있는 상태일 때
3   Vector3 palmNormal = hand.PalmNormal; //손바닥이 어느 방향으로 위치해있는지 3D좌표로 저장
4
5   if (palmNormal.x > 0.3f || palmNormal.x < -0.3f) { //손바닥의 위치가 좌우로 움직여야 하는 값을 넘어서는지 확인
6     horizonLeapSpeed = palmNormal.x;
7     totalWristMovement += Math.Abs(palmNormal.x);
8     wristMovementText.text = $"Wrist Movement: {totalWristMovement}";
9   }
10  }
11  else {
12    horizonLeapSpeed = 0f;
13  }
14
15  if (palmNormal.z > 0.3f || palmNormal.z < -0.3f) { //손바닥의 위치가 앞뒤로 움직여야 하는 값을 넘어서는지 확인
16    verticalLeapSpeed = palmNormal.z;
17    totalWristMovement += Math.Abs(palmNormal.z);
18    wristMovementText.text = $"Wrist Movement: {totalWristMovement}";
19  }
20  else {
21    verticalLeapSpeed = 0f;
22  }
23
24  moveDirection = new Vector2(horizonLeapSpeed, verticalLeapSpeed);
25  rb.velocity = new Vector2(moveDirection.x * moveSpeed, moveDirection.y * moveSpeed);
26
27  Vector2 clampedPosition = rb.position;
28
29  clampedPosition.x = Mathf.Clamp(clampedPosition.x, 0f, 10f);
30  clampedPosition.y = Mathf.Clamp(clampedPosition.y, 0f, 10f);
31
32  rb.position = clampedPosition;
33
34  }
35  else {
36    horizonLeapSpeed = 0f;
37    verticalLeapSpeed = 0f;
38  }
}
```

IsPointingPose

```
1 bool IsPointingPose(Hand hand) {
2   foreach (Finger finger in hand.Fingers) {
3     if (finger.Type == FingerType.Index || finger.Type == FingerType.Middle || finger.Type == FingerType.Ring || finger.Type == FingerType.Small) {
4       if (!finger.IsExtended) {
5         //검지가 퍼져있지 않다면
6         return false;
7       }
8       if (finger.IsExtended) {
9         //검지를 제외한 다른 손가락이 퍼져있지 않다면
10        return false;
11      }
12    }
13  }
14  return true; //검지만 퍼져있다면 true를 반환
}
```

OnUpdateFrame

```
2 if (frame.Hands.Count > 0) { //사용자가 손을 인식하면
3   hand = frame.Hands[0]; //인식한 손 중 맨 처음의 손
4
5   if (IsPointingPose(hand)) { //인식한 손이 가리키는 손동작을 인식한 시간을 저장
6     if (isPointing) {
7       pointingStartTime = Time.time;
8     }
9     isPointing = true;
10    //특정 손동작을 인식한 시간을 저장
11    else {
12      elapsedTime = 0f;
13      StartBar.ChangeHeight(0);
14    }
15
16    if (elapsedTime > 0.5f) {
17      if (isPointing) {
18        isPointing = false;
19      }
20      StartBar.ChangeHeight(1);
21    }
22
23    DetectHandTilt(hand);
24  }
25  else {
26    rb.velocity = Vector2.zero;
27  }
28
29  }
30
31  }
32
33  }
34
35  }
36
37  }
38
39  }
40
41  }
42
43  }
44
45  }
46
47  }
48
49  }
50
51  }
52
53  }
54
55  }
56
57  }
58
59  }
60
61  }
62
63  }
64
65  }
66
67  }
68
69  }
70
71  }
72
73  }
74
75  }
76
77  }
78
79  }
80
81  }
82
83  }
84
85  }
86
87  }
88
89  }
90
91  }
92
93  }
94
95  }
96
97  }
98
99  }
100 }
```

catFlipClap

```
1 Hand leftHand = null; // 왼손을 저장할 변수 추가
2 Hand rightHand = null; // 오른손을 저장할 변수 추가
3
4 // 프레임에 있는 모든 손에 대해 반복문 실행
5 foreach (Hand hand in frame.Hands) {
6   // 왼손일 경우 leftHand 변수에 저장
7   if (hand.IsLeft) leftHand = hand;
8   // 오른손일 경우 rightHand 변수에 저장
9   else if (hand.IsRight) rightHand = hand;
10  }
11
12  // 왼손과 오른손이 모두 존재할 경우
13  if (leftHand != null && rightHand != null) {
14    // 왼손과 오른손의 손바닥 위치를 3D 좌표로 저장
15    Vector3 leftPalmPosition = leftHand.PalmPosition;
16    Vector3 rightPalmPosition = rightHand.PalmPosition;
17
18    // 손바닥의 위치를 3D 좌표로 저장
19    Vector3 palmNormal = hand.PalmNormal;
20
21    // 각 손의 위치가 얼마나 떨어져 있는지 확인
22    float clapDistanceThreshold = 0.5f; // 박수
23    float handsDistance = Vector3.Distance(leftPalmPosition, rightPalmPosition);
24    if (handsDistance < clapDistanceThreshold) {
25      // 박수 소리를 내기 위해 클립을 재생
26      // ...
27    }
28  }
29 }
```

catMovingUPDOWN

```
1 Vector3 currentHandPosition = hand.PalmPosition;
2 Vector3 previousHandPosition = hand.PalmPosition;
3 Vector3 previousLeftHandPosition = previousLeftHandPosition;
4 Vector3 previousRightHandPosition = previousRightHandPosition;
5
6 // 이전 손 위치가 추가된다고 생각해
7 if (previousHandPosition != Vector3.zero) {
8   // 현재 손 위치와 이전 손 위치의 차이를 구함
9   Vector3 handDirection = (currentHandPosition - previousHandPosition).normalized;
10  // 손의 각도를 계산
11  float handSpeed = hand.PalmVelocity.magnitude;
12
13  // 손의 속도가 일정 속도 이하일 경우 박수 소리가 들리지 않음
14  if (handSpeed < 0.5f) {
15    // 박수 소리를 내지 않음
16  }
17  else {
18    // 박수 소리를 내기 위해 클립을 재생
19    // ...
20  }
21
22  // 손의 위치를 업데이트
23  previousHandPosition = currentHandPosition;
24
25  // 왼손과 오른손의 위치를 업데이트
26  if (hand.IsLeft) {
27    previousLeftHandPosition = currentLeftHandPosition;
28  }
29  else if (hand.IsRight) {
30    previousRightHandPosition = currentRightHandPosition;
31  }
32
33  // 박수 소리를 내기 위해 클립을 재생
34  // ...
35
36  // 박수 소리를 내기 위해 클립을 재생
37  // ...
38
39  // 박수 소리를 내기 위해 클립을 재생
40  // ...
41
42  // 박수 소리를 내기 위해 클립을 재생
43  // ...
44
45  // 박수 소리를 내기 위해 클립을 재생
46  // ...
47
48  // 박수 소리를 내기 위해 클립을 재생
49  // ...
50
51  // 박수 소리를 내기 위해 클립을 재생
52  // ...
53
54  // 박수 소리를 내기 위해 클립을 재생
55  // ...
56
57  // 박수 소리를 내기 위해 클립을 재생
58  // ...
59
60  // 박수 소리를 내기 위해 클립을 재생
61  // ...
62
63  // 박수 소리를 내기 위해 클립을 재생
64  // ...
65
66  // 박수 소리를 내기 위해 클립을 재생
67  // ...
68
69  // 박수 소리를 내기 위해 클립을 재생
70  // ...
71
72  // 박수 소리를 내기 위해 클립을 재생
73  // ...
74
75  // 박수 소리를 내기 위해 클립을 재생
76  // ...
77
78  // 박수 소리를 내기 위해 클립을 재생
79  // ...
80
81  // 박수 소리를 내기 위해 클립을 재생
82  // ...
83
84  // 박수 소리를 내기 위해 클립을 재생
85  // ...
86
87  // 박수 소리를 내기 위해 클립을 재생
88  // ...
89
90  // 박수 소리를 내기 위해 클립을 재생
91  // ...
92
93  // 박수 소리를 내기 위해 클립을 재생
94  // ...
95
96  // 박수 소리를 내기 위해 클립을 재생
97  // ...
98
99  // 박수 소리를 내기 위해 클립을 재생
100 // ...
}
```

catJumpFlip

```
2 // 프레임에 있는 모든 손에 대해 반복문 실행
3 foreach (Hand hand in frame.Hands) {
4   // 왼손일 경우 leftHand 변수에 저장
5   if (hand.IsLeft) leftHand = hand;
6   // 오른손일 경우 rightHand 변수에 저장
7   else if (hand.IsRight) rightHand = hand;
8 }
9
10 // 왼손과 오른손이 모두 존재할 경우
11 if (leftHand != null && rightHand != null) {
12   // 왼손과 오른손의 손바닥 위치를 3D 좌표로 저장
13   Vector3 leftPalmPosition = leftHand.PalmPosition;
14   Vector3 rightPalmPosition = rightHand.PalmPosition;
15
16   // 손바닥의 위치를 3D 좌표로 저장
17   Vector3 palmNormal = hand.PalmNormal;
18
19   // 각 손의 위치가 얼마나 떨어져 있는지 확인
20   float clapDistanceThreshold = 0.5f; // 박수
21   float handsDistance = Vector3.Distance(leftPalmPosition, rightPalmPosition);
22   if (handsDistance < clapDistanceThreshold) {
23     // 박수 소리를 내기 위해 클립을 재생
24     // ...
25   }
26 }
27
28 // 박수 소리를 내기 위해 클립을 재생
29 // ...
30
31 // 박수 소리를 내기 위해 클립을 재생
32 // ...
33
34 // 박수 소리를 내기 위해 클립을 재생
35 // ...
36
37 // 박수 소리를 내기 위해 클립을 재생
38 // ...
39
40 // 박수 소리를 내기 위해 클립을 재생
41 // ...
42
43 // 박수 소리를 내기 위해 클립을 재생
44 // ...
45
46 // 박수 소리를 내기 위해 클립을 재생
47 // ...
48
49 // 박수 소리를 내기 위해 클립을 재생
50 // ...
51
52 // 박수 소리를 내기 위해 클립을 재생
53 // ...
54
55 // 박수 소리를 내기 위해 클립을 재생
56 // ...
57
58 // 박수 소리를 내기 위해 클립을 재생
59 // ...
60
61 // 박수 소리를 내기 위해 클립을 재생
62 // ...
63
64 // 박수 소리를 내기 위해 클립을 재생
65 // ...
66
67 // 박수 소리를 내기 위해 클립을 재생
68 // ...
69
70 // 박수 소리를 내기 위해 클립을 재생
71 // ...
72
73 // 박수 소리를 내기 위해 클립을 재생
74 // ...
75
76 // 박수 소리를 내기 위해 클립을 재생
77 // ...
78
79 // 박수 소리를 내기 위해 클립을 재생
80 // ...
81
82 // 박수 소리를 내기 위해 클립을 재생
83 // ...
84
85 // 박수 소리를 내기 위해 클립을 재생
86 // ...
87
88 // 박수 소리를 내기 위해 클립을 재생
89 // ...
90
91 // 박수 소리를 내기 위해 클립을 재생
92 // ...
93
94 // 박수 소리를 내기 위해 클립을 재생
95 // ...
96
97 // 박수 소리를 내기 위해 클립을 재생
98 // ...
99
100 // 박수 소리를 내기 위해 클립을 재생
}
```

DetectLeftHandPose

```
2 if (IsFist(hand)) {
3   isFistR = true;
4 }
5
6 if (IsScissors(hand)) {
7   isScissorsR = false;
8   isPalmR = false;
9 }
10 else {
11   //주먹 손동작을 하고 있는지 확인
12   if (!hand.Fingers[0].IsExtended && !hand.Fingers[1].IsExtended && !hand.Fingers[2].IsExtended && !hand.Fingers[3].IsExtended) {
13     isFistR = true;
14   }
15   //가위 손동작을 하고 있는지 확인
16   else if (hand.Fingers[0].IsExtended && hand.Fingers[1].IsExtended && !hand.Fingers[2].IsExtended && !hand.Fingers[3].IsExtended) {
17     isScissorsR = true;
18   }
19   //손바닥을 펴고 있는지 확인
20   else if (hand.Fingers[0].IsExtended && hand.Fingers[1].IsExtended && hand.Fingers[2].IsExtended && hand.Fingers[3].IsExtended) {
21     isPalmR = true;
22   }
23 }
24
25 //박수
26 isScissorsR = false;
27 isFistR = false;
28 }
```

DetectRightHandPose

```
2 if (IsFist(hand)) {
3   isFistL = true;
4 }
5
6 if (IsScissors(hand)) {
7   isScissorsL = false;
8   isPalmL = false;
9 }
10 else {
11   //주먹 손동작을 하고 있는지 확인
12   if (!hand.Fingers[0].IsExtended && !hand.Fingers[1].IsExtended && !hand.Fingers[2].IsExtended && !hand.Fingers[3].IsExtended) {
13     isFistL = true;
14   }
15   //가위 손동작을 하고 있는지 확인
16   else if (hand.Fingers[0].IsExtended && hand.Fingers[1].IsExtended && !hand.Fingers[2].IsExtended && !hand.Fingers[3].IsExtended) {
17     isScissorsL = true;
18   }
19   //손바닥을 펴고 있는지 확인
20   else if (hand.Fingers[0].IsExtended && hand.Fingers[1].IsExtended && hand.Fingers[2].IsExtended && hand.Fingers[3].IsExtended) {
21     isPalmL = true;
22   }
23 }
24
25 //박수
26 isScissorsL = false;
27 isFistL = false;
28 }
```

ScoreSave

```
3 // API 응답으로 받은 데이터에 대해 저장
4 var scoreData = new PostData {
5   gameId = gameId,
6   memberId = memberId,
7   gameScore = pScore,
8 };
9
10 // JSON 데이터를 UTF-8 인코딩
11 byte[] bodyRaw = System.Text.Encoding.UTF8.GetBytes(JsonUtility.ToJson(scoreData));
12
13 // 현재 게임의 점수를 가져오고 방금 클리어한 게임 점수와 비교해
14 // 현재 점수가 더 크면 api post 통신으로 score 저장
15 IEnumerator ProcessScore(string playerId, string gameId, int pScore)
16 {
17   // api 통신을 통해 저장된 해당 게임의 점수를 가져오는 코루틴 실행
18   yield return StartCoroutine(GetScore(playerId, gameId));
19
20   // 현재 점수가 더 크면 api post 통신으로 score 저장
21   if (beforeScore < pScore) {
22     StartCoroutine(ScoreSave(currentGameId, pScore));
23   }
24 }
25
26 // 응답 대기
27 if (www.result == UnityWebRequest.Result.ConnectionError || www.result == UnityWebRequest.Result.ProtocolError) {
28   // 요청 오류 시 응답 처리
29   print(www.downloadHandler.text);
30   Debug.Log("Response Code: " + www.responseCode);
31   Debug.Log("Request Error: " + www.error);
32 }
33 else {
34   // 정상 연결되었을 시 응답 처리
35   print(www.downloadHandler.text);
36   Debug.Log("Response Code: " + www.responseCode);
37   Debug.Log("Response: " + www.downloadHandler.text);
38 }
39 }
```

ProcessScore

```
2 // 현재 점수가 더 크면 api post 통신으로 score 저장
3 IEnumerator ProcessScore(string playerId, string gameId, int pScore)
4 {
5   // api 통신을 통해 저장된 해당 게임의 점수를 가져오는 코루틴 실행
6   yield return StartCoroutine(GetScore(playerId, gameId));
7
8   // 현재 점수가 더 크면 api post 통신으로 score 저장
9   if (beforeScore < pScore) {
10    StartCoroutine(ScoreSave(currentGameId, pScore));
11  }
12 }
13 }
```

GetScore

```
4 // UnityWebRequest Get 요청 생성
5 UnityWebRequest www = UnityWebRequest.Get(url);
6
7 // 요청 전송
8 yield return www.SendWebRequest();
9
10 // 응답 대기
11 if (www.result == UnityWebRequest.Result.ConnectionError || www.result == UnityWebRequest.Result.ProtocolError) {
12   // 요청 오류 시 응답 처리
13   print(www.downloadHandler.text);
14   Debug.Log("Response Code: " + www.responseCode);
15   Debug.Log("Request Error: " + www.error);
16 }
17 else {
18   // 정상 연결되었을 시 응답 처리
19   print(www.downloadHandler.text);
20   Debug.Log("Response Code: " + www.responseCode);
21   Debug.Log("Response: " + www.downloadHandler.text);
22 }
23
24 // JSON 데이터를 UTF-8 인코딩
25 byte[] bodyRaw = System.Text.Encoding.UTF8.GetBytes(JsonUtility.ToJson(scoreData));
26
27 // 현재 게임의 점수를 가져오고 방금 클리어한 게임 점수와 비교해
28 // 현재 점수가 더 크면 api post 통신으로 score 저장
29 IEnumerator ProcessScore(string playerId, string gameId, int pScore)
30 {
31   // api 통신을 통해 저장된 해당 게임의 점수를 가져오는 코루틴 실행
32   yield return StartCoroutine(GetScore(playerId, gameId));
33
34   // 현재 점수가 더 크면 api post 통신으로 score 저장
35   if (beforeScore < pScore) {
36     StartCoroutine(ScoreSave(currentGameId, pScore));
37   }
38 }
39 }
```

05

향후 계획

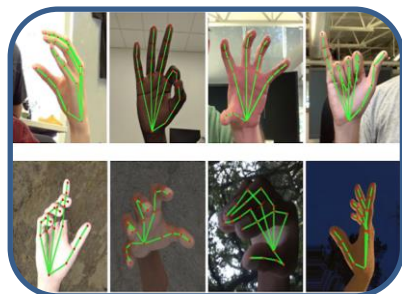
01. 개발동기

02. 프로젝트 소개

03. 시연영상

04. 사용된 기술

05. 사업성 및 계획



학회 성과 발표

- 한국 차세대 컴퓨팅 학회



유아 소근육 운동 게임 창업

- AR/VR 연동



산업협력 EXPO 체험형 부스 운영

감사합니이다

Q & A
