
오픈소스 SW프로젝트03 최종보고서

주제: 손인식을 활용한 게임

컴퓨터공학과

20171614

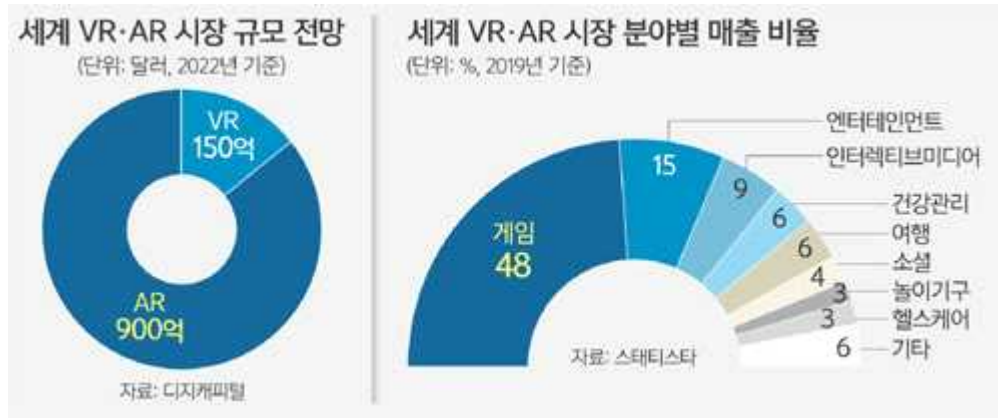
노순건

목 차

I 서론	3
II 현황과 관련 제품들	3
III 기대효과	4
IV 진행 과정	4
1. 1/9~11	4
2. 1/12	5
3. 1/13~16	6
4. 1/17~19	8
V 실행 과정 및 결과	10
VI 사용한 오픈소스와 적용한 라이선스	11
VII 결론 및 느낀점	12

I. 서론

4차산업 시대에 진입하면서 VR, Ar와 같은 기술들이 주목받기 시작했다. 이에 따라 VR-AR 시장규모는 성장 가능성이 크게 평가되었고 수요 또한 높아져 시장규모도 성장했으며¹⁾, 그중 제일 주목받고 있는 시장은 게임 시장이다.²⁾



특히 VR 기술과 VR 기기의 센서와 카메라를 이용³⁾하여 hand-physics-lab⁴⁾와 같이 아무런 입력 기기 없이 손만을 사용하여 플레이할 수 있는 게임들이 실험적으로 개발되고 있다.⁵⁾ 이처럼 가상 속 객체를 직접 손으로 만지며 플레이할 수 있는 게임을 VR 기기를 통해서가 아니라 카메라를 통해서, 인공지능 객체 인식기를 사용하면 만들 수 있을 것으로 생각하여, 이 프로젝트를 진행하게 되었다.

II. 현황과 관련 제품들

카메라 기반 게임은 Xbox나 Playstation 과 같은 거치형 콘솔 게임기에서 독보적이었다. 특히 Sony의 Playstation 은 카메라에 대해 더욱 신경 썼는데, Playstation 2의 eye toy에서부터 현재의 PS Camera 까지 카메라를 게임에 활용하기 위해 계속해서 하드웨어를 개발해왔다.⁶⁾ 이를 사용하는 게임 중 유명한 것은 전신을 카메라에 인식시켜 박자에 맞춰 춤을 추는 Just Dance와 Ar 기술을 활용하여 가상의 로봇과 상호작용할 수 있는 게임인 The PlayRoom이 있다. 이와 같은 카메라 기반의 게임들도 있지만, 그 외에도 Alien: Isolation 이나 Surgeon Simulator와 같이 선택적, 간접적으로 카메라를 사용하는 게임들도 있었다.⁷⁾ 하지만 이렇게 카메라와 관련된 게임들이 많이 있음에도 불구하고, 사람들은 그 존재들을 잘 알지 못했다. 게임 관련 사이트인 steam의 한 사용자는 이러한 게임들은 성공하지 못했으며, 그러므로 PC 게임에서 게임개발자들이 이러한 시도를 하지 않는다고 말한다.⁸⁾

그의 말처럼 PC에서 플레이할 수 있는 게임들은 찾기 어려웠다. 그나마 찾을 수 있었던 게임은 2021년에 출시된 Before Your Eyes라는 게임이었다. 이 게임은 웹캠을 통해 눈을 인식하고, 눈을 감으면 이야기가 순식간에 진행되는 독특한 방식으로 카메라를 이용했다. 이와 같은 방식을 플레이어들은 혁신적인 상호작용, 참신한 게임방식이라며 호평을 남겼으며, 9개의 상을 받았다.⁹⁾

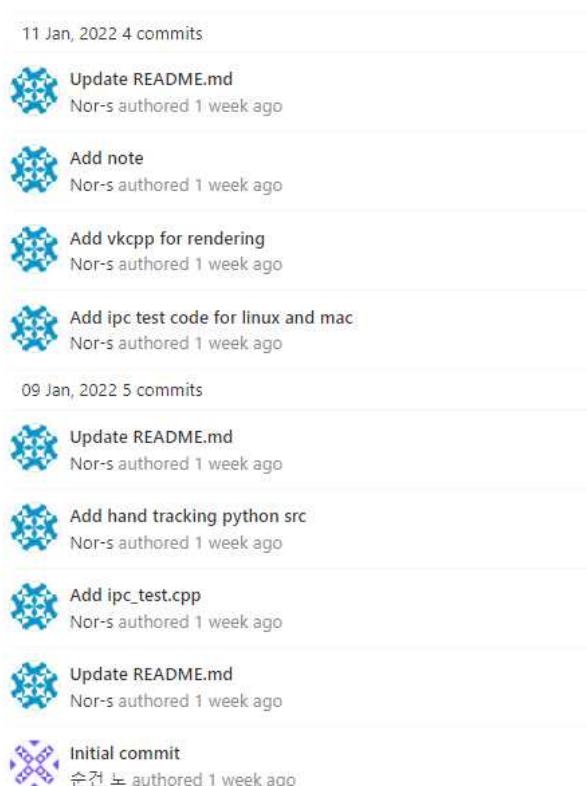
III. 기대효과

1. 앞서 본 Before Your Eyes와 같이 창의적으로 카메라 인식을 활용하면, 플레이어들에게 낮은 즐거움을 느끼게 해줄 수 있다.
2. 게임은 치매 예방에 도움이 된다. 하지만 만약 카메라를 통해 신체를 인식시켜 게임을 한다면, 두뇌 사용능력과 함께 신체 활동성을 증가시킬 수 있다. 이러한 기대효과는 VR 기기를 이용한 게임에서도 마찬가지인데, 카메라를 사용한다면, 더욱더 저렴한 가격과 몸이 다소 불편한 노년층들에게 눈과 목에 부담을 덜어줄 수 있다.¹⁰⁾
3. 스마트폰과 같은 간단한 인터페이스 제공으로 노년층이나 유아층에 맞춤 PC 게임을 제공할 수 있다.
4. 아직 완벽하지 않은 인식기술이지만, 오히려 이점을 장점으로 삼아 플레이어를 괴롭히고, 이러한 플레이어의 우스꽝스러운 모습을 즐겁게 볼 수 있는 게임으로 만들면 많은 사람에게 즐거움을 줄 수 있다. 선행로 향아리 게임, 휴먼 폴 플랫폼이 있다. ¹¹⁾

IV. 진행 과정

깃 랩 주소: <https://sw-git.chosun.ac.kr/nsg/hand-window>

1. 1/9~11



- 제일 먼저 주제를 정하고 한 것은 손을 인식할 수 있는 오픈소스를 찾는 것이었다.

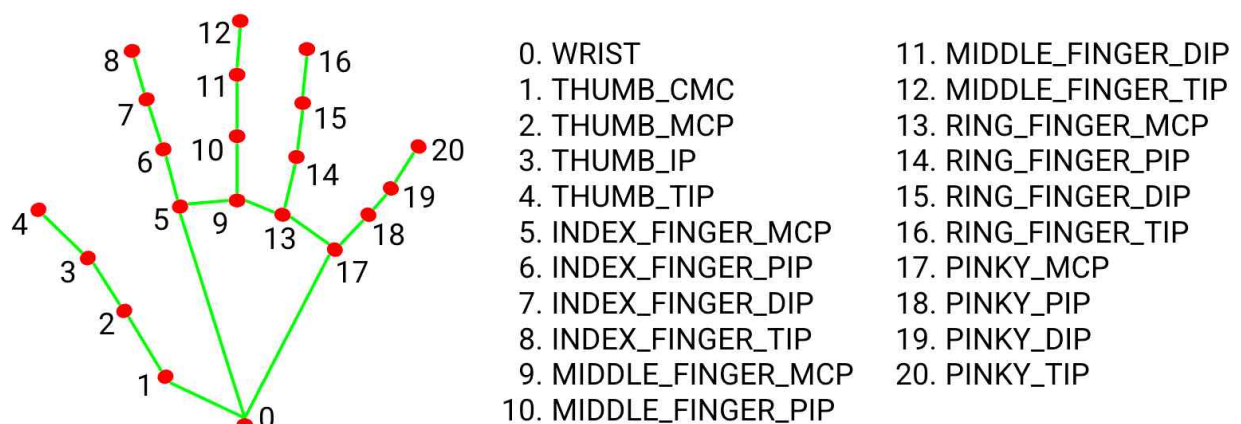
- 그 결과, 구글에서 만든 오픈소스 크로스 플랫폼인 MediaPipe를 찾았다.
- MediaPipe는 커스텀할 수 있는 ML 솔루션을 제공하며, 특히 스트리밍, 라이브에 특화되어있는 오픈소스이다.
- 그러므로 이 오픈소스의 손인식 솔루션을 사용하여 손 인식기능을 게임 화면에 렌더링하고자 했다.
- MediaPipe는 여러 언어를 지원하는데, Python MediaPipe가 제일 정보가 많았기 때문에 Python으로 코드를 작성했다.
- 그리고 이전에 C++로 작성했던 그래픽스 관련 소스를 활용하기 위해 이 파이썬 프로세스와 C++ 프로세스 간에 통신할 수단이 필요했다.
- 찾아본 통신할 수단은 총 3가지였다. 공유메모리를 활용하는 것, redis를 활용하는 것, embedded python으로 c++에서 파이썬 코드를 사용하는 것이었다.
- 이 중 프로세스 간 통신 기법인 공유메모리를 활용했으며, 그 이유는 성능이 가장 뛰어나다고 알려졌기 때문이었다.¹²⁾
- 이를 위해 작성한 파이썬 코드 일부는 다음과 같다.

```

58 +     for hand_landmarks in results.multi_hand_landmarks:
59 +         # ipc 공유메모리를 위한 변수
60 +         shm = mmap.mmap(0, 512, "Local\\MPipe")
61 +         if shm:
62 +             for idx in range(0, 21):
63 +                 shm.write(struct.pack('f', hand_landmarks.landmark[idx].x))
64 +                 shm.write(struct.pack('f', hand_landmarks.landmark[idx].y))
65 +                 shm.write(struct.pack('f', hand_landmarks.landmark[idx].z))

```

- 이 코드는 손의 landmarks를 공유메모리를 가져오는 것으로 아래의 점의 좌표들을 효율적으로 C++ 프로세스에서 보이도록 한다.



- 또한, 이 기간 동안 학습한 것을 잊지 않기 위해, 다시 볼 가능성을 위해 note 폴더에 기록했다.

2. 1/12

- 이전에 작성했던 그래픽스 API 중 하나인 Vulkan API를 사용하여, 공유메모리에 접근하고, 이를 렌더링하였다.

12 Jan, 2022 2 commits



Fix minor error

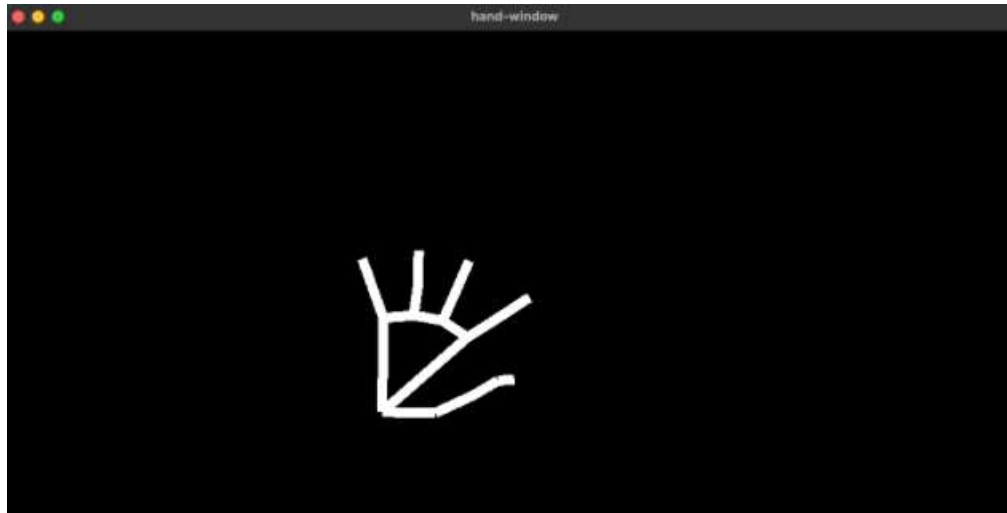
Nor-s authored 1 week ago



Add hand object for rendering

Nor-s authored 1 week ago

- Fix minor error는 손 객체에 할당한 메모리를 해제해주지 않아 생긴 오류를 해결하기 위한 커밋으로, 프로그램 끝부분에 정리해주는 함수를 사용하여 해결하였다.



3. 1/13~16

16 Jan, 2022 3 commits



Update note

Nor-s authored 3 days ago



Add interaction for hand and box

Nor-s authored 3 days ago



Add Box2D object

Nor-s authored 3 days ago

15 Jan, 2022 3 commits



Update note

Nor-s authored 4 days ago



Add resize function for hand object && Fix python code for invert y-axis

Nor-s authored 4 days ago



Update note

Nor-s authored 4 days ago

14 Jan, 2022 1 commit



Update note

Nor-s authored 5 days ago

13 Jan, 2022 1 commit



Update note

Nor-s authored 6 days ago

- 이 기간에는 물리 엔진 오픈소스인 Box2D를 사용하여 손객체와 다른 객체가 상호작용할 수 있도록 구현했는데, Box2D를 사용하기 위해, Box2D 공식문서를 읽었고, note 폴더에 다시 볼 가능성이 있으므로 잊지 않기 위해 기록하였다.
- 아래는 Box 객체를 초기화하는 코드로, 마찰, 밀도, 모양 등을 설정한다.

```
void BoxObject::init_box_object(const VkExtent3D &extent,
                               const glm::vec3 &position)
{
    b2BodyDef bodyDef;
    glm::vec2 meter_pos = BoxManager::get_meters(position.x, position.y);
    glm::vec2 meter_ext = {BoxManager::get_meters(extent.width), BoxManager::get_meters(extent.height)};

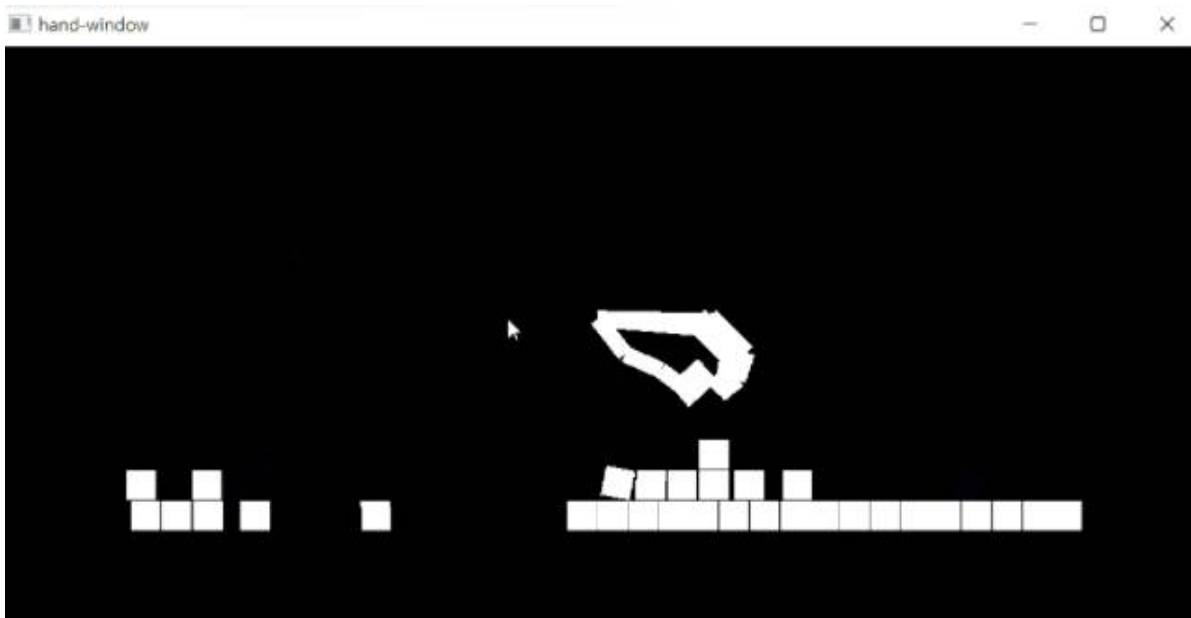
    bodyDef.type = b2_dynamicBody;
    bodyDef.position.Set(meter_pos.x, meter_pos.y);
    bodyDef.userData.pointer = reinterpret_cast<uintptr_t>(this);

    body_ = BoxManager::getInstance()->create_body(&bodyDef);

    b2PolygonShape dynamicBox;
    dynamicBox.SetAsBox(meter_ext.x / 2.0, meter_ext.y / 2.0f);

    b2FixtureDef fixtureDef;
    fixtureDef.shape = &dynamicBox;
    fixtureDef.density = 1.0f;
    fixtureDef.friction = 0.3f;
```

- 이 객체들을 생성하고 렌더링해보니 여러 오류들을 직면했다. 큰 오류는 두 가지로 하나는 충돌이 제대로 일어나지 않는 에러, 나머지 하나는 손 떨림 현상이 심하다는 것이었다.












- 이를 해결할 방안을 모색하였고, 해결방안은 다음과 같았다.
- 충돌오류 해결방안: box2d의 객체들은 속도와 힘으로 물체를 시뮬레이션한다. box2d에대한 문서를 읽기 전, 먼저 렌더링을 하기 위해 손의 모든 마디를 위치(position)기반으로 주어진 좌표에 객체가 위치하도록 코드를 작성하였다. 따라서 box2d를 적용한 이후 프레임마다 손의 마디들의 위치는 순간순간마다 순간 이동한 것처럼 이동한 것으로 다른 객체들이 인식하게 되었고 그 과정에서 box2d 시뮬레이션 결과는 잘못된 결과를 내놓게 되는 것이었다. 그러므로 손 객체의 이동하

는 방법을 변경하면 문제를 해결할 수 있다.

- 손 떨림 해결방안: 칼만 필터와 같은 필터들을 사용하여 잡음을 제거하는 것으로, 칼만 필터는 물체의 측정값에 확실적인 오차가 포함되고, 또한 물체의 특정 시점에서의 상태가 이전 시점의 상태와 선형적인 관계를 맺고 있는 경우 적용 가능한, 물체의 위치를 추정하는 필터이다.
- 이 오류들을 해결하고 만들고자 했던 게임인 물체들과 상호작용하는 퍼즐게임을 만들려 했으나, 남은 기간 동안 다 해낼 수 없다고 생각하여 다른 방향으로 프로젝트를 진행하였다.

4. 1/17~19

	Update note Nor-s authored 1 day ago
	Add sleep for vscode compound launch Nor-s authored 1 day ago
	objects never sleep Nor-s authored 1 day ago
	Fix window setting Nor-s authored 1 day ago
18 Jan, 2022 5 commits	
	Add gesture for spacebar Nor-s authored 2 days ago
	Add python code for pinball hand tracking Nor-s authored 2 days ago
	Update CMakeLists.txt Nor-s authored 2 days ago
	Fix Errors from differences in pixel and frame size Nor-s authored 2 days ago
	Add opensource game(SpaceCadetPinball) Nor-s authored 2 days ago

- 그 방향은 제스처를 사용하여 게임을 플레이하는 것이었으며, 그 게임은 오픈소스에서 찾았다.
- 오픈소스 게임 후보는 cut the rope를 모작한 깃허브의 프로젝트와 SpaceCadetPinball 이라는 윈도우의 핀볼게임을 역공학을 통해 코드를 재구현한 오픈소스 프로젝트 두 개로 좁혀졌고, 최종적으로 핀볼게임을 선택했다.
- 이 두 개의 게임을 선택한 이유는 물리 기반 게임 중 오픈소스인 것 그리고 C++를 사용한 것을 찾다 보니, 이 두 개만 찾을 수 있었기 때문이다.
- SpaceCadetPinball 소스코드를 다운받고, 빌드하는데 문제가 생겨, CMakeLists.txt를 수정해서 해결하였고, 빌드하고 플레이해보니 손의 움직이는 각도에 따라 핀볼을 치는 막대의 각도를 세세

하게 조정할 수 있겠다고 생각했으며, 또한 손의 속도에 따라 막대의 속도도 변하게 할 수 있다고 생각했다.

- 그리고 이 게임의 소스코드를 분석하고 수정하여 막대를 움직여 봤으나, 충돌이 일어나지 않는 오류가 발생하였다.



- 이 오류는 파이썬 모듈인 keyboard를 사용하여 제스처당 키입력을 할당하여 해결하였다.
- 그로 인해 앞에서 생각한 막대의 세세한 조정은 할 수 없지만, 오리지널 게임 그 자체는 플레이 할 수 있도록 만들 수 있었다.
- 제스처는 수학적 계산을 통해 관절의 각도에 따라 제스처를 지정하였고 각도를 계산하는 코드는 아래와 같다.

```
def get_angle(p1, p2, p3):  
    a = math.sqrt((p1.x-p2.x)**2 + (p1.y -p2.y)**2)  
    b = math.sqrt((p3.x-p2.x)**2 + (p3.y -p2.y)**2)  
    c = math.sqrt((p1.x-p3.x)**2 + (p1.y -p3.y)**2)  
    x = (a*a +b*b - c*c)/(2*a*b)  
    return math.acos(x)
```

- 오른손 검지: / (각도 110를 기준으로 위아래)
- 왼손 검지: z (각도 110을 기준으로 위아래)
- 양손의 소지: 스페이스 바

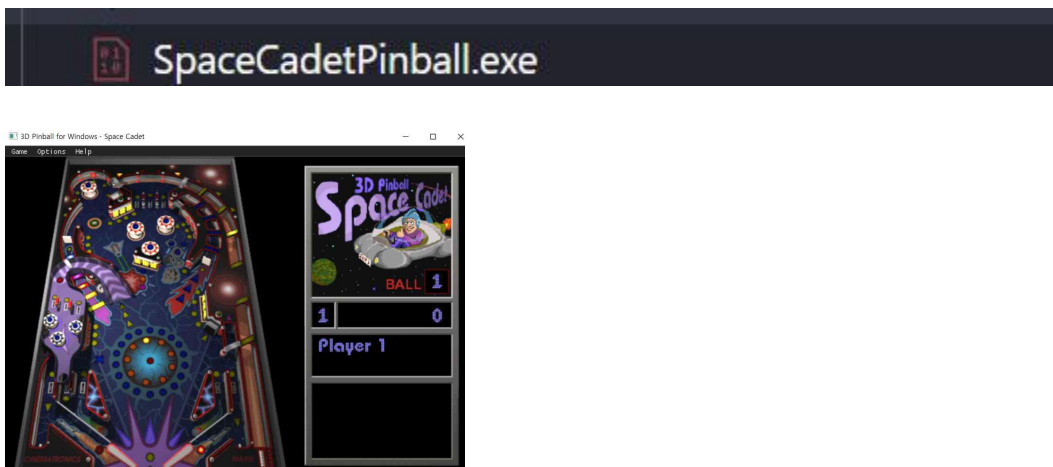
V. 실행 과정 및 결과

최종 프로그램을 빌드하기 위해서 필요한 의존성 있는 라이브러리는 다음과 같다.

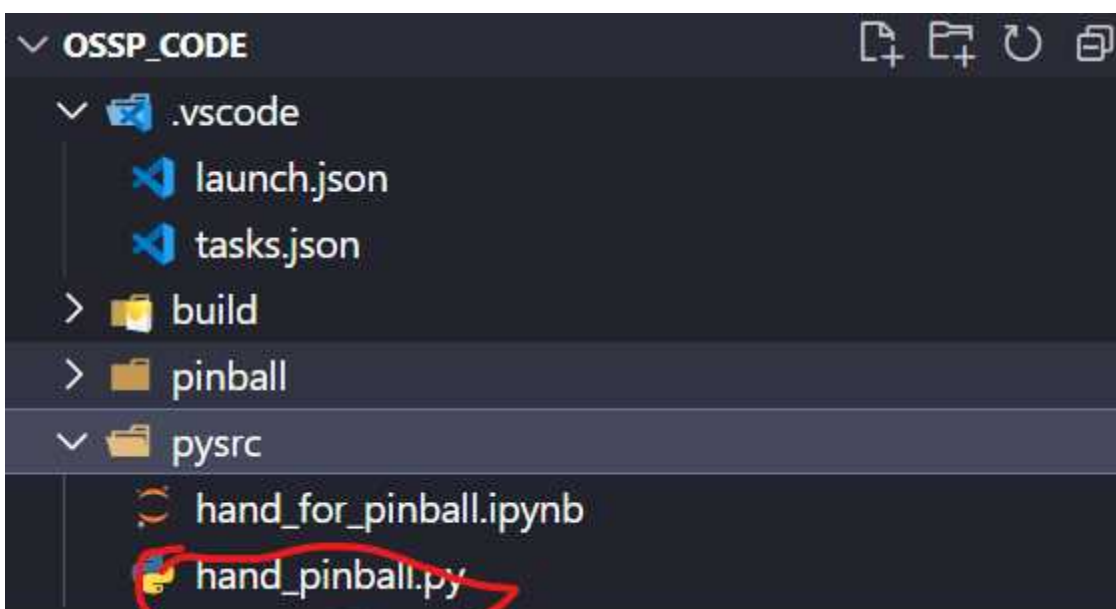
1. C++11: SDL2, SDL2_mixer
2. python3.8.11 : cv2, sys, mediapipe, keyboard, logging, math

python 모듈들은 pip 으로 설치할 수 있고, C++게임인 핀볼을 빌드하기 위해서는 SDL2와 SDL2_mixer는 cmake가 find_package로 적절히 경로를 찾을 수 있는 곳에 위치해야 하거나 직접 지정해 줘야 한다.

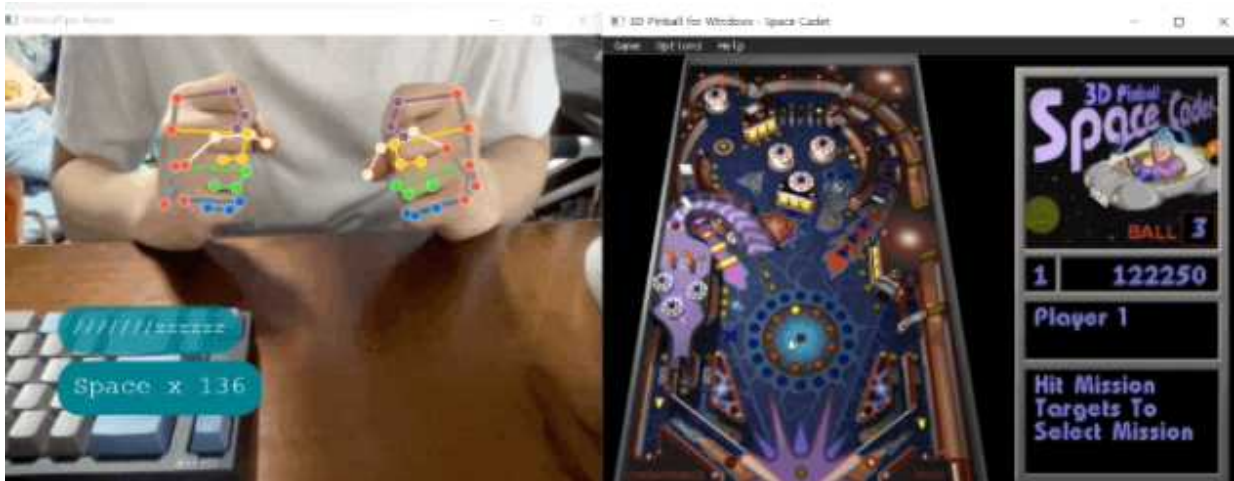
실행하는 과정은 다음과 같다. pinball/bin 폴더 안에 있는 SpaceCadetPinball.exe를 실행하면 게임창이 뜬다.



그리고 mediapipe를 사용한 코드인 pysrc의 hand_pinball.py를 실행하면



아래와 같이 손 제스처로 게임을 플레이할 수 있다.



실행결과 영상: <https://www.youtube.com/watch?v=WwyjKpvAi4c>

VI. 사용한 오픈소스와 적용한 라이선스

사용한 오픈소스들을 명시하기 위해 NOTICE.md 파일을 생성하였고, 그 안에 라이선스 사본을 명시하여 저작권자를 표시해주었다.

Name	Last commit	Last update
external/stb	Add vkcpp for rendering	1 week ago
game	Fix window setting	1 day ago
note	Update note	1 day ago
pysrc	Add sleep for vscode compound launch	1 day ago
resources	Add vkcpp for rendering	1 week ago
src	objects never sleep	1 day ago
test	Fix minor error	1 week ago
.gitignore	objects never sleep	1 day ago
CMakeLists.txt	Update CMakeLists.txt	2 days ago
LICENSE	Add LICENSE	1 day ago
NOTICE.md	Merge branch 'main' of http://sw-git.chosun.ac.kr:8091/nsg/hand-window	1 day ago
README.md	Update note	1 day ago

- mediapipe: AI 프레임워크, Apache-2.0 라이선스
- box2d: 물리 엔진, MIT 라이선스
- stb: 이미지 파일을 로드하는 라이브러리(텍스처), MIT 라이선스
- glm: OpenGL을 위한 수학라이브러리, MIT 라이선스
- glfw: OpenGL을 위한 멀티플랫폼 라이브러리, zlib 라이선스
- SpaceCadetPinball: 윈도우즈의 핀볼을 역공학을 통해 재구현한 리소스파일은 포함되지 않은 오픈소스 게임, MIT license

수정한 오픈소스는 SpaceCadetPinball 게임으로, 수정한 부분은 빌드하는 스크립트로 나의 환경에 맞게 수정했어야했다.

적용한 라이선스는 Apache2.0 라이선스이다. 그 이유는 사용한 오픈소스 중 Apache2.0라이선스가 있기 때문이다. 예를 들면 MIT와 같은 무제약 라이선스를 사용하게 될경우 배포할 때 사용자가 이 Apache2.0의 제약들을 지키지 않을 수 있어 문제가 생길 가능성이 있기 때문이다. 그러므로 Apache2.0라 이선스를 유지하였다.¹³⁾

VII. 결론 및 느낀점

이 프로젝트를 진행하면서 카메라 기반 게임에 대한 결론은 다음과 같다.

1. 카메라 인식이 밝기에 대해 까다로우므로 사용자의 카메라, 조명 등 사전 준비가 필요하다. 이는 즉, 플레이어에게 부담을 느끼게 해주며, 불편함을 느끼게 해준다.
2. 하지만 보는 사람에게는 흥미를 유발해주며, 요즘 게임 소비형태인 게임을 하는 유튜버를 시청하는 구조를 이용하면 충분히 소재로서 의미가 있다.
3. 2D 퍼즐 게임이나 샌드 박스류의 게임, 물리 기반 2D 게임에서의 활용 가능성이 어느 정도 있다.
4. 하지만 게임을 플레이하는 기기는 PC, 콘솔로 제한된다. 스마트폰과 같은 모바일 기기로는 화면이 작고 카메라의 범위가 좁기 때문에 불편함을 유발한다
5. 결국, II에서 스팀의 한 게이머가 말한 것처럼, 만드는 시간과 노력과 비교하면 얻는 가치는 적기 때문에 게임회사에서 만들지 않는 것 또한 맞는 말 같다. 하지만 다양한 게임을 추구하며 어떠한 게임의 방식을 제안하는 인디 게임에서는 충분히 가능성이 있다고 본다.

마지막으로 느낀 점은 다음과 같다.

이 강의를 수강하면서 느낀 점은 나는 아무것도 할 줄 모른다는 것이었다. 프로젝트를 기획하는 능력 또한 모자랐고, 내가 할 수 있는 능력 내에서 구현할 수 있는 아이디어 또한 떠올리기 힘들었다. 주제를 정하고 나서도 방황하였다. 시간을 효율적으로 분배하지 못했고, 정한 주제의 완성한 프로젝트의 모습 또한 희미했을 뿐이었다. 게임을 만들고자 했으면, 오픈소스 게임 엔진을 찾고, 최소기능제품을 만들어야 했었다. 하지만 그러지 못했고 그런 생각을 떠오른 것은 프로젝트 마감일이 발등에 떨어졌을 때였다.

오픈소스에 대한 느낀 점은 오픈소스를 가져다 사용하는 것 또한 공부할 게 많다는 것이었다. 예를 들면 물리 엔진인 box2d를 사용하기 위해 소비한 시간은 2일이었던 점이다. 또한, 가져온 코드를 빌드할 때에도 문제가 많이 생겼는데, 저작권자의 개발 환경과 나의 환경을 어느 정도 일치하게 해야 했기 때문이다. 또한, 라이선스에 관해서도 신경 써야 할 것들이 많이 있었다. 사용한 오픈소스들에 의해 내가 적용해야 할 라이선스가 제한되는 것, 사용해서는 안 되는 라이선스들이 있다는 것을 명심해야겠다고 생각했다.

프로젝트에 대해 느낀 점은 생각했던 기능들을 전부 집어넣지 못했다는 것에 아쉬움을 느꼈다. 핀볼

게임을 손 제스처로 플레이할 수 있게 만들고자 했었는데, 그 이유는 손의 움직임에 따라 막대의 각도를 세세하게 조정할 수 있으면 재미있겠다고 생각해서였기 때문이었다. 하지만 이 기능을 추가하려면 SDL이라는 또 다른 오픈소스를 이해해야 했으며, 그 구조 또한 공부해야 했었다.

이처럼 많이 부족한 프로젝트였지만 부족한 점들을 다시금 일깨워 준 프로젝트였고, 앞으로 더 큰 규모의 더 나은 남 보기에 부끄럽지 않은 프로젝트를 하기 위해 자기계발을 게을리하지 않고 이번 방학 마지막까지 열심히 공부해야겠다고 느꼈다.

-
- 1) 이종혁, VR·자율주행 성장에 공간정보 국내 산업 규모만 10조 원 육박, 매일경제, 2022.01.05, <https://www.mk.co.kr/news/economy/view/2022/01/13738/>
 - 2) 김건호, 하는 게임서 보는 게임으로... 새로운 진화 시동 건 VR, 세계일보, 2020.07.21, <http://m.segye.com/view/20200721518903>
 - 3) Bobby Carlton, "Hands-On: Oculus Quest Hand Tracking Feels Great, But It's Not Perfect", vrscout, 2019.08.01, <https://vrscout.com/news/hands-on-oculus-quest-hand-tracking/>
 - 4) <https://www.holonautic.com/hand-physics-lab>
 - 5) <https://uploadvr.com/best-oculus-quest-hand-tracking/>
 - 6) <https://www.neogaf.com/threads/looking-back-at-playstations-camera-based-experiments.1540812/>
 - 7) <https://www.thegamer.com/playstation-4-camera-best-games-ranked/>
 - 8) <https://steamcommunity.com/discussions/forum/0/2284960483092134448/>
 - 9) https://store.steampowered.com/app/1082430/Before_Your_Eyes/
 - 10) 조재민, 게임이 젊은 층의 전유물?-게임 활용 치매 예방 확대, 디멘시아 뉴스, 2021.11.05, <https://www.dementianews.co.kr/news/articleView.html?idxno=4486>
 - 11) 김혜지, 망한 조작감의 역습? 향아리 게임과 휴먼 폴 플랫폼의 성공 비결, 앱스토리, 2020.07.08, <https://news.appstory.co.kr/report13490>
 - 12) <https://stackoverflow.com/questions/1235958/ipc-performance-named-pipe-vs-socket>
 - 13) <https://www.quora.com/Can-I-use-Apache-licensed-source-code-and-redistribute-it-as-MIT-license>