



## Portfolio

**열정적인**  
**프론트엔드**  
**개발자**

**박종준**입니다





**박종준 Park Jongjun**

1991. 05. 24  
010 - 5002 - 1524  
poiufgin7373@naver.com

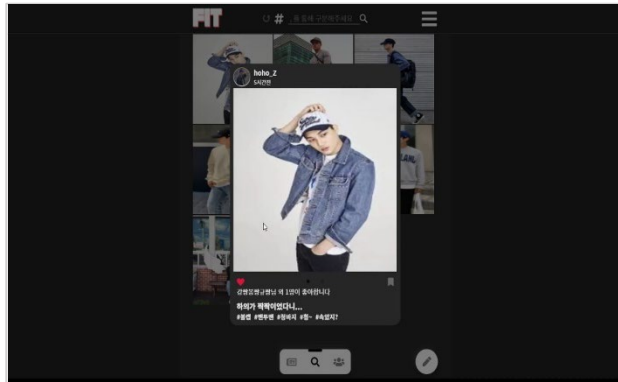
저는 재료공학과를 전공했습니다. 목표를 향해 끊임없이 달려가는 도중에 문득 의문이 들었습니다. ‘무엇이 날 행복하게 해줄까?’ 생각은 깊어졌고 정답은 그리 멀지 않은 곳에 있었습니다.

바로 즐거움입니다. 어릴 적 꿈이었던 프로그래머는 즐거움을 주었고 열정을 불어넣어 주었습니다.

학업	수상내역	보유기술
2010.03 ~ 2016.02 충남대학교 재료공학과  2016.03 ~ 2018.02 (수료) 광주과학기술원 신소재공학부	2020.11 삼성 청년 SW 아카데미 3기 자율프로젝트 전국 1위 (cobit)	Python HTML CSS JavaScript Vue.js Django Unity C#
교육과정	자격사항	
2020.01 ~ 2020.12 삼성 청년 SW 아카데미 3기	2020.08 정보처리기사	

## 프로젝트

### 웹디자인



20.07.08 ~ 20.08.21

패션 SNS

**FIT ON YOU**  GitHub

### 블록체인

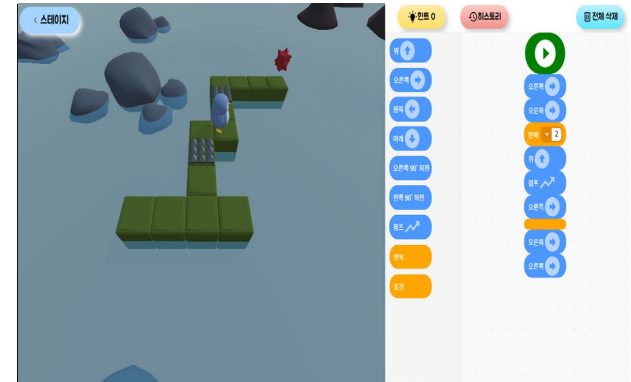


20.08.24 ~ 20.10.08

이더리움 이력서

**JOB 이로운 생활**  GitHub


### 3D - 웹



20.10.12 ~ 20.11.27

코딩 교육 플랫폼

**COBIT**  GitHub

 삼성 청년 SW 아카데미 3기  
자율프로젝트 전국 1위

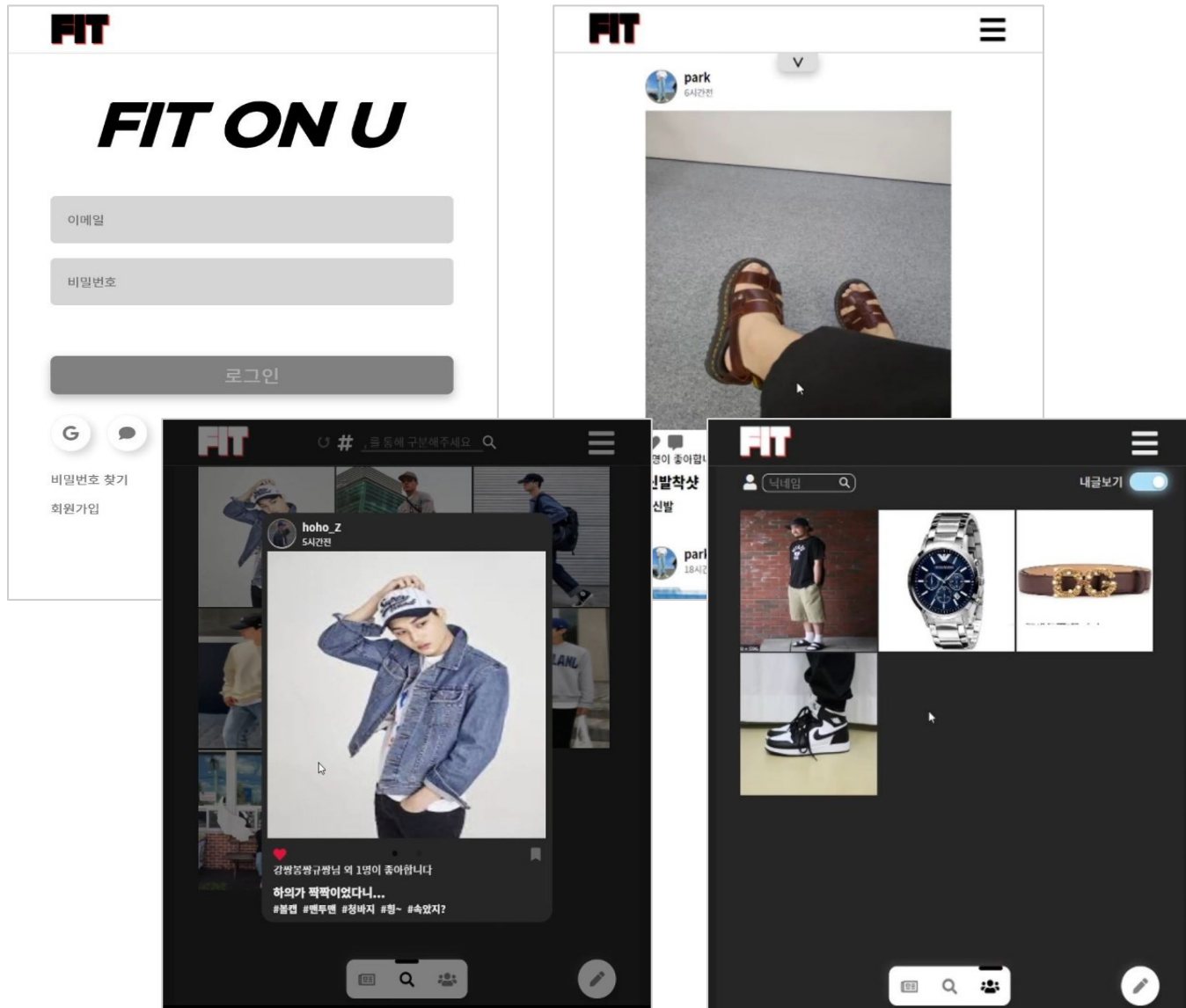
웹디자인

# 패션 SNS FIT ON YOU

작업기간 20.07.08 ~ 20.08.21  
참여인원 5명  
기여도 30%

패션 정보 공유와 리셀 커뮤니티를 결합한  
패션 소셜 네트워크 서비스





## 목표

대부분의 SNS는 단일 태그 검색만 가능하기 때문에 다양한 조합의 패션 정보를 얻기 힘들다. 따라서 FIT ON YOU는 다중 태그 검색을 통해 원하는 정보를 얻을 수 있으며 검색 기록에 따라 각종 피드를 추천 받을 수 있다. 또한 리셀 커뮤니티를 추가하여 사용자간의 중고거래 솔루션도 제시했다.

또한, 주제가 웹디자인이기 때문에 UI/UX에 집중하여 작업을 진행했다.

## 담당역할

프론트엔드 (와이어프레임, 레이아웃, 웹 퍼블리싱)

## 기술스택

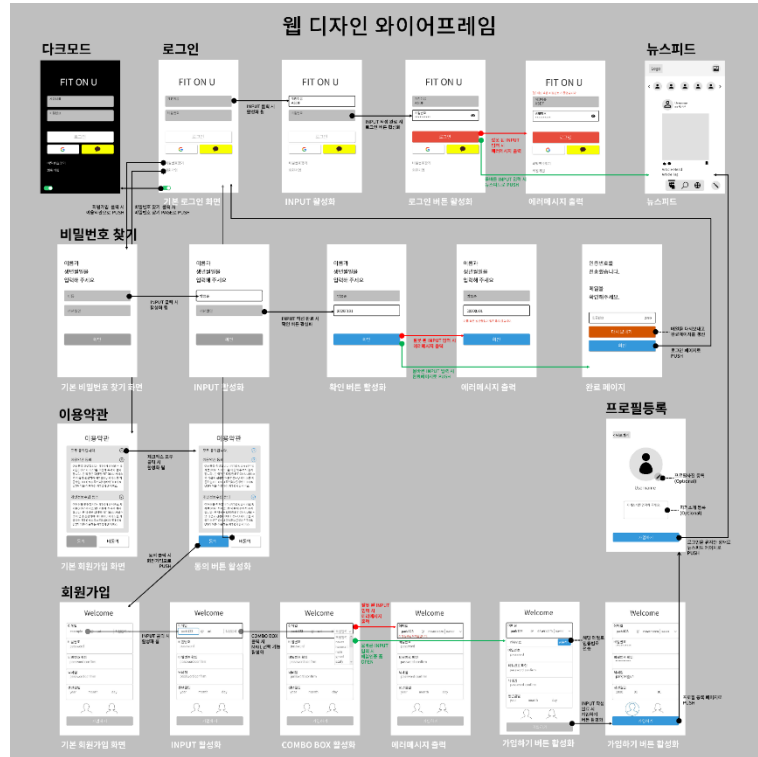
HTML, CSS, JavaScript, Vue.js

## 특이사항

반응형 웹, 다크모드 구현, UI 라이브러리(bootstrap, vuetify) 미사용

# Wireframe

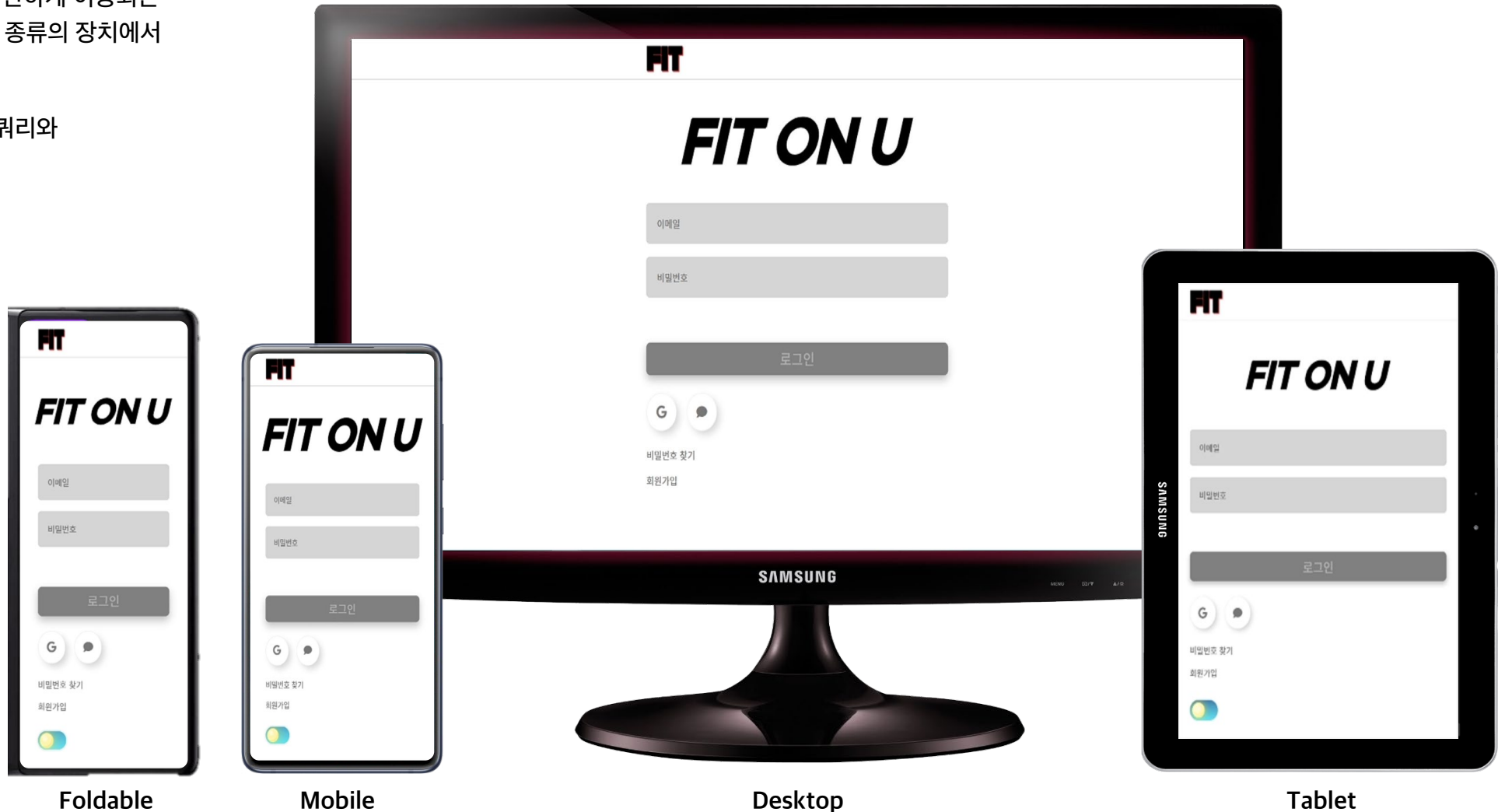
바로가기



## 반응형 웹

SNS는 장치의 종류와 무관하게 이용되는 플랫폼이기 때문에 모든 종류의 장치에서 사용이 가능하도록 함

반응형은 CSS의 미디어쿼리와 vw, vh를 사용하여 구현



Foldable

Mobile

Desktop

Tablet

## Dark Mode

사용자의 편의를 위해  
Light Mode와 Dark Mode를 구현

```
darkOn() {  
  const Dark = this.$cookies.get('dark')  
  if (Dark === 'on') {  
    this.$cookies.set('dark', 'off')  
  } else {  
    this.$cookies.set('dark', 'on')  
  }  
},
```

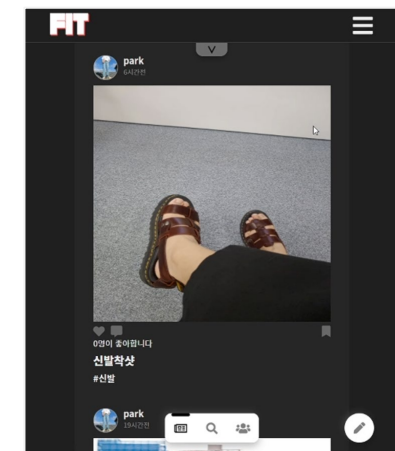
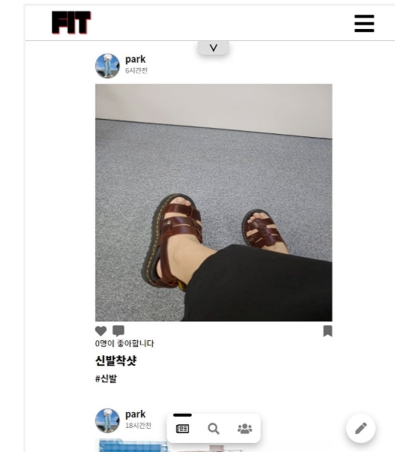
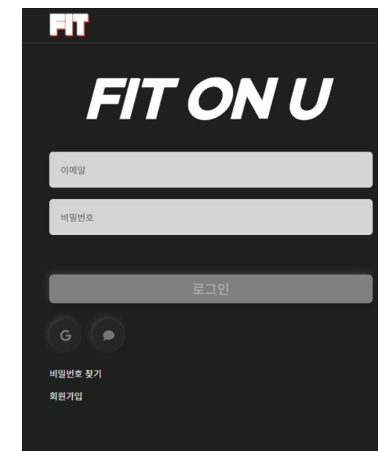
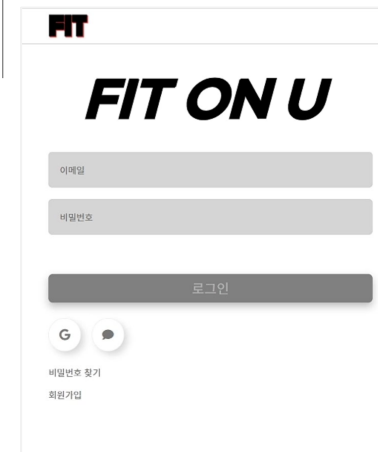
```
defaultDark() {  
  const Dark = this.$cookies.get('dark')  
  const HTML = document.querySelector('html')  
  const wrap = document.querySelector('.wrap')  
  if (Dark === null) {  
    this.$cookies.set('dark', 'on')  
  }  
  if (Dark === 'off') {  
    HTML.classList.add('black')  
    wrap.classList.add('wrap-dark')  
  } else {  
    HTML.classList.remove('black')  
    wrap.classList.remove('wrap-dark')  
  }  
},
```

### [토글 변경하기]

토글 클릭 시  
모드의 변화를 쿠키에 저장

### [클래스 변경]

Watch()에서 토글 변경을  
감지하면  
해당 페이지에서 색상 변경  
이 필요한 DOM에 클래스  
추가/제거



[Light Mode]

[Dark Mode]



## SVG LOGO

SVG를 이용하여 역동적인 로고를 제작

Figma를 이용하여 Stroke만 남긴 텍스트를 SVG로 추출한 후

CSS의 stroke-dasharray를 통해 stroke를 점선으로 만들고 stroke-dashoffset을 keyframe으로 조절하여 역동적인 애니메이션을 구축함



※ 10 프레임 기준

## 체계적 구조

유지보수의 용이를 위해  
디렉토리와 파일 구조를  
체계적으로 구성

```
✓ src
  > assets
  ✓ components
    ✓ css
      # alarm.css
      # community.css
      # directmessage.css
      # dm.css
      # feed.css
      # feedwrite.css
      # findpassword.css
      # findpasswordok.css
      # join.css
      # joinconfirm.css
      # joinprofile.css
      # login.css
      # newpassword.css
      # otheruser.css
      # pagenotfound.css
      # passwordchange.css
      # profileedit.css
      # profileinform.css
      # resellmessage.css
      # search.css
```

```
✓ src
  > assets
  ✓ components
    > css
    ✓ BookMark.vue
    ✓ CommentModal.vue
    ✓ Curation.vue
    ✓ Follower.vue
    ✓ Following.vue
    ✓ HashSearch.vue
    ✓ MapModal.vue
    ✓ MyFeed.vue
    ✓ PostModal.vue
    ✓ SearchModal.vue
    ✓ UserSearch.vue
```

```
✓ router
  JS index.js
✓ utils
  JS timecal.js
✓ vuex
  JS actions.js
  JS getters.js
  JS index.js
  JS mutations.js
  JS store.js
```

```
✓ views
  ✓ Alarm.vue
  ✓ Community.vue
  ✓ DirectMessage.vue
  ✓ DM.vue
  ✓ Feed.vue
  ✓ FeedWrite.vue
  ✓ FindPassword.vue
  ✓ FindPasswordOk.vue
  ✓ Join.vue
  ✓ JoinConfirm.vue
  ✓ JoinConfirmNew.vue
  ✓ Login.vue
  ✓ NewPassword.vue
  ✓ OtherUser.vue
  ✓ PageNotFound.vue
  ✓ PasswordChange.vue
  ✓ ProfileEdit.vue
  ✓ ProfileInform.vue
  ✓ ResellMessage.vue
  ✓ Search.vue
  ✓ Settings.vue
  ✓ SocialJoin.vue
```

시연영상



[https://www.youtube.com/watch?v=\\_wvjTHjn7WA](https://www.youtube.com/watch?v=_wvjTHjn7WA)

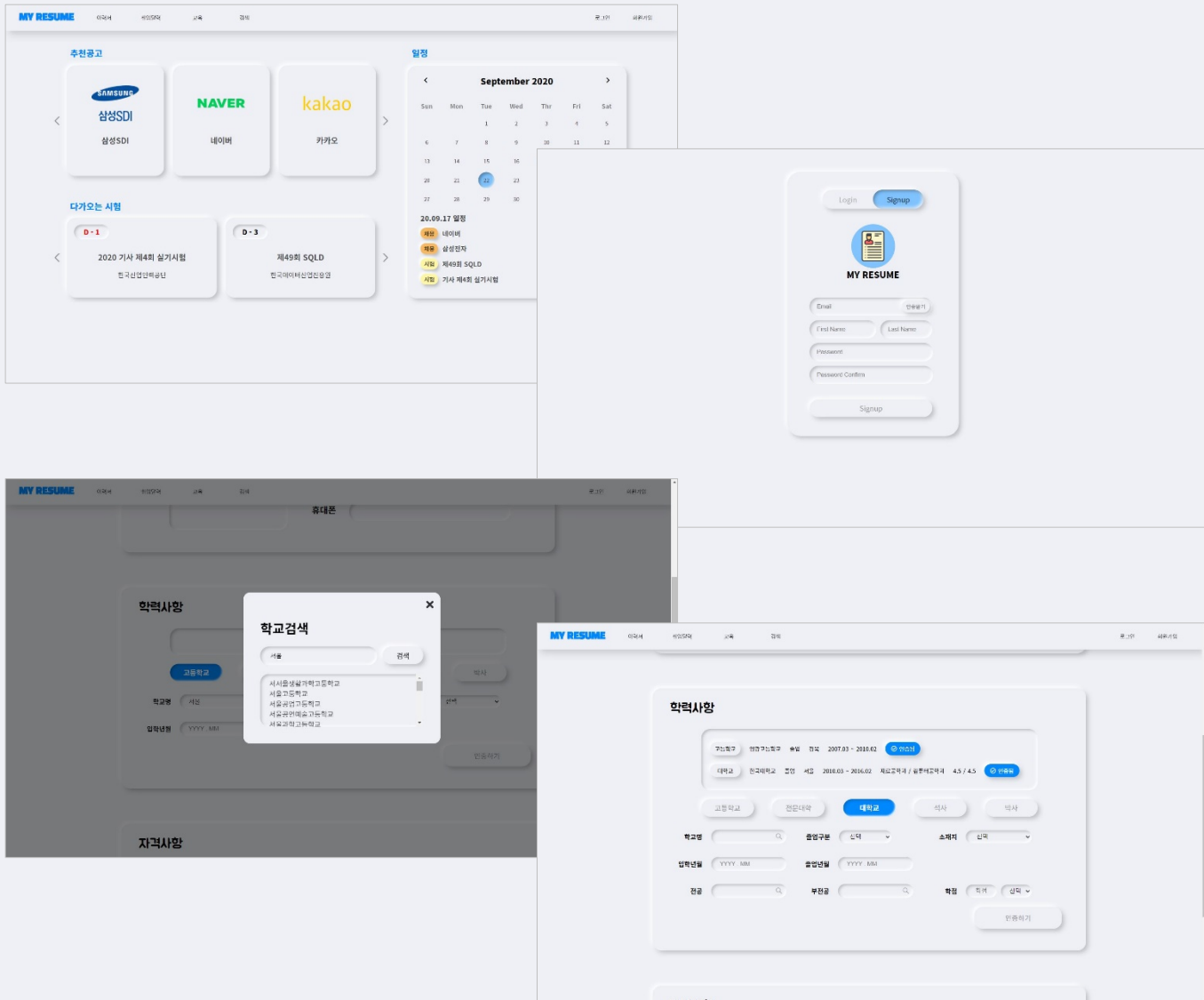
## 블록체인

# 이더리움 이력서 JOB 이로운 생활

작업기간 20.08.24 ~ 20.10.08  
참여인원 5명  
기여도 50%

블록체인 분산원장을 활용한  
투명성 있는 이력서 제공 서비스





## 목표

많은 취업준비생들은 수 많은 회사에 제각기 다른 이력서를 제출한다. 매번 작성하기에는 많은 시간이 할애된다. 따라서 JOB 이로운 생활에서는 이력서 통합 솔루션을 제공하고자 한다. 공통 양식의 이력서를 활용하여 재작성을 방지하고 블록체인 분산원장을 통해 투명성을 높여 학력에 대한 위변조를 막을 수 있다.

또한, 기업 입장에서 자사 내에 별도의 인증시스템을 구축하지 않아도 되기 때문에 비용적으로 이점을 줄 수 있다.

## 담당역할

프론트엔드 (와이어프레임, 레이아웃, 웹 퍼블리싱, 비동기 네트워크 연동)

## 기술스택

HTML, CSS, JavaScript, Vue.js, Ethereum

## 특이사항

Neumorphism 디자인, 오픈 API(학교/학과 정보) 사용, UI 라이브러리(bootstrap, vuetify) 미사용, 1인 프론트엔드

Wireframe

바로가기



## Neumorphism

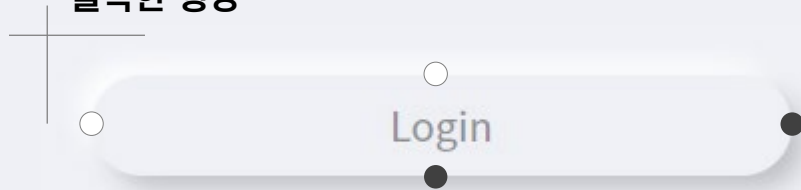
Neumorphism은 그림자와 명암 차로 배경과 UI를 구분하는 디자인 방식이다.

CSS의 box-shadow 옵션을 통해 그림자를 만들어 입체적인 UI를 형성

볼록한 형상은 좌-상에는 밝게, 우-하에는 어둡게

오목한 형상은 좌-상에는 어둡게, 우-하에는 밝게

볼록한 형상

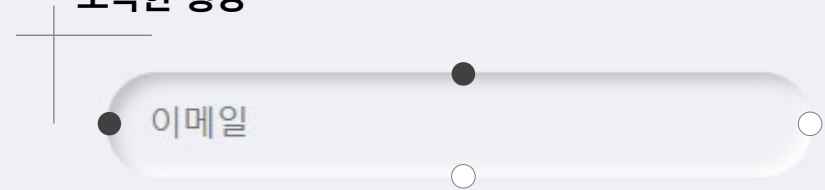


box-shadow:

6px 6px 10px -1px rgba(0,0,0,0.2), ●

-6px -6px 10px -1px rgba(255,255,255,1); ○

오목한 형상



box-shadow:

0 0 0 0 rgba(0,0,0,0), ⊗

0 0 0 0 rgba(0,0,0,0), ⊗

inset 4px 4px 6px -1px rgba(0,0,0,0.2), ●

inset -3px -3px 4px -1px rgba(255,255,255,1); ○

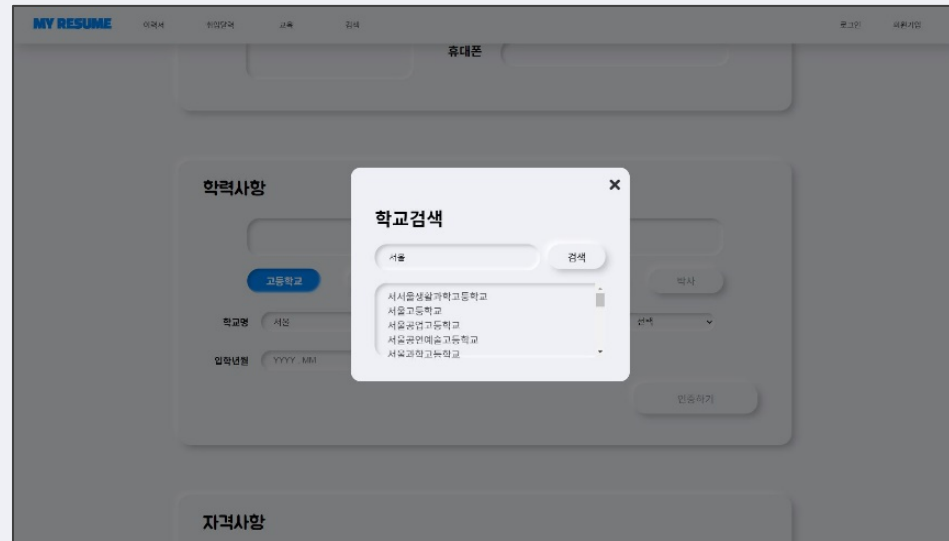
## 오픈 API

학교, 학과 검색의 용이성과 DB 저장의 통일을 위해  
커리어넷([링크](#))의 오픈 API를 사용

```
getSchool(type, name, num) {  
  if (num === 0) {  
    axios.get('https://www.career.go.kr/cnet/openapi/getOpenApi', {  
      params: {  
        apiKey: API_KEY,  
        svcType: 'api',  
        svcCode: 'SCHOOL',  
        gubun: type,  
        searchSchulNm: name,  
        contentType: 'json'  
      })  
    }).then(res => {  
      let ARR = []  
      for (let i=0; i<res.data.dataSearch.content.length ; i++) {  
        ARR.push(res.data.dataSearch.content[i].schoolName)  
      }  
      this.schoolList = Array.from(new Set(ARR))  
    })  
    .catch(() => {})  
  } else {  
    axios.get('https://www.career.go.kr/cnet/openapi/getOpenApi', {  
      params: {  
        apiKey: API_KEY,  
        svcType: 'api',  
        svcCode: 'SCHOOL',  
        gubun: type,  
        searchSchulNm: name,  
        sch1: num,  
        contentType: 'json'  
      })  
    }).then(res => {  
      let ARR = []  
      for (let i=0; i<res.data.dataSearch.content.length ; i++) {  
        ARR.push(res.data.dataSearch.content[i].schoolName)  
      }  
      this.schoolList = Array.from(new Set(ARR))  
    })  
    .catch(() => {})  
  }  
},
```

### [학교정보 받기]

type(고등학교, 대학교)에 따른 구분과 입력 값인 name을 통해 요청을  
보낸 후 받은 응답을 배열에 저장하여 HTML에 리스트업 함



### [검색결과 모달]

응답을 통해 리스트업 한 결과를  
순차대로 보여주며 클릭 시 모달  
종료 후 기존 입력창에 자동완성 됨



## API 정의서

약 50여개의 URL을 비동기 네트워크(axios)를 이용하여 Django 백엔드와 연계를 구축

범주	기능	HTTP Method	URL
Admin	관리자 사이트	GET	/admin
Rest-Auth	회원가입	POST	/rest-auth/signup
	로그인	POST	/rest-auth/login
	로그아웃	POST	/rest-auth/logout
Account	유저정보	POST	/accounts
	이메일 인증	POST	/accounts/<email>
	지갑 생성	POST	/accounts/wallet/<str:password>/
	지갑 조회	GET	/accounts/wallet/<int:id>/list/
	이력서 등록	POST	/accounts/register/<int:id>/
	이력서 구매	POST	/accounts/item/<int:id>/
	이력서 조회	POST	/accounts/item/list/
	영상 구매	POST	/accounts/video/<int:id>/
	구매 내역 조회	POST	/accounts/transaction/list/
Article	이력서 리스트	GET	/articles
	이력서 생성	POST	/articles/create
	이력서 디테일	GET	/articles/<article_pk>
	이력서 수정	PUT	/articles/<article_pk>
	이력서 삭제	DELETE	/articles/<article_pk>
Certificate	자격증 리스트	GET	/articles/<int:article_pk>/certificates/
	자격증 생성	POST	/articles/<int:article_pk>/certificates/create/
	자격증 디테일	GET	/articles/<int:article_pk>/certificates/<int:certificate_pk>/
	자격증 수정	PUT	/articles/<int:article_pk>/certificates/<int:certificate_pk>/
	자격증 삭제	DELETE	/articles/<int:article_pk>/certificates/<int:certificate_pk>/

범주	기능	HTTP Method	URL
Language	어학 리스트	GET	/articles/<int:article_pk>/languages/
	어학 생성	POST	/articles/<int:article_pk>/languages/create/
	어학 디테일	GET	/articles/<int:article_pk>/languages/<int:language_pk>/
	어학 수정	PUT	/articles/<int:article_pk>/languages/<int:language_pk>/
	어학 삭제	DELETE	/articles/<int:article_pk>/languages/<int:language_pk>/
Career	경력 리스트	GET	/articles/<int:article_pk>/careers/
	경력 생성	POST	/articles/<int:article_pk>/careers/create/
	경력 디테일	GET	/articles/<int:article_pk>/careers/<int:career_pk>/
	경력 수정	PUT	/articles/<int:article_pk>/careers/<int:career_pk>/
	경력 삭제	DELETE	/articles/<int:article_pk>/careers/<int:career_pk>/
Selfintroduction	자기소개서 리스트	GET	/articles/<int:article_pk>/selfintroductions/<int:recruitment_pk>/
	자기소개서 생성	POST	/articles/<int:article_pk>/selfintroductions/<int:recruitment_pk>/create/
	자기소개서 디테일	GET	/articles/<int:article_pk>/selfintroductions/<int:recruitment_pk>/<int:selfintroduction_pk>/
	자기소개서 수정	PUT	/articles/<int:article_pk>/selfintroductions/<int:recruitment_pk>/<int:selfintroduction_pk>/
	자기소개서 삭제	DELETE	/articles/<int:article_pk>/selfintroductions/<int:recruitment_pk>/<int:selfintroduction_pk>/
Recruitment	공고 리스트	GET	/recruitments/
	공고 리스트(메인)	GET	/recruitments/showlist/
	공고 리스트(달력)	GET	/recruitments/calendar/<str:date>/
	공고 생성	POST	/recruitments/create/
	공고 디테일	GET	/recruitments/<int:recruitment_pk>/
Introduction	공고 자기소개서 항목 리스트	GET	/recruitments/<int:recruitment_pk>/introductions/
	공고 자기소개서 항목 생성	POST	/recruitments/<int:recruitment_pk>/introductions/create/
	공고 자기소개서 항목 디테일	GET	/recruitments/<int:recruitment_pk>/introductions/<int:introduction_pk>/
	공고 자기소개서 항목 수정	PUT	/recruitments/<int:recruitment_pk>/introductions/<int:introduction_pk>/
	공고 자기소개서 항목 삭제	DELETE	/recruitments/<int:recruitment_pk>/introductions/<int:introduction_pk>/
	공고 지원	GET	/recruitments/<int:recruitment_pk>/apply/<int:article_pk>/

## 체계적 구조

유지보수의 용이를 위해  
디렉토리와 파일 구조를  
체계적으로 구성

```

src
├── assets
├── components
│   └── css
│       ├── applicant.css
│       ├── buymodal.css
│       ├── calendar.css
│       ├── career-modal.css
│       ├── home.css
│       ├── login.css
│       ├── mail-validation-modal.css
│       ├── major-search.css
│       ├── mycoin.css
│       ├── mycontract.css
│       ├── mylist.css
│       ├── mypage.css
│       ├── other-resume.css
│       ├── recruit-home.css
│       ├── recruit-modal.css
│       ├── recruit-write.css
│       ├── resume-edit.css
│       ├── resume.css
│       ├── school-search.css
│       ├── search.css
│       ├── user-modal.css
│       └── video.css

```

```

src
├── assets
├── components
│   └── css
│       ├── BuyModal.vue
│       ├── BuyResume.vue
│       ├── CareerModal.vue
│       ├── MailValidationModal.vue
│       ├── MajorSearch.vue
│       ├── MyCoin.vue
│       ├── MyList.vue
│       ├── MyVideo.vue
│       ├── RecruitModal.vue
│       ├── SchoolSearch.vue
│       ├── SignModal.vue
│       ├── SignResume.vue
│       └── UserModal.vue

```

```

router
├── index.js
└── vuex
    ├── actions.js
    ├── getters.js
    ├── index.js
    ├── mutations.js
    └── store.js

```

```

views
├── Applicant.vue
├── Calendar.vue
├── Home.vue
├── Login.vue
├── MyContract.vue
├── Mypage.vue
├── OtherResume.vue
├── RecruitHome.vue
├── RecruitWrite.vue
├── Resume.vue
├── ResumeEdit.vue
├── Search.vue
└── Video.vue

```

시연영상



<https://www.youtube.com/watch?v=SNnlk2UHmCQ>

3D - 웹

## 코딩 교육 플랫폼 COBIT

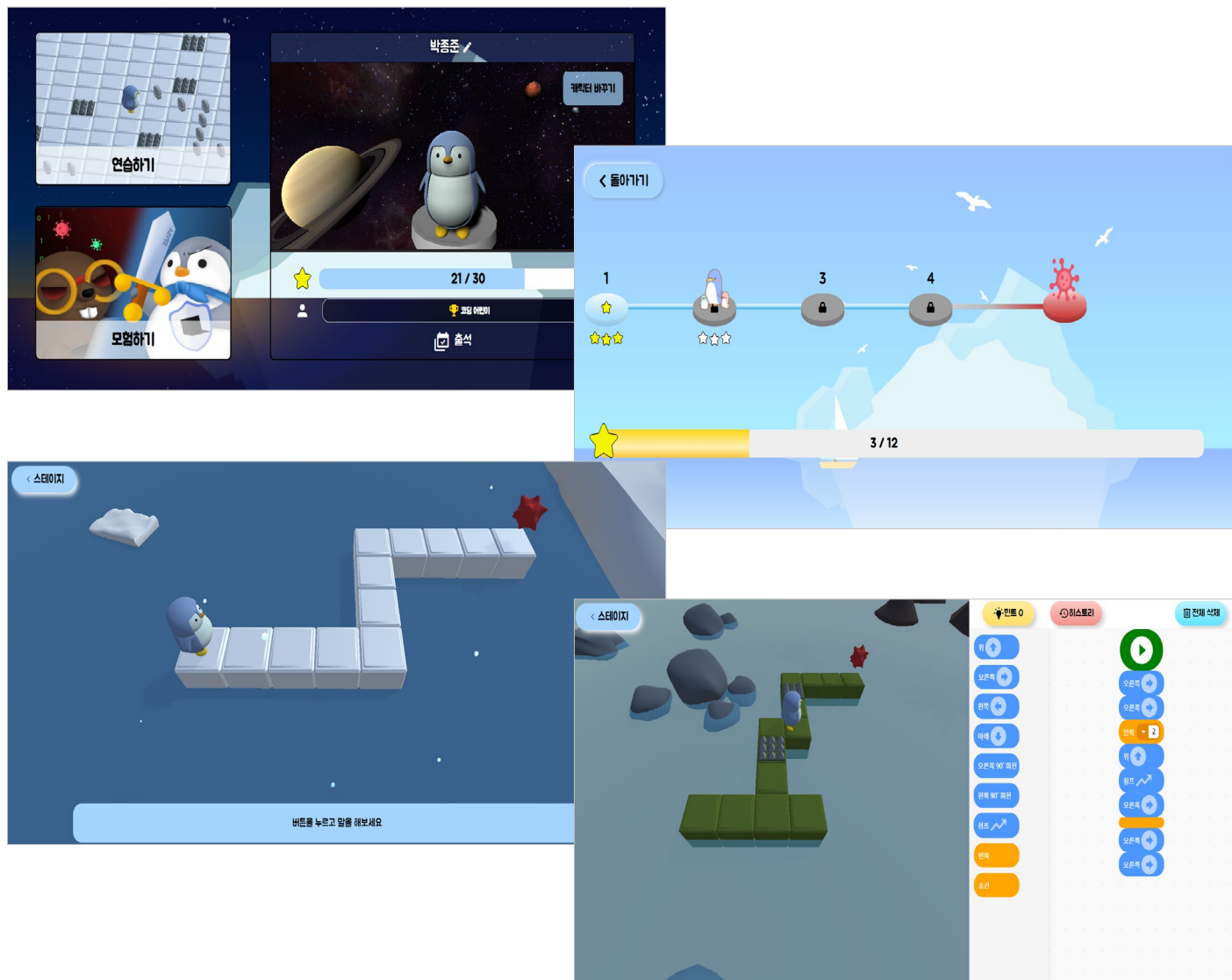
작업기간 20.10.12 ~ 20.11.27  
참여인원 5명  
기여도 20%



삼성 청년 SW 아카데미 3기  
자율프로젝트 전국 1위

블록코딩과 STT(Speech to Text)를  
3D 게임에 접목  
코딩 교육의 진입장벽을 낮춘 웹 플랫폼





## 목표

SW교육의 의무화로 코딩 조기교육의 필요성은 증가했다. 기존 서비스인 Scratch는 어린 학생들이 이용하기에는 몰입도가 떨어지고 진입장벽이 높은 편이다. 따라서 COBIT은 3D 게임을 도입하여 흥미를 유발하고 튜토리얼을 제공하여 진입장벽을 낮춘 블록코딩 교육 플랫폼이다. 또한, STT(Speech to Text)를 이용해 자연어를 코드화 하는 시스템을 도입하여 개발자의 사고를 향상시킬 수 있다.

## 담당역할

프론트엔드 (웹 퍼블리싱, 3D 모델 제어, 유니티-웹 연동)

## 기술스택

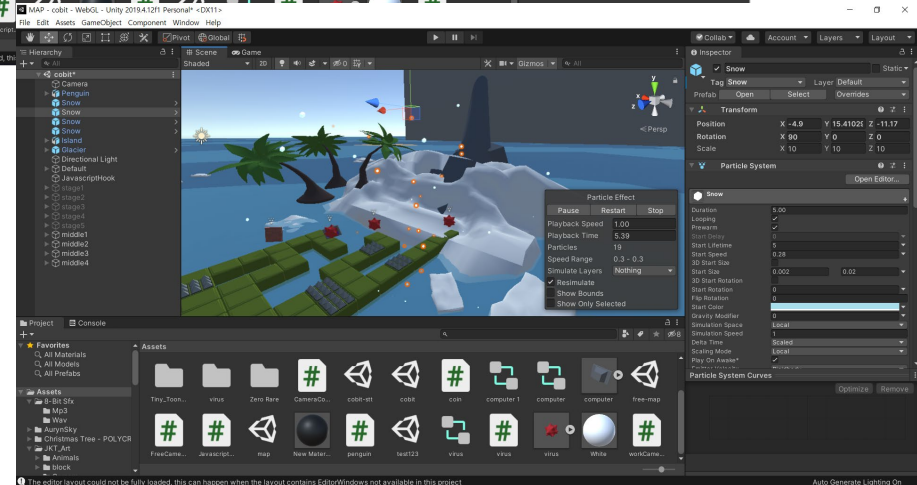
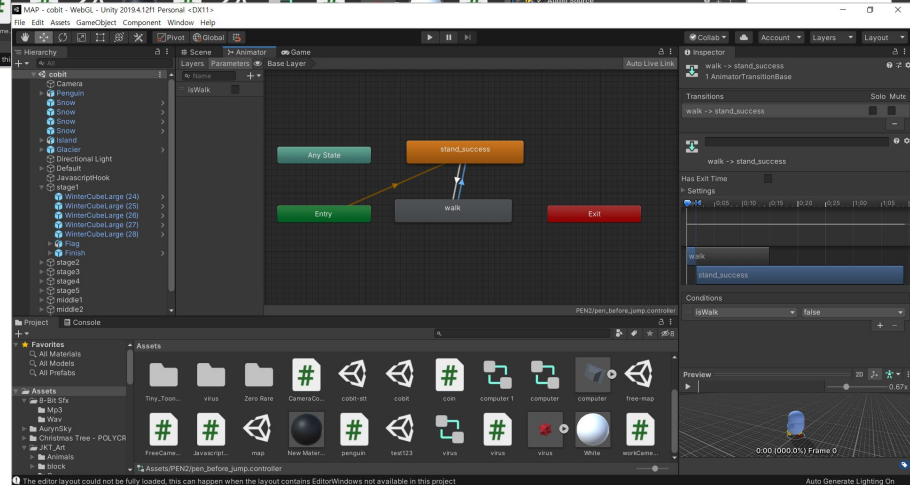
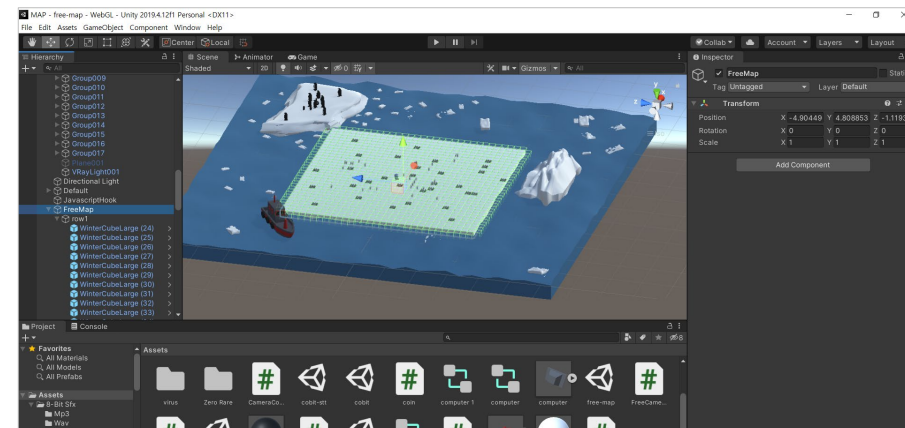
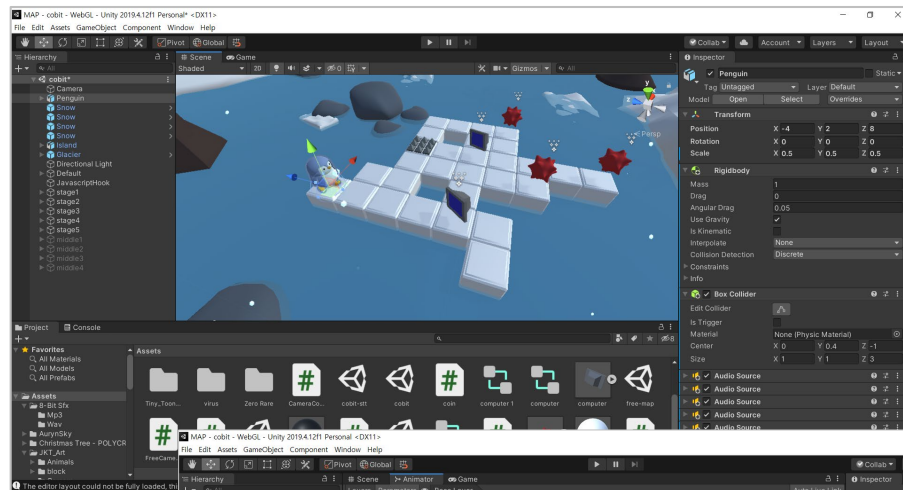
HTML, CSS, JavaScript, Vue.js, Unity, C#, WebGL

## 특이사항

처음 다루는 언어와 도구인 Unity, C#, WebGL을 프로젝트를 진행하면서 학습함

## Unity

3D 게임 제작 도구인 Unity를 통해  
캐릭터의 위치 및 애니메이션, 맵 제작, 파티클, 충돌, 사운드, 중력 등과 같은  
효과 및 구성을 제작했다.



## C# Script

Unity의 오브젝트에 C# Script를 연결하여  
변수와 함수를 통해 캐릭터의 이동/충돌 제어, 이펙트 변화, 맵 이동/변화 제어 등과 같은  
오브젝트 컨트롤을 설정했다.

### [사용된 변수들]

```
[SerializeField] private GameObject cam;
[SerializeField] private GameObject penguin;
[SerializeField] private GameObject[] finish;
[SerializeField] private GameObject[] snow;
[SerializeField] private GameObject glacier;
[SerializeField] private GameObject island;
[SerializeField] private GameObject stage0;
[SerializeField] private GameObject stage1;
[SerializeField] private GameObject stage2;
[SerializeField] private GameObject stage3;
[SerializeField] private GameObject stage4;
[SerializeField] private GameObject stage5;
[SerializeField] private GameObject middle1;
[SerializeField] private GameObject middle2;
[SerializeField] private GameObject middle3;
[SerializeField] private GameObject middle4;
[SerializeField] private GameObject Virus;
bool[] isStage = { false, false, false, false, false, false };
bool[] isMiddleStage = { false, false, false, false, false };
float[] cameraRotate = { 55.0f, 90.0f, 0 };
float[] cameraX = { 0, -11.8f, -11.81f, -11.36f, -8.9f, -11.32f };
float[] cameraY = { 0, 11.0f, 11.98f, 12.11f, 11.77f, 12.45f };
float[] cameraZ = { 0, 4.0f, 4.56f, 2.0f, 5.07f, 3.29f };
float[] middleCameraX = { 0, -7.32f, -8.6f, -12.0f, -8.85f };
float[] middleCameraY = { 0, 10f, 10f, 13.0f, 12.36f };
float[] middleCameraZ = { 0, 2.84f, 4.1f, -5.0f, -0.7f };
Vector3 target = new Vector3(0, 0, 0);
float target2 = 0.0f;
ArrayList myArr = new ArrayList();
float moveSpeed = 0.05f;
int i = 0;
bool isSet = true;
bool isSet2 = true;
bool isStart = false;
bool isClear = false;
bool isWhile = false;
bool isStop1 = false;
bool isStop2 = false;
bool isTrap = false;
bool isIfRun = false;
bool isIfRun2 = false;
bool isEnd = false;
public int ifCount;
public int coinCount = 0;
public int heartCount = 5;
string fileName;
string[] myIf;
```

### [캐릭터 이동 제어 함수]

```
참조 2개
public void UpMove()
{
    isSet = false;
    moveSpeed = 0.05f;
    Animator animator = penguin.GetComponent<Animator>();
    AudioSource[] audio = penguin.GetComponent<AudioSource>();
    audio[1].Play();
    animator.SetBool("isWalk", true);
    float xPosition = penguin.transform.localPosition.x;
    float yPosition = penguin.transform.localPosition.y;
    float zPosition = penguin.transform.localPosition.z;
    xPosition = xPosition + 2f;
    target = new Vector3(xPosition, yPosition, zPosition);
    penguin.transform.localEulerAngles = new Vector3(0, 270, 0);
}
```

### [캐릭터 충돌 제어 함수]

```
void OnTriggerEnter(Collider collider)
{
    if (collider.gameObject.name == "Glacier" || collider.gameObject.name == "Island")
    {
        Invoke("GameOver", .5f);
    }
    if (collider.gameObject.name == "Trap")
    {
        if (SceneManager.GetActiveScene().name == "free-map")
        {
            if (collider is CapsuleCollider)
            {
                Invoke("HeartCount", .5f);
            }
            if (collider is BoxCollider)
            {
                GameObject.Find("JavaScriptHook").SendMessage("SetTrap");
            }
        }
        else
        {
            if (collider is CapsuleCollider)
            {
                Invoke("GameOver", .5f);
            }
            if (collider is BoxCollider)
            {
                GameObject.Find("JavaScriptHook").SendMessage("SetTrap");
            }
        }
    }
}
```

### [맵 변화 제어]

```
참조 0개
public void Stage(int num)
{
    for (int k = 0; k < isStage.Length; k++)
    {
        if (k == num)
        {
            isStage[k] = true;
            cam.transform.localPosition = new Vector3(cameraX[k], cameraY[k], cameraZ[k]);
            penguin.transform.localPosition = new Vector3(-4.0f, 2f, 8.0f);
            penguin.transform.localEulerAngles = new Vector3(0, 0, 0);
        }
        else
        {
            isStage[k] = false;
        }
    }
    if (fileName == "cobit")
    {
        island.SetActive(false);
        middle1.SetActive(false); middle2.SetActive(false); middle3.SetActive(false); middle4.SetActive(false);
    }
    stage1.SetActive(isStage[1]); stage2.SetActive(isStage[2]);
    stage3.SetActive(isStage[3]); stage4.SetActive(isStage[4]); stage5.SetActive(isStage[5]);
}
```

### [게임 제어 함수]

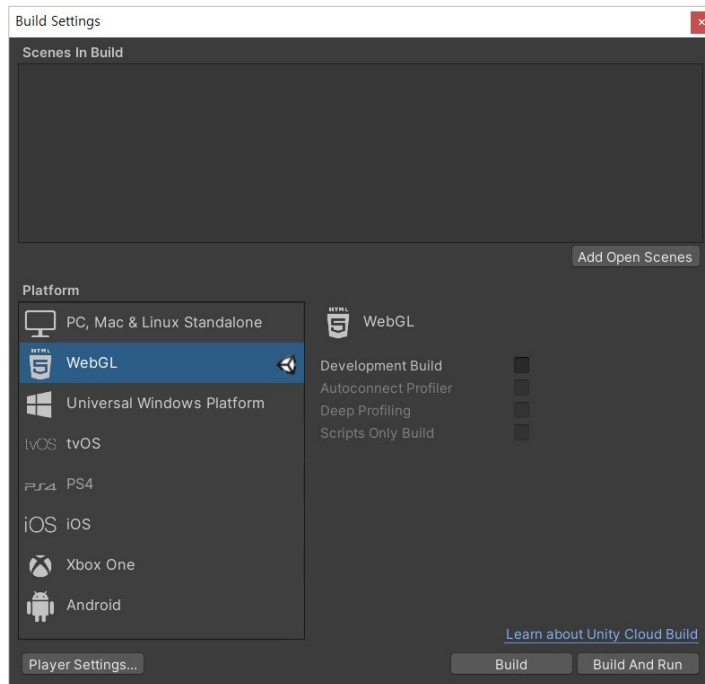
```
참조 0개
public void RestartGame()
{
    Vector3 target = Vector3.zero;
    SceneManager.LoadScene(fileName);
}
참조 2개
public void GameOver()
{
    AudioSource[] audio = penguin.GetComponent<AudioSource>();
    Rigidbody rigidbody = penguin.GetComponent<Rigidbody>();
    if (!isEnd)
    {
        audio[4].Play();
        rigidbody.AddForce(Vector3.up*4, ForceMode.Impulse);
        isEnd = true;
        customEvent2();
    }
}
참조 0개
public void EndGame()
{
    isClear = true;
    finish = GameObject.FindGameObjectsWithTag("Finish");
    for (int y = 0; y < finish.Length; y++)
    {
        ParticleSystem particle = finish[y].GetComponent<ParticleSystem>();
        particle.Play();
    }
    AudioSource[] audio = penguin.GetComponent<AudioSource>();
    audio[3].Play();
    Invoke("CallEvent", 1.2f);
}
```



## WebGL

Unity의 3D 그래픽을 웹에서 사용하기 위해 JavaScript API인 WebGL을 사용했다.  
Vue.js의 라이브러리인 vue-unity-webgl을 사용하여 Vue.js에서 WebGL로 빌드한 Unity와 연동을 했다.

### [webGL Build]



### [Vue.js에서 Unity 로드]

```
<unity src="glacier/Build/glacier.json"
      | unityLoader="glacier/Build/UnityLoader.js"
      | ref="myInstance">
</unity>
```

### [Vue.js에서 Unity의 Script 함수 호출]

```
this.$refs.myInstance.message('JavascriptHook', "Go");
```



## jslib

jslib는 Unity 플러그인으로 WebGL로 빌드한 경우 C# Script로 jslib에 존재하는 JavaScript 함수를 호출 할 수 있다.  
Vue.js에서 Unity로 호출은 가능하나 반대로 Unity에서 Vue.js에서 호출이 불가능 하여 jslib를 이용하여 이러한 문제를 해결했다.

과정은 다음과 같다.

1. Unity의 C# Script에서 jslib의 JS 함수를 호출한다.
2. 해당 함수는 dispatchEvent 함수로 특정 Event를 생성한다.
3. Vue.js에서 addEventListener 함수로 Event를 감지하여 Unity - jslib - Vue.js의 호출을 완성한다.

### [1. C#에서 jslib 호출]

```
[DllImport("__Internal")]
참조 1개
private static extern void startEvent();
참조 0개
void Start()
{
    startEvent();
}
```

### [2. jslib 함수의 Event 생성]

```
mergeInto(LibraryManager.library, {
  startEvent: function() {
    window.dispatchEvent(new CustomEvent("start", {
      detail: 'on'
    }));
  },
},
```

### [3. Vue.js의 Event 감지]

```
created() {
  window.addEventListener('start', this.handleStart)
},

handleStart() {
  console.log(event)
  setTimeout(() => {
    this.LevelLoad();
    this.loadMyCharacter();
  }, 10);
},
```



### [감지된 Event]

▶ CustomEvent {isTrusted: false, detail: "on", type: "start", target: Window, currentTarget: Window, ...}

[CodeBlock.vue?881c:1287](#)

시연영상



<https://www.youtube.com/watch?v=dia5Z5yaEAk>



# 감사합니다

박종준

010-5002-1524

poiufgin7373@naver.com