Portfolio. -



박서연

| 프로필



박서연 PARK SEOYEON

#INFJ

#배려

#책임감

Tel. 010.7924.4520 E-mail. syys10280@naver.com

합학력사항

우석대학교 | 정보보안학과 입학 2020. 03 - 2022. 02 군산대학교 | 컴퓨터정보공학과 편입학 2022. 03 - 2025. 02

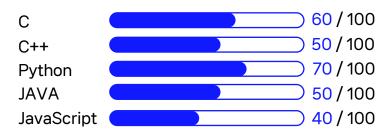
🦥 활동내용/수상이력

학부 연구생 | 군산대학교 ubit 연구실 2023. 01 교내 테크노마켓 은상 | 군산대학교 2023. 11

▲ 자격사항/어학

정보처리기사 | 한국산업인력공단 2024.06

쏫 스킬





가개발기록

Github https://github.com/syys1028/Portfolio

Velog https://velog.io/@tjdus1028

| 프로젝트 목록

01	술 취한 사람 길 찾기 게임 프로젝트	C / Visual Studio
02	퀸-맥클러스키 구현 프로그램	C++ / Visual Studio
03	RSA 구현 프로그램	Python / PyCharm
04	사용자 맞춤형 광고 추천 프로젝트	Python / Spyder
05	멀티모달 장면 검색 시스템 (팀 프로젝트)	Python / VSCode
06	교통 장애 경고 내비게이션 (팀 프로젝트)	JAVA / Android Studio

01. 개인 프로젝트-C/C++

2022, 11, 16 ~ 2022, 12, 01

물 취한 사람 길 찾기 게임 프로젝트

프로젝트 소개

술 취한 사람이 집에 무사히 도착하도록 자동/수동으로 플레이하는 게임입니다. 맵은 사분면으로 나뉘고 분면마다 이벤트가 발생합니다. 자동 모드에서는 술에 취한 것처럼 랜덤으로 경로를 이동하다가, 횟수를 초과하면 저격수가 등장하게 됩니다. 경찰에게 발각되면 지정된 횟수 안으로 수동 이동 모드를 통해 집까지 이동해야 합니다.

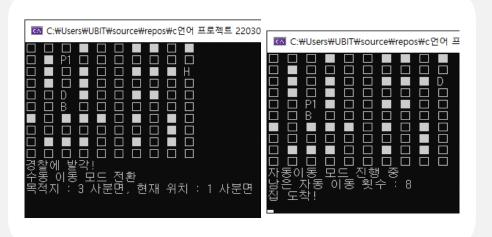
💡 핵심 구현 내용

- 1. 맵 생성 및 출력 기능 (벽, 길, 집, 술집 등 랜덤 생성)
- 2. 자동 이동 모드에서 술 취한 사람의 랜덤 경로 탐색
- 3. 경찰 등장 및 수동 이동 모드 전환
- 4. 분면 이동 성공 시 자동 모드 전환되고 실패 시 경찰 수 증가 및 저격수 등장
- 5. 집 도착 시 승리, 경찰 혹은 저격수에 시야에 들어오면 패배

🏋 기술 스택

C언어 / Visual Studio 2019

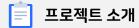
```
* chic 三年世年 2203098 町村田 (新世) - Microsoft Vacal Studio
작원(f) 면접(1) 보기(V) 프루젝트(P) 원드(B) 디너그(D) 힘(M) 도구(T) 테스트(S) 분석(A) 참(W) 도용당(H)
              // 경찰1 생성
               while (flag != 0) {
                                          // 경찰1 성성하기
                                               // 경찰 렌덤 x좌표 생성
// 경찰 렌텀 y좌표 생성
                  police1_x = rand() % 10:
                  policel_v = rand() % 10;
                  Pl_num = check_quadrant(policel_x, policel_y);
                  if (check_quadrant(drunk_man_x, drunk_man_y) == P1_num) [
                     result = map[policel_x][policel_y]: // 집인지 백인지 반별한 결과를 result에 저장
                      if (result - 0) {
                         flag = 0;
   139
   141
   142
               // trap8 : 1번 실패 후 경찰 만난 때까지 반목
               D_num = check_ousdrant(drunk_man_x, drunk_man_y):
               while (trap3 != 0) {
                 system("cls");
                  random D.step_autoO; // 경찰에 발각되기 전까지 자동이동
                  random_P_step_auto(&policel_x, &policel_y, Pl_num):
                  Sleep(speed);
                   if (check_P(policel_x, policel_y) == 1) [
                     printf("경찰에 발각!Mn");
                      printf("수동 이동 모드 전환 Wh"
```



01. 개인 프로젝트-C/C++

2023. 05. 26 ~ 2023. 06. 09

···· 퀸-맥클러스키 구현 프로그램



퀸-맥클러스키 알고리즘은 복잡한 논리식을 간소화 하는 방법 중 하나로, 해당 알고리즘을 구현한 프로젝트 입니다.

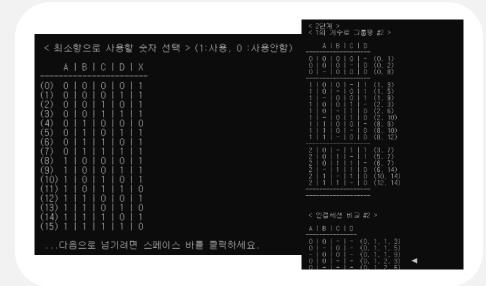
이 프로그램은 각 단계를 거쳐 불필요한 부분을 제거해 최적의 논리식을 찾아 회로를 간단하게 만들어 주는 기능을 합니다.

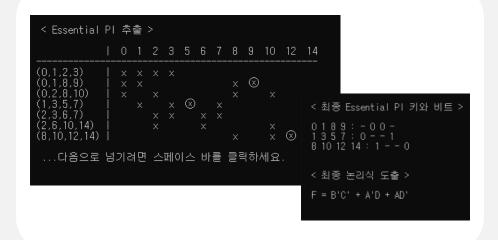
💡 핵심 구현 내용

- 1. 사용자의 입력을 받아 최소항 테이블 구성
- 2. 이진수로 변환 후, 1의 개수로 그룹화
- 3. 첫 번째 인접 그룹 비교 후 최소항 후보(PI) 결정
- 4. 필요 시, 두 번째 인접 그룹 비교 후 최소항 후보(PI) 결정
- 5. 선택 표 도식화 및 Prime Implicant(PI) 도출
- 6. 필수 최소항(EPI) 도출 및 논리식 도출

父 기술 스택

C++ / Visual Studio 2019

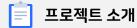




01. 개인 프로젝트-Python

2021. 10. 15 ~ 2021. 10. 30 (update 2024.09)

" RSA 암호화 알고리즘 구현 프로그램



RSA 암호화 알고리즘을 구현한 것으로, 큰 소수의 곱을 기반으로 하여 보안성이 높은 암호화 알고리즘 입니다.

Python을 사용하였으며, 실행 시 자동으로 512비트 길이의 RSA 키를 생성하고, 사용자의 공개키와 비밀키를 생성해 메시지를 암호화 및 복호화하는 기능을 구현했습니다.

💡 핵심 구현 내용

- 1. 모듈러 거듭제곱 연산
- 2. 유클리드 및 확장 유클리드 알고리즘
- 3. 모듈러 역원 계산
- 4. 소수 판별 및 소수 생성 (밀러-라빈 알고리즘)
- 5. RSA 키 생성 및 메시지 암호화 및 복호화

❤ 기술 스택

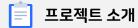
Python / PyCharm

```
class RSA:
                                                                                         if n <= 1:
                                                                                                              # n이 소수일지 검사
                                                                                           return False
   def __init__(self, bits=1024): # RSA 커플 조기하
     self.e = 65537 # <u>일반적으로 사용되는</u> 공세 지수
                                                                                         if n <= 3:
      self.generate_keys()
                                                                                         if n % 2 == 0:
                                                                                         r, d = 0, n - 1
                                                                                                              # n - 1 = 2^r + d 전쟁로 변환
p = generate_prime(self.bits)
                                                                                         while d % 2 -- 8:
                                                                                            d //- 2
          q = generate_prime(self.bits)
                                                                                            a = random.randrange(2, n - 1)
      phi = (p - 1) * (q - 1)
                                                                                            x = modular_exponentiation(a, d, n)
       if gcd(self.e, phi) != 1:
          raise ValueError("e의 g(n)은 서로소의야 합니다.")
                                                                                             for _ in range(r - 1):
      self.d = modular_inverse(self.e, phi)
                                                                                               x = modular_exponentiation(x, 2, n)
          raise ValueEprop("d의 모듈관 역원을 계산할 수 없습니다.")
                                                                                 is_prime() > while d % 2 == 0
6 usages - A syys1028
def modular_exponentiation(base, exponent, modulus): # 커뮤제곱 제반
                                                                                     def gcd(a, b): # 유클리드 알고리즘으로 최대공약수를 계산
                                                                                           a, b = b, a % b
                                                                                     def extended_gcd(a, b): # 확장한 유플라드 말고리용
          result - (result * base) % modulus
                                                                                     \# ax + by = gcd(a, b)
       evenment = evenment // 2
```

01. 개인 프로젝트-Python

2022. 06. 08 ~ 2022. 06. 22 (update 2024.09)

··· 사용자 맞춤형 광고 추천 프로젝트



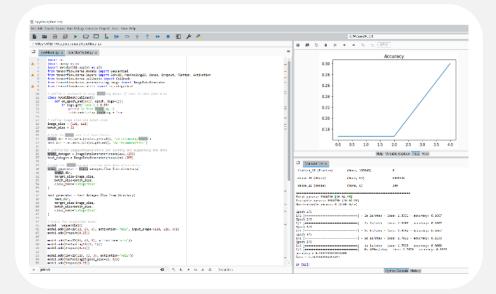
이미지 분류 모델을 통해 사용자의 연령, 성별, 관심사에 따라 최적 화된 상품을 추천합니다. TensorFlow를 사용해 이미지 데이터를 학습시키고, 학습된 모델을 기반으로 사용자가 관심 있는 제품을 분 석한 후, 추천 결과를 제공합니다. 이 프로젝트는 광고 개인화와 추 천 알고리즘을 통해 사용자 경험을 극대화하는 데 목적을 둡니다.

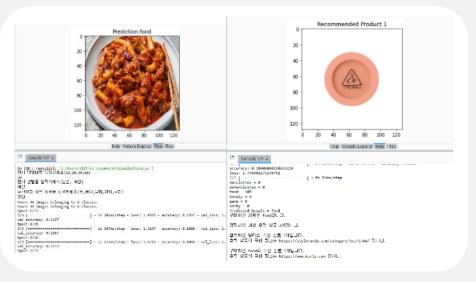
♀ 핵심 구현 내용

- 1. 데이터 수집 및 전처리
- 2. 이미지 분류 모델 학습 후 성능 평가 및 모델 저장
- 3. 사용자 정보 입력 및 구매 품목 분류
- 4. 실시간 이미지 분석 및 추천
- 5. 추천 상품 시각화 및 링크 제공

🏋 기술 스택

Python / Spyder (anaconda3) / TensorFlow, CNN

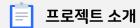




02. 팀 프로젝트-Python

2024. 08. 23 ~ 2024. 08. 25 (update 2024.09)

멀티모달 장면 검색 시스템



동영상에서 자막, 음성, 장소, 인물 정보를 추출하고, 이를 데이터베이스에 저장하여 사용자가 원하는 장면을 쉽게 검색하고 재생할 수있는 시스템입니다. 키워드를 입력하면 일치하는 장면의 타임스탬프를 제공하고, 그 장면으로 바로 이동할 수 있습니다.

- 💡 핵심 구현 내용
- 1. 웹과 데이터베이스 연동 서버 구축
- 2. 자막, 음성, 장소 인식을 위한 Python 모듈 개발
- 3. 영상에서 추출된 데이터를 바탕으로 장면 검색 및 영화 추천
- 4. 영상 데이터 번역 및 파일 다운로드 기능 제공
- 🚨 인원 구성

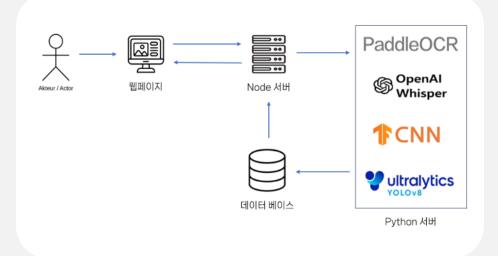
4명 (프론트엔드 1명, 백엔드 및 모듈 개발 3명)

💢 기술 스택

장면 분석: Python / Jupyter Notebook, VSCode / MySQL / PaddleOCR, Whisper, Place365-resnet50, Google Cloud API (Translate)

웹 사이트 : Node.js, Express, PHP





"멀티모달 장면 검색 시스템" 기능 소개

자막/음성/장소 인식

• 자막 인식 - PaddleOCR

252.pmg ['세건의 잔실을 가져가 버런기 전까지는'] 세리의 관심을 가져가 버리기

Tesseract, EasyOCR 등 기존의 라이브러리에 비해 한글 인식에 탁월한 정확도를 보임

• 음성 인식 - Whisper 모델

Start Time: 224.2, End Time: 229.85 Sentence: But I still can't get the hoop slime 16. That's a shame. So Welanie wa

Start Tine: 229,86, End Time: 236,38

Sentence: But of course her nother said no so she three a fit and swashed her Jo-Start Tine: 238.26, End Time: 243.88

Sentence: And she got grounded so now she can't come to my pajanas sieepover par

Sentence: Hey, what are you doing for your science project this year? That is un

Start Time: 250,41, End Time: 256,95 Sentence: Einbinder took an interest in sherry himself. I was thinking of showin

다양한 언어가 지원되며, 높은 정확도와 빠른 처리 속도로 일상 대화 뿐만 아니라 노래, 효과음 등 인식 가능.

• 장소 인식 - CNN Resnet50





Place365에서 제공하는 데이터셋을 사용해서 장소를 인식하는 CNN 모델을 생성

0.303 -> park

데이터베이스

ıme	timeStamp	subText	voiceText	placeText	isMainActor	actorName	actorGender
naid's tale	00:27:18	아버지 일은 유감이 야	(NULL)	restaurant / bar	Υ	케이틀린	women
naid's tale	00:27:20	그래도 자주 볼 수 있겠네 그	But um, I hope I get to see	road_transportation / bridge	Υ	케이틀린	women
naid's tale	00:27:21	(NULL)	(NULL)	road_transportation / bridge	Y	케이틀린	women
naid's tale	00:27:23	물론이죠	(NULL)	road_transportation / bridge	Υ	케이틀린	women
naid's tale	00:27:24	(NULL)	(NULL)	road_transportation / bridge	Υ	케이틀린	women
naid's tale	00:27:26	(NULL)	Okay, wasn't awkward at a	restaurant / restaurant	N	(NULL)	(NULL)
naid's tale	00:27:27	좋아 전혀 어색하지 않았지?	(NULL)	restaurant / restaurant	Υ	제리	man
naid's tale	00:27:30	(NULL)	(NULL)	restaurant / restaurant	Υ	케이틀린	women
		******	- 1 2			******	· · · · · · · ·

업로드 목록 및 정보



A mermaid's tale

- 3.21- United States
- 개봉 납시: 2017년 총 성영 시간: 1시간 32분

명화 내용은 연예의 인간이 만나는 미야기로, 연예의 인간이 함께 공란할 수 있는 방법은 찾는 과정은 다릅니다. 주민공들은 서로른 이제하고 한구가 되려고 노력하 때, 민이와 인간 간의 교류와 이래를 통해 새로운 관계를 향심하려고 합니다. 단미 의 연결한 근실과 연간의 이해와 부장이 역시면서 전개되는 이야기입니다.

출연진









검색 및 재생



(2017) A mermaid's tale (2017) A mermaid's tale - 00:01:12 Subtitle: ERMAIDS TALE 인이공수 이야기

인어나오는 장면

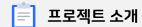
Search

(2017) A mermaid's tale - 00:01:15

02. 팀 프로젝트-JAVA

2023. 08. 25 ~ 2023. 08. 27

··· 교통 장애 경고 내비게이션



열화상/광각 카메라로 도로의 다양한 위험 요소(공사 현장, 사고, 장애물, 낙석, 포트홀, 블랙아이스)를 실시간으로 감지하여 DB에 저장하고, 이를 활용해 지도 앱에서 경고해주는 시스템입니다. 사용자의 위치를 기반으로 해당 위험 요소들이 지도에 표시되고, 70미터 이내로 접근 시 음성 경고를 제공합니다.

💡 핵심 구현 내용

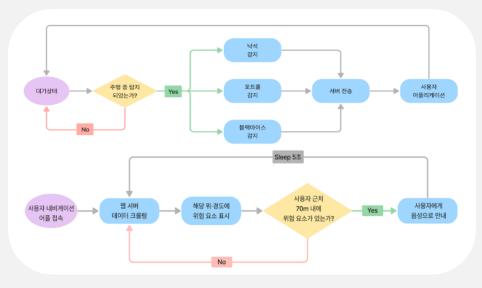
- 1. DB에서 실시간 정보 수집 및 JSON 파싱
- 2. Kakao Map API를 활용한 위험 요소 지도 시각화
- 3. 위험 요소에 따라 맞춤형 아이콘을 지도에 표시
- 4. 용자 위치 기반 70m 반경 접근 시 음성 경고 제공
- 💴 인원 구성

4명 (AI 모듈 개발 2명, 안드로이드 앱 개발 2명)

🏋 기술 스택

Java / Android Studio / MariaDB, phpMyAdmin / Kakao Map API / Text-to-Speech (TTS)





"교통 장애 경고 내비게이션 " 기능 소개

위험 요소 시각화

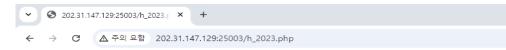


실시간 위치 및 음성 경고



데이터베이스 및 JSON 파싱

도로명	방향	돌발구분	돌발내용	X	Υ	돌발종료일시
순천완주선	순천 방향	공사	(2차로)교량점검작업중	127.3488	35.3448	-
중앙선	부산 방향	공사	(2차로)시설물교체작업증	127.9226	37.3312	2023-08-17 15:26
동해선(삼척-속초)	삼척 방향	공사	(2차로)조인트보수작업증	129.0919	37.5022	-
동해선(삼척-속초	삼척 방향	공사	(1차로)조인트보수작업증	129.0919	37.5022	-
중앙선	춘천 방향	공사	로)중앙분리대개량공사작업	128.5848	36.3398	2023-08-17 17:28
부산외곽선	창원 방향	공사	(갓길)노면보수작업종료	128.975	35.2702	2023-08-17 10:32
+1 LT114	1 11 11141	7.11	/기기가 하네기가나시죠	1070440	204500	



("blackice":[],"open_api":[("datetime":"2023-11-29 00:00:00","latitude":"37.5935","longitude":"126.7686","type":" 00:00:00","latitude":"34.86164","longitude":"127.58408","type":"공사","detail_type":"시설몰보수작업","message": ("datetime":"2023-11-29 00:00:00","latitude":"36.33469","longitude":"126.63914","type":"공사","detail_type":"시설몰보수작업","message":"],("datetime":"2023-11-29 00:00:00","latitude":"38.44042","longitude":"128.44401","type":"공사',"detail_type":"시설몰보수작업","message":""},("datetime":"2023-11-29 00:00:00","latitude":"37.62462","longi 00:00:00","latitude":"37.62462","longi 00:00:00","latitude":"37.612462","longi 10:00:00:00","latitude":"37.62462","longi 10:00:00:00","latitude":"37.62462","longi 10:00:00:00","latitude":"37.62462","longi 10:00:00:00","latitude":"38.06756","longitude":"128.64178","type":"공사","detail_type":"시설물보수작업","message":""},("datetime":"2023-11-29 00:00:00","latitude":"37.60658","longi 10:00:00","latitude":"37.21503","longitude":"37.21503","longitude":"37.21503","longitude":"37.21503","longitude":"37.21503","longitude":"37.21503","longitude":"37.21503","longitude":"35.08009","longitude":"126.95389","type":"3사", "detail_type":"", "detail_type":", "detail_type":

