

# Oriented AI Development Programmer

“안녕하세요. 정장영입니다.”

Email : [akillness38@gmail.com](mailto:akillness38@gmail.com)

LinkedIn : [www.linkedin.com/in/akillness38](https://www.linkedin.com/in/akillness38)

Github : <https://github.com/akillness>

# 약력

- 학력 ----

- 2014.09 - 2016.08 홍익대학교 게임학과 석사학위
  - 2007.03 - 2014.02 홍익대학교 게임소프트웨어공학 학사학위

- 경력 ----

- 2016.09~2018.02 Com2us 게임제작본부 재직
  - 2018.03~2024.06 NCSOFT AI Tech센터 재직

- 수상 ----

- 2016.01.28 “연애센서” HCI APP MARKET 장려상 수상

- 특허 ----

- 2019.08 - 2020.12 게임 나이도 결정 방법 및 장치 | 특허번호 : 10-2195-4710000
  - 2017.11 - 2017.12 감성 컴퓨팅 장치 및 그 동장 방법 | 특허번호 : 10-2017-0009076

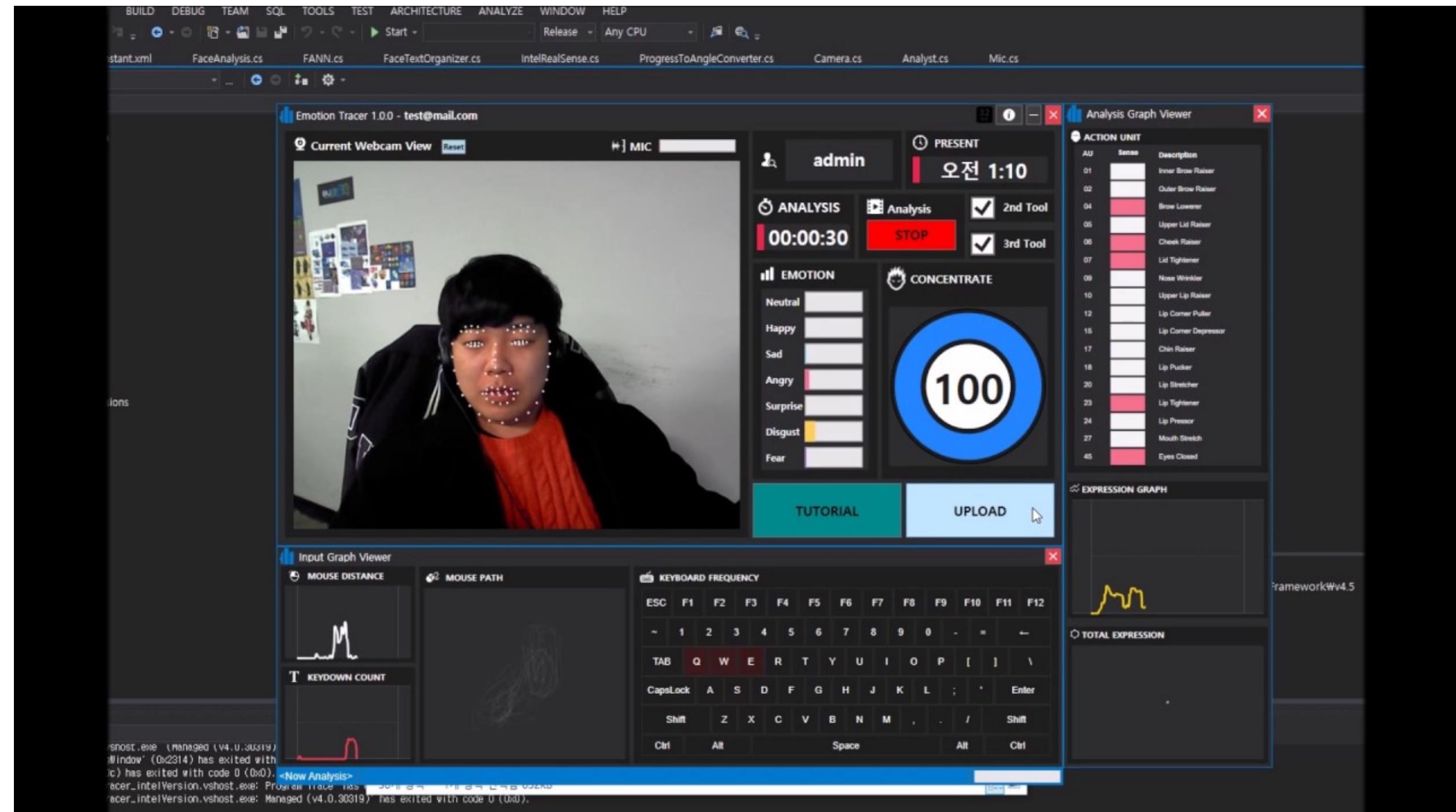
# 안녕하세요. 저는,

- DirectX를 이용해 윈도우 프로그래밍을 접하면서, 게임제작에 필요한 프로그래밍을 시작했습니다. Server-client 구조의 3D 게임을 졸업작품으로 제작한 경험을 바탕으로, 게임플레이어의 유저 행동분석을 위한 표정 행동 분석툴 개발 프로젝트에 참여하여 2D Landmarker로부터 특징점을 추출해 MLP 기반의 감성분류기능을 제작했고, Emotion Tracer라는 프로젝트 출시 경험을 통해 머신러닝을 접하게 되었습니다.
- 모바일 게임회사 Com2us 에 재직하면서 Tensorflow를 게임에 접목해 볼 기회가 생겨, “던전딜리버리” 게임에 강화학습 기반의 Playing Agent를 제작한 경험이 있습니다. 이 경험은 딥러닝을 게임이 적용해 볼 수 있는 첫 경험이 되었고, 이를 바탕으로 NCSOFT AI Center 조직에 합류하게 되었습니다.
- NCSOFT에서 개발중인 3-Match 게임의 난이도 측정을 위해 MCTS 기반의 Playing Agent를 제작하여 승률을 통한 난이도 예측 도구를 제작하였습니다. 이를 확장하여 난이도에 따른 레벨 생성 도구 제작을 진행했고, 레벨 생성을 위한 생성모델 GAN 을 채택하고 해당 모델에 학습 데이터를 생성하기 위해 패턴생성 기법인 Cellular Automata 를 이용해 학습 데이터를 제작했습니다. 해당 기술을 이용해 사내 NCDP2020에서 발표를 진행하였고, 특허출원까지 할 수 있었습니다.
- 딥 러닝의 시대를 경유하며, 전 세계적으로 다양한 모델의 트랜드가 있었지만 게임에 적용가능한 기술에 대해서는 연구되고 적용될 방향이 많다고 생각합니다. 현재는 딥 러닝이 대표가 되는 AI 기술을 이용해 게임 컨텐츠, 밸런싱, 제작파이프라인 가속화 등 기여할 부분이 많다고 생각하고 있습니다. 빠른 시일내로 게임에 AI 기술을 적용하여 출시 하고 싶은 목표를 가지고 최근 LLM 모델의 대표인 Transformer 모델을 확장한 GPT 모델들도 게임에 적용에 대해 관심있게 탐구중이며 게임에서 해결가능한 문제를 탐구하며 연구중입니다.

# 경력 상서

- 2017.12.26 홍익대학교 EGLab, “감성분석기술” JTBC 특허기술이전

- 얼굴 표정 분석기, Emotion Tracer
- 생체 신호 탐지기, Emotion Insole

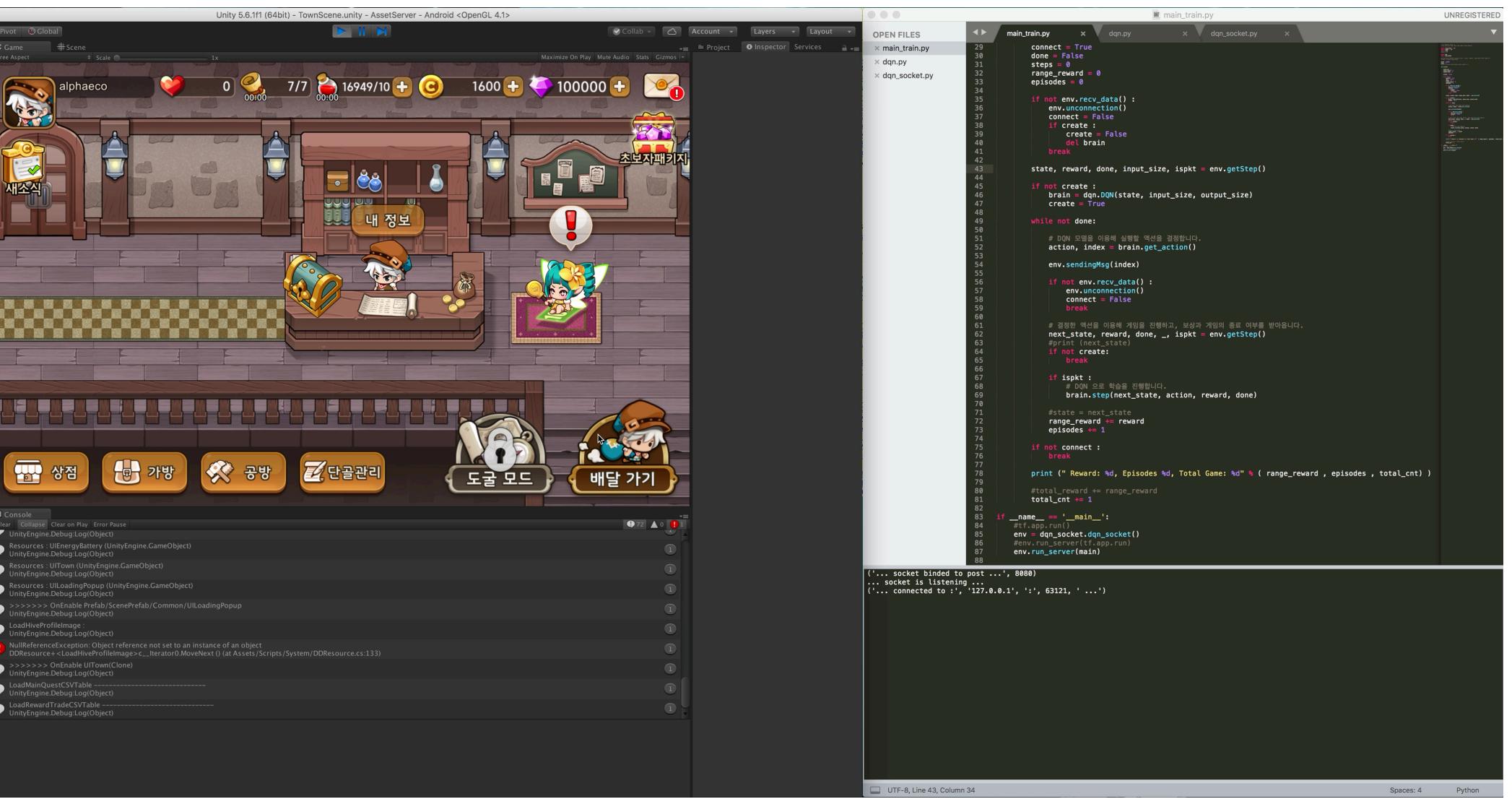


Emotion Tracer 툴 및 GCP 기반 데이터 결과 확인 연동서비스

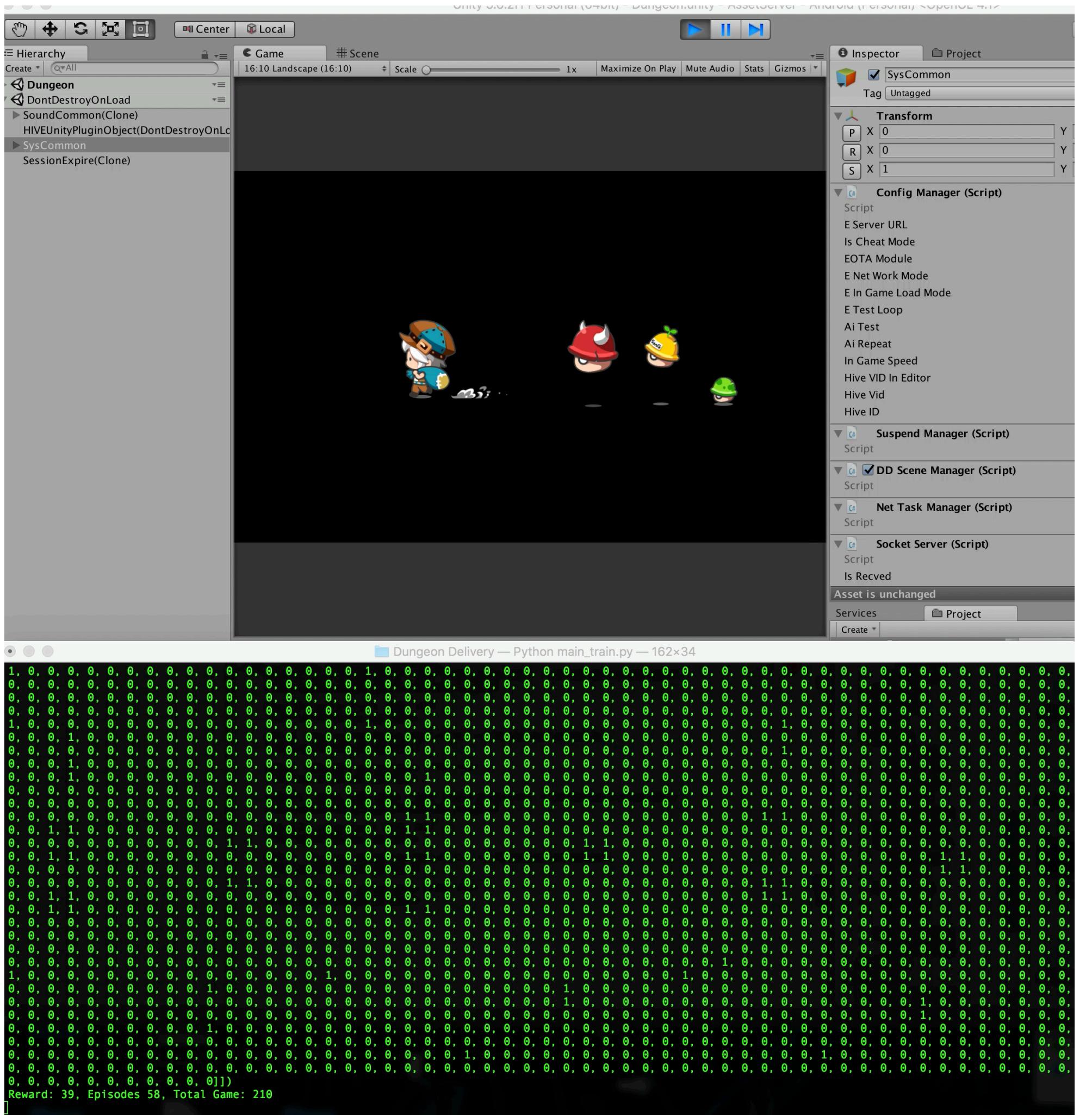
# 경력 상세

- 2017.07 ~ 2017.12 Com2us “던전 딜리버리” 강화학습 에이전트 적용 테스트

- 게임에 AI 기술 적용을 위한 탐색 및 가능성 타진
  - Tensorflow로 작성된 강화학습 모델 설계 및 소켓 통신을 이용한 게임 적용
  - Gridmap 을 이용한 환경정보, Action Decision을 위한 DQN 적용



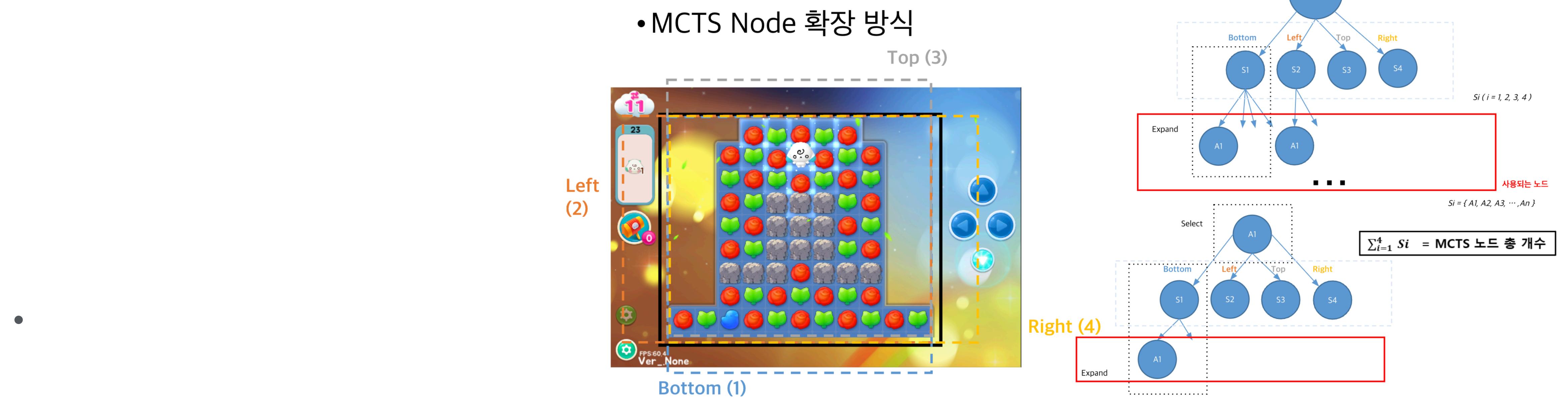
# 던전딜리버리 학습환경 (Server-Client)



# 학습된 에이전트 결과

# 경력 상세

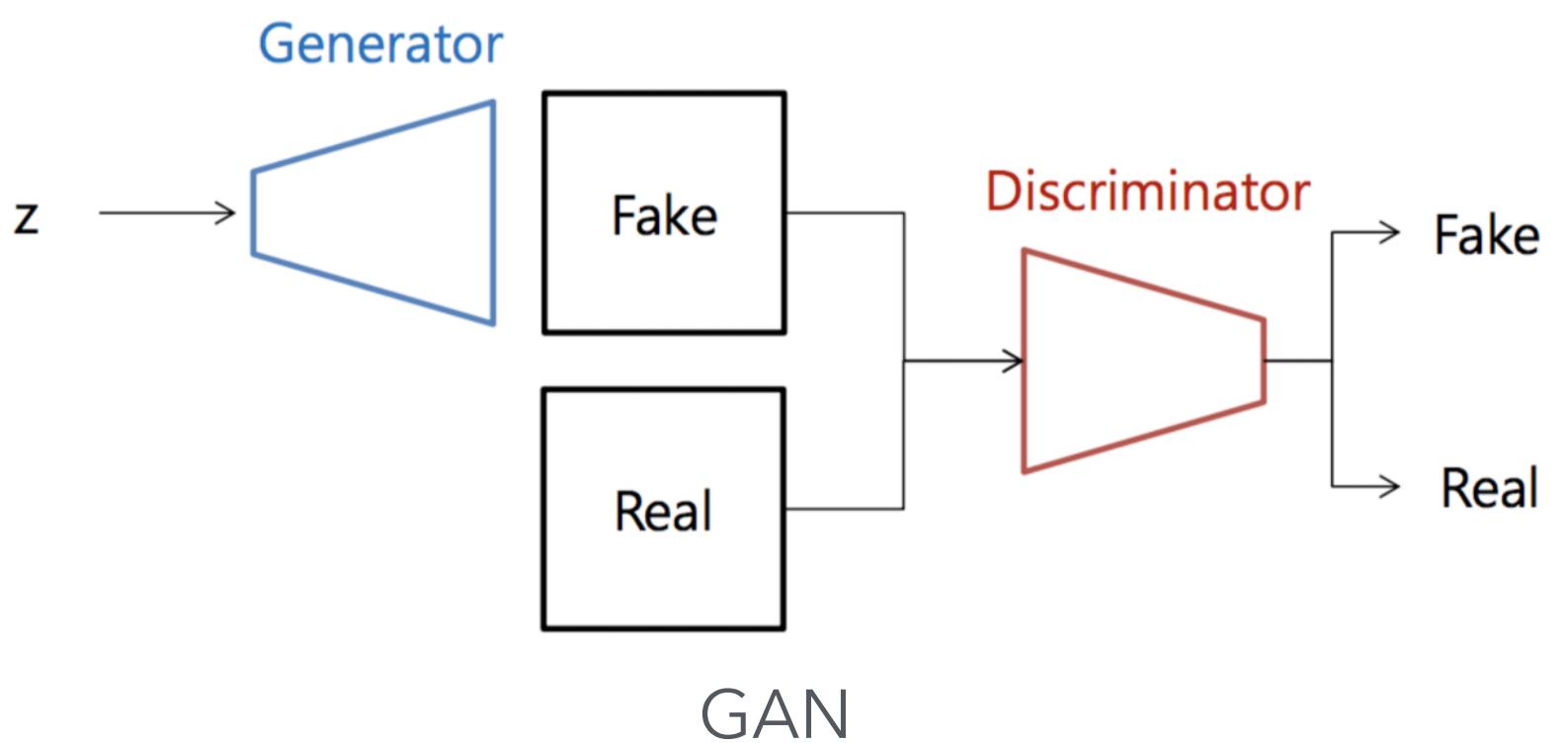
- 2018.03 ~ 2018.09 NCSOFT MCTS (Monte Carlo Tree Search) 기반 3-Match 퍼즐 게임 난이도 평가 Agent 개발
  - 3-Match 게임의 난이도를 시뮬레이션 기반으로 평가하기 위한 Agent 개발
  - MCTS 기능 구현 및 게임 로직에 Agent 적용
  - 가속화 및 난이도 평가 결과를 통계치를 이용해 도출



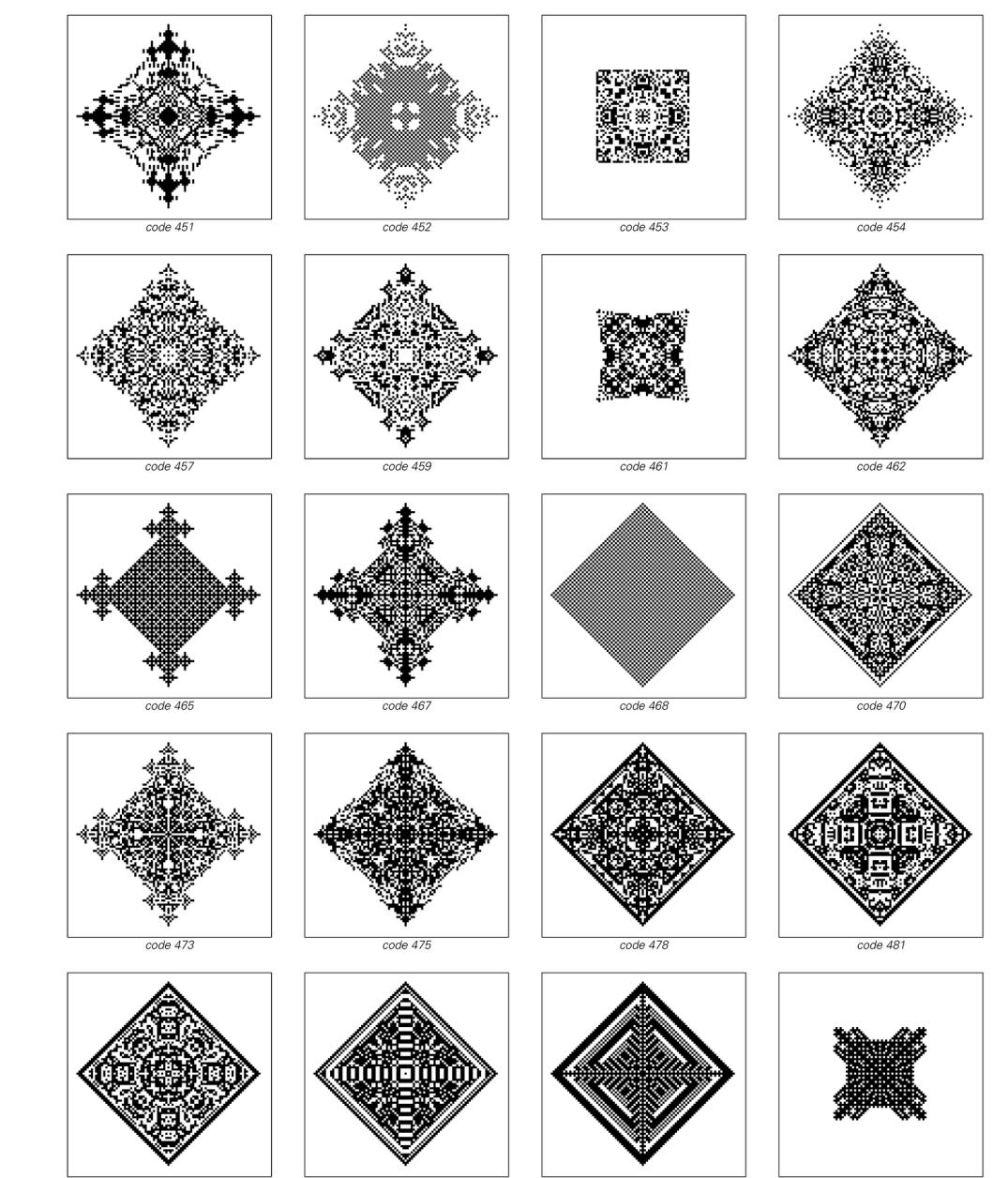
# 경력 상세

- 2018.10 ~ 2021.03 NCSOFT CA (Cellular Automata) 기반 3-Match 퍼즐게임 Level Generator 개발

- 3-Match 퍼즐 게임 기획에 활용하기 위해 제작 및 제안
- CA 기능 구현 및 CA 기반 Generator 개발
- CA Generative Combination 을 제안하여 Co-relationship pattern 개선
- GAN (Generative Adversarial Networks) 모델 학습데이터로 활용



Game of Life



Patterns generated by two-dimensional cellular automata from the previous page, but now after twice as many steps.

174

Patterns by CA

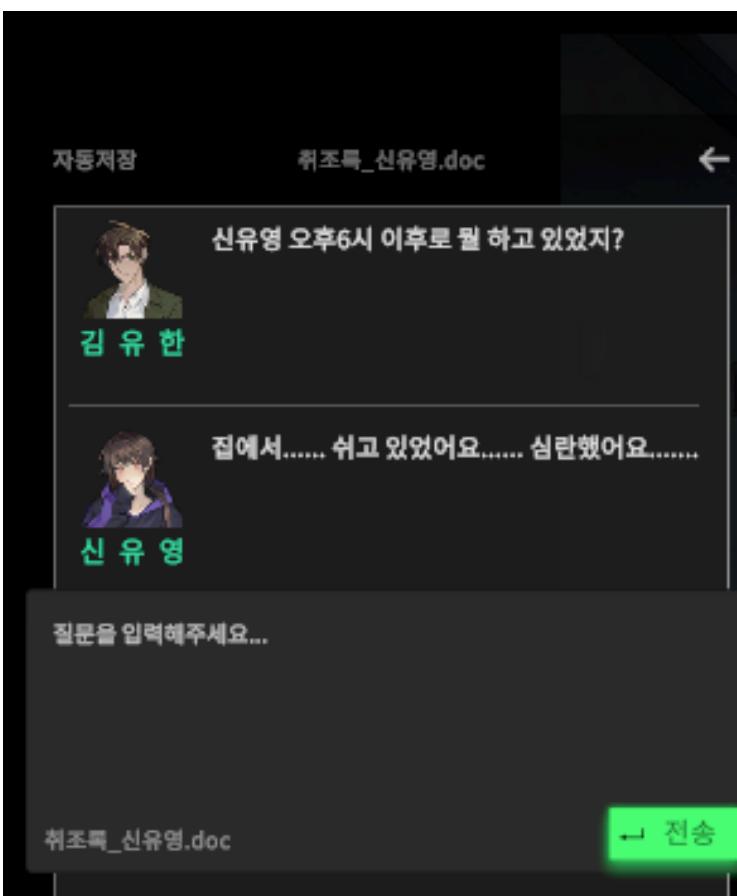
# 경력 상세

- 2021.03 ~ 2021.10 NCSOFT CA (Cellular Automata) 기반 3-Match 퍼즐게임 Level Generator Web 서비스 개발
  - 3-Match 퍼즐 게임 기획 사용성에 대한 서비스 제안
  - GUI 제공 및 학습된 GAN 모델의 유저 친화적인 Parameter 조작을 통한 \*.ZIP 형태의 결과물 제공
  -

# 경력상서

- 2024.07 ~ 2024.08 추리게임용 LLM Chatbot 경량화 및 런타임 최적화
  - Polyglot-kr-1.3B 을 PEFT LoRA 학습
  - LLM Python server 경량화를 위한 batch 로 분리 및 구조 설계
  - Application service 를 위한 CPU 추론 모델 ONNX exporting

```
model loading...
/Users/jangyoung/Documents/Github/LiarHeart/Assets/StreamingAssets/Scripts
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.
model completely loaded
실행 시간: 11.57147 초
클라이언트와 연결을 대기합니다...
클라이언트가 연결되었습니다. 주소: ('127.0.0.1', 61251)
<socket.socket fd=5, family=2, type=1, proto=0, laddr=('127.0.0.1', 25001), raddr=('127.0.0.1', 61251)>
받은 답변: Shinyooyoung!@#$#@!신유영 오후 6시 이후로 뭘 하고 있었지?
답변 생성 중
character: Shinyooyoung
questionText: 신유영 오후 6시 이후로 뭘 하고 있었지?
<startoftext>이름:신유영
질문: 신유영 오후 6시 이후로 뭘 하고 있었지?
답변:
생성된 전체 텍스트: <startoftext>이름:신유영
질문: 신유영 오후 6시 이후로 뭘 하고 있었지?
답변: 집에서..... 쉬고 있었어요..... 심란했어요.....
감정: 감정 없음
생성된 답변: 집에서..... 쉬고 있었어요..... 심란했어요.....
생성된 감정: 감정 없음
실행 시간: 12.23388 초
```



추리게임에 적용된 화면

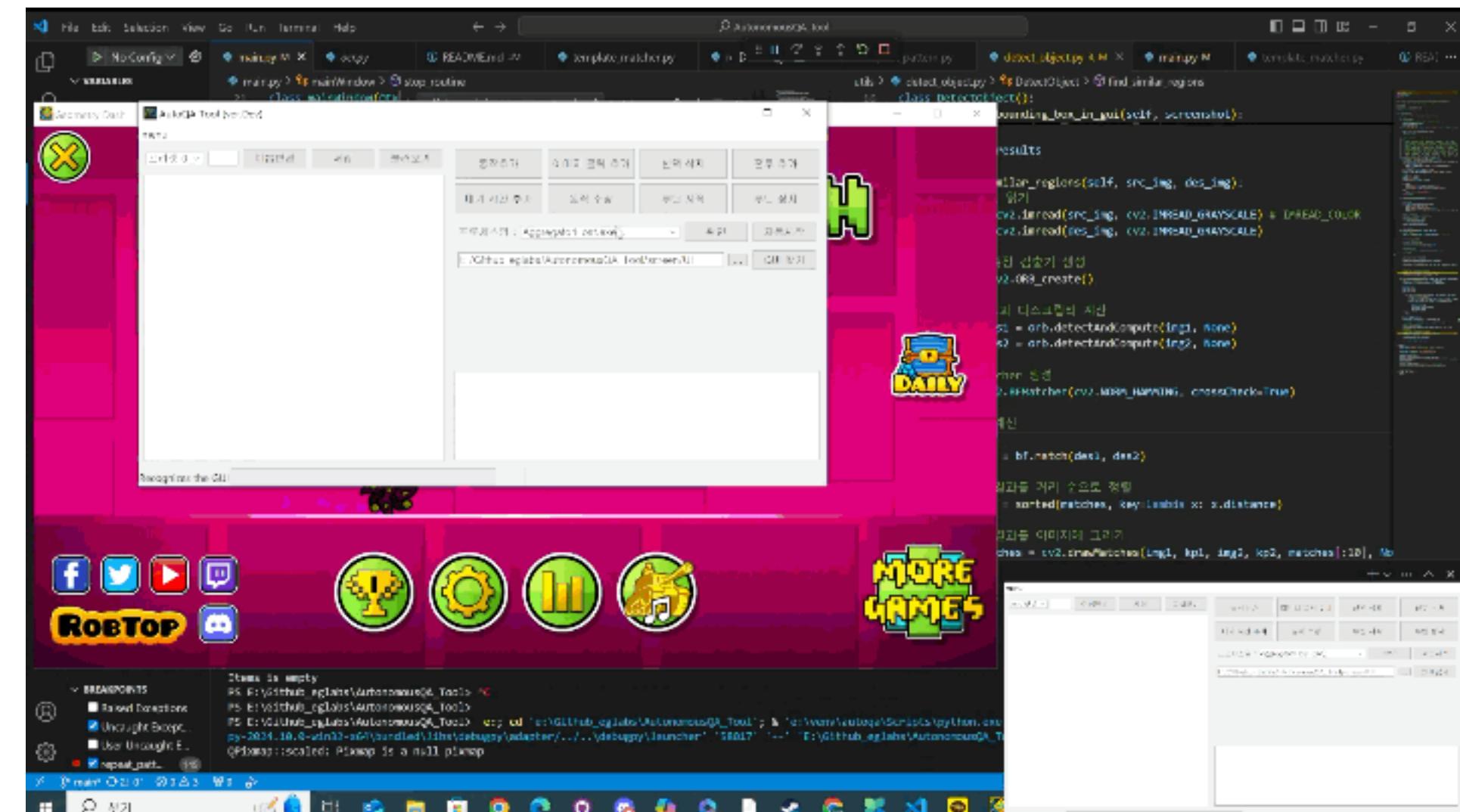
```
PeftModelForCausalLM(
    (base_model): LoraModel(
        (model): GPTNeoXForCausalLM(
            (gpt_neox): GPTNeoXModel(
                (embed_in): Embedding(30080, 2048)
                (emb_dropout): Dropout(p=0.0, inplace=False)
                (layers): ModuleList(
                    (0-23): 24 x GPTNeoXLayer(
                        (input_layernorm): LayerNorm((2048,), eps=1e-05, elementwise_affine=True)
                        (post_attention_layernorm): LayerNorm((2048,), eps=1e-05, elementwise_affine=True)
                        (post_attention_dropout): Dropout(p=0.0, inplace=False)
                        (post_mlp_dropout): Dropout(p=0.0, inplace=False)
                        (attention): GPTNeoXSdpaAttention(
                            (rotary_emb): GPTNeoXRotaryEmbedding()
                            (query_key_value): lora.Linear(
                                (base_layer): Linear(in_features=2048, out_features=6144, bias=True)
                                (lora_dropout): ModuleDict(
                                    (default): Dropout(p=0.1, inplace=False)
                                )
                            )
                            (lora_A): ModuleDict(
                                (default): Linear(in_features=2048, out_features=8, bias=False)
                            )
                            (lora_B): ModuleDict(
                                (default): Linear(in_features=8, out_features=6144, bias=False)
                            )
                            (lora_embedding_A): ParameterDict()
                            (lora_embedding_B): ParameterDict()
                            (lora_magnitude_vector): ModuleDict()
                        )
                        (dense): Linear(in_features=2048, out_features=2048, bias=True)
                        (attention_dropout): Dropout(p=0.0, inplace=False)
                    )
                )
                (mlp): GPTNeoXMLP(
                    (dense_h_to_4h): Linear(in_features=2048, out_features=8192, bias=True)
                    (dense_4h_to_h): Linear(in_features=8192, out_features=2048, bias=True)
                    (act): GELUActivation()
                )
            )
        )
        (final_layer_norm): LayerNorm((2048,), eps=1e-05, elementwise_affine=True)
    )
    (embed_out): Linear(in_features=2048, out_features=30080, bias=False)
)
```

```
▶ generate_answer("신유영", "너는 사건이 일어났던 시간에 무엇을 하고 있었지?")
generate_answer("박강현", "너는 사건이 일어났던 시간에 무엇을 하고 있었지?")
generate_answer("이도윤", "너는 사건이 일어났던 시간에 무엇을 하고 있었지?")
▶ 이름:신유영
질문: 너는 사건이 일어났던 시간에 무엇을 하고 있었지?
답변: 그 시간엔... 어디 있었는지 말 못합니다. 하지만 달리 같이 있었던 사람은 없었어요. 믿어주세요!
감정: 긴장
이름:박강현
질문: 너는 사건이 일어났던 시간에 무엇을 하고 있었지?
답변: 화요일 그 시간엔 집에 있었을 겁니다... 아마도요.
감정: 긴장
이름:이도윤
질문: 너는 사건이 일어났던 시간에 무엇을 하고 있었지?
답변: 화요일 그 시간엔 집에 있었습니다. 피곤해서 자고 있었죠.
감정: 감정없음
```

PEFT LoRA 학습결과

# 경력 상세

- 2024.08 ~ 2024.09 이미지 기반 General AutoQA Tool
  - OpenCV 기반 Scaled Template Matching 알고리즘 적용
  - Thread Pool (GIL) 이용한 속도 최적화 및 Window Handler 적용
  - Hierarchy Folder Template Image 탐색 규칙 적용



AutoQA Tool 동작 화면

# 활동

- 2023.04.17 NC Research AVATAR
- 2020.04.28 NCDP2020 “매치3 퍼즐의 레벨 자동 생성 및 난이도 평가 도구 개발” 발표
- 2019.03 - 2022.12 NCFellowship 운영관리
- 2017.12 JTBC “이론상 완벽한 남자” 특허 기술 이전
- 2016.01.28 HCI Korea 2016, “게임 플레이 분석을 위한 얼굴 표정량 변화탐지 시스템” 발표
- 2015.12.10 PeerJ (SCI-E) 2015, “Efficiently Detecting Outlying Behavior in Video-Game Players”
- 2015.11.12 - 2015.11.15 G-Star 2015, “Emotion Tracer” 전시

# 기타

- 경력사항 : <https://akillness.github.io/assets/pdf/resume.pdf>
- 기술블로그 : <https://akillness.github.io/about/>

QnA