

좀비 피하기 AI 게임 제작

최종보고

2020.06.21

6조 이 졸업의 끝을 잡고

박지수, 한다란, 남태수, 이유진

목차

1. 서론

- 1.1 연구/개발 배경
- 1.2 프로젝트 내용

2. 본론

- 2.1 시스템/GUI 구성
- 2.2 사용 사례
- 2.3 적용 기법 및 기술
- 2.4 핵심 연구/개발 과제
- 2.5 업무 분담/일정 계획
- 2.6 개발 환경 및 구현

3. 결론

- 3.1 달성 목표 및 성과
- 3.2 프로젝트의 의의

1. 서론

1.1 연구/개발 배경

1.2 프로젝트 내용

1.1 연구/개발 배경

01

GA(Genetic Algorithm)

중요성 부각

- 최근 인공지능과 머신러닝의 급속한 발전으로 GA에 대한 관심 증가
- 수학적으로 명확히 정의되지 않은 문제에도 적용 가능하여 유용성과 확장성을 가짐

02

개체의 유연한 대응능력의

필요성

- 학습데이터에 대해서는 오차가 감소하는데 비해 실제 데이터에 대해서는 오차가 증가하는 과대적합(overfitting) 문제 존재
- 다양한 변수의 존재와 증가로 유연한 대응능력의 필요성 증가

1.1 연구/개발 배경

본 프로젝트의 결과가 가지는 의미

- 인공지능망을 활용한 일반적인 문제 해결 모델의 경우 훈련 환경과 테스트 환경이 다를 때 설립한 모델의 성능이 하락할 수 있음
- 단계적 학습이 인공지능망을 사용한 문제 해결 모델의 성능을 향상시킴을 증명함

향후 활용 가능성

- 인공지능망으로 해결하고자 하는 문제의 환경이 정적이지 않고 정확한 문제 해결 방향을 정하기 어려울 때 단계적 학습법이 해의 단서가 될 수 있음

1.2 프로젝트 내용

01

GA를 이용해 세대가 지남에 따라 진화하는
플레이어 AI 제작

02

환경변화를 추가하여 기존보다 개선된 성능을
가진 개체 생성

1.2 프로젝트 내용

프로젝트 개발 순서

01 구조 설계 와 클래스 구현

- 플레이어, 좀비, 아이템 클래스
- 외부 정보를 받아들이는 시각역할을 하는 센싱 함수

02 인공 신경망과 유전알고리즘 구조 설계와 구현

- 순전파 형식의 다층신경망구조
- 유전알고리즘의 기본 함수와 기능

1.2 프로젝트 내용

프로젝트 개발 순서

03 그래픽 요소 도트 작업과 맵의 구현

- 플레이어, 좀비, 아이템, 벽 도트 제작
- 제작한 벽 도트를 활용해 다양한 맵 구조 생성

04 A* 알고리즘 구현

- 좀비의 행동 패턴 알고리즘

1.2 프로젝트 내용

프로젝트 개발 순서

05 게임 실행 파일인 `game.py` 파일 제작

- 다른 파일들에 구현한 모든 기능들을 종합하여 정상적으로 게임이 구동되도록 제작
- 게임의 성능에 영향을 미치는 하이퍼 파라미터 최적화

06 테스트 맵 구현

- 성능 테스트를 위한 신규 맵 제작

1.2 프로젝트 내용

프로젝트 개발 순서

07 단계적 학습군 과 대조군의 생존성 검증

- 생존 시간의 척도인 평균 step 비교
(생존 시간 = step)

08 보고서 작성

1.2 프로젝트 내용

Abstract

1. 본 프로젝트는 인공지능망을 사용한 문제풀이에 있어 문제의 환경 변화에 대응하기 위한 방법으로 동일 환경 학습이 아닌 단계적 환경변화 학습의 효과에 대한 실험이다.
2. 테스트를 위해 해당 조는 좀비를 피해 살아남아야 하는 인공지능 생존 게임을 제작하였다. 게임의 구성 요소로서는 좀비, 플레이어, 아이템, 벽이 있다.
3. 좀비는 A* 알고리즘을 기반으로 플레이어를 쫓는다. 플레이어는 인공지능망과 유전알고리즘을 기반으로 학습, 진화를 거듭하며 좀비에게 닿을 시 사망, 좀비로 변한다.
4. 플레이어와 좀비는 시야의 범위 제한이 있어 자신의 시야 내에 들어온 물체에 대해서만 반응할 수 있다.

1.2 프로젝트 내용

Abstract

5. 플레이어는 일정 시간이 지나면 자동적으로 사망하며 아이템을 획득할 시 생존 가능 시간이 늘어난다.
6. 주된 유전 형질로는 플레이어가 좀비에게 타겟팅이 된 시점을 기준으로 생존 시간을 사용하였다.
7. 단계적 학습군의 경우 총 4단계의 다른 맵 에서 학습이 진행되며 이후 대조군과 새로운 맵 에서 최종 생존시간 비교를 하였다.
8. 테스트 결과 단계적 학습을 거친 개체들은 그렇지 않은 대조군보다 생존성이 평균 30% 이상 우수한 것으로 밝혀졌다.

* 플레이어의 생존 시간은 플레이어가 죽기 전까지 걸은 걸음(step)으로 계산한다

1.2 프로젝트 내용

1 단계



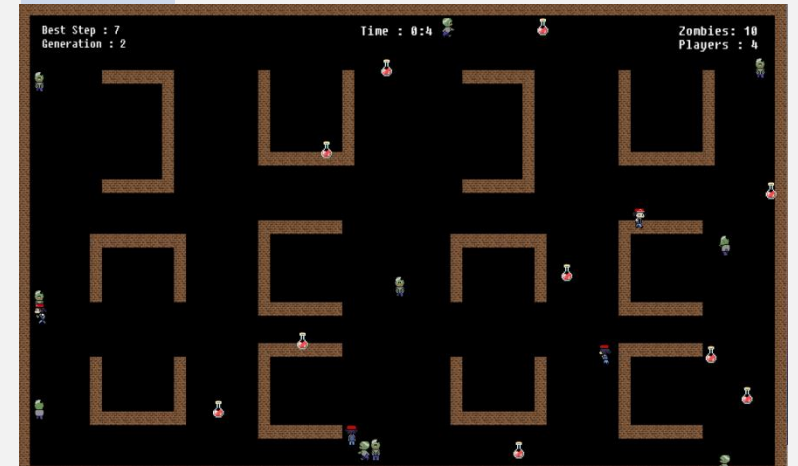
2 단계



3 단계



4 단계



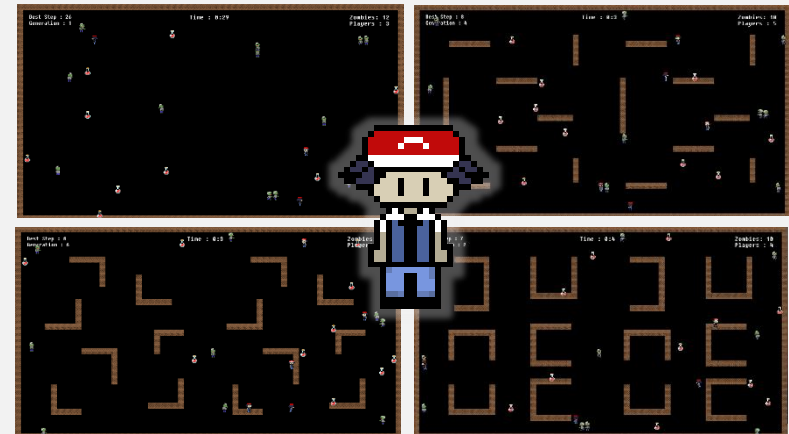
1.2 프로젝트 내용

한가지 맵에서만 학습한
플레이어 AI



VS

4가지 맵을 단계별로 학습한
플레이어 AI



2. 본론

2.1 시스템/GUI 구성

2.2 사용 사례

2.3 적용 기법 및 기술

2.4 핵심 연구/개발 과제

2.5 업무 분담/일정 계획

2.6 개발 환경 및 구현

2.1 시스템/GUI 구성

Best Step / 세대 출력

시간 출력

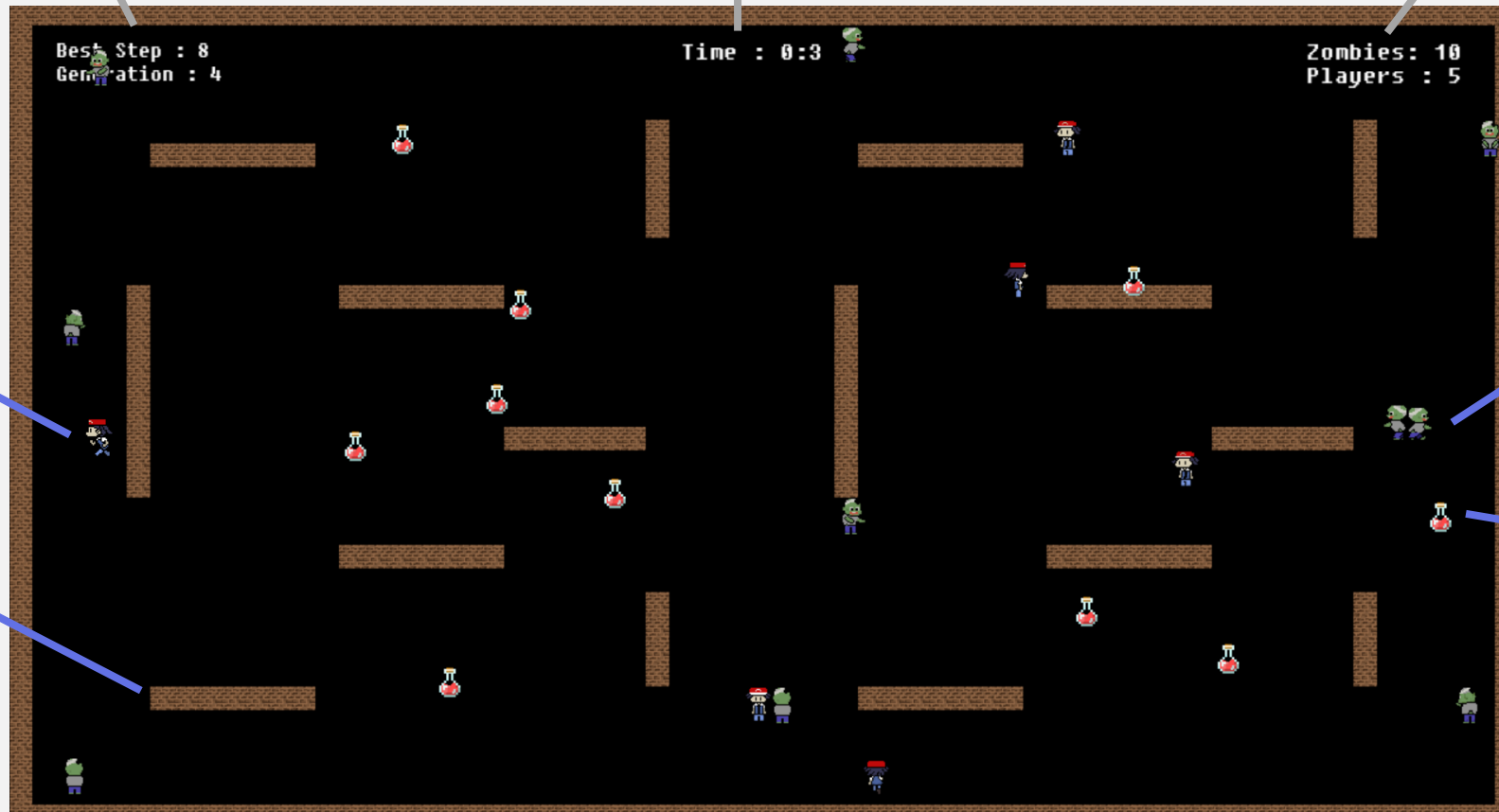
남은 플레이어 AI / 좀비 수 출력

플레이어 AI

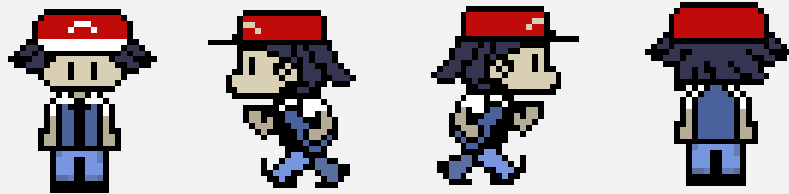
벽

좀비

아이템



2.1 시스템/GUI 구성



플레이어 AI

좀비를 피해 생존하는 방법을 학습하는 플레이어



아이템

플레이어 AI의 지속시간을 연장



좀비

플레이어 AI를 감염시키기 위해 쫓아다니는 게임 빌런



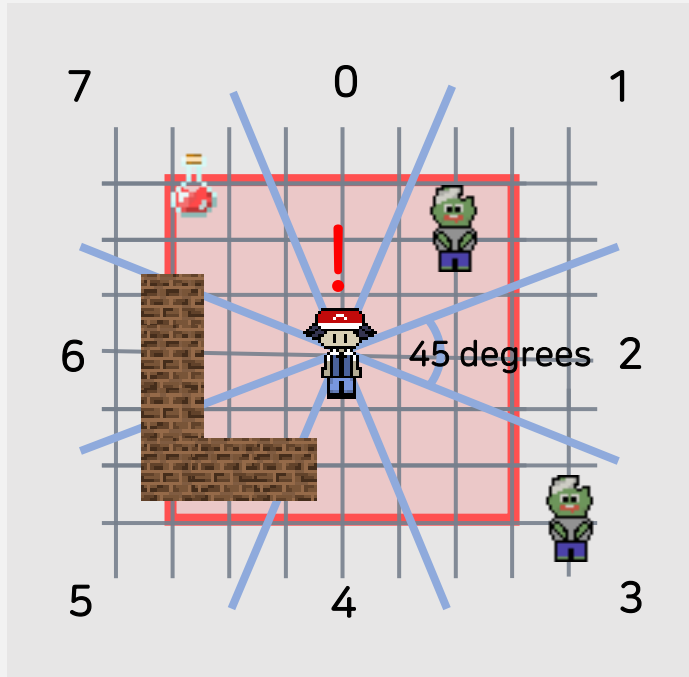
벽(맵, 오브젝트)

게임이 진행되는 배경을 구성

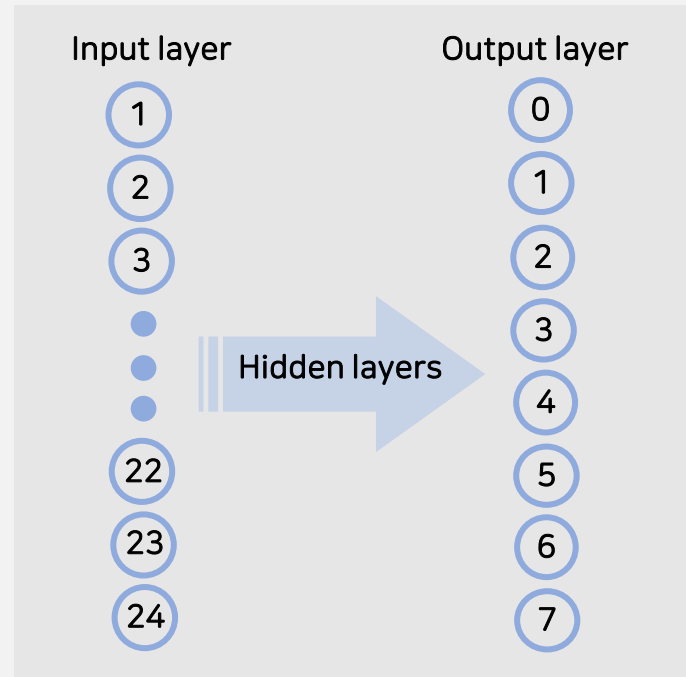
2.1 시스템/GUI 구성

1. 플레이어 AI

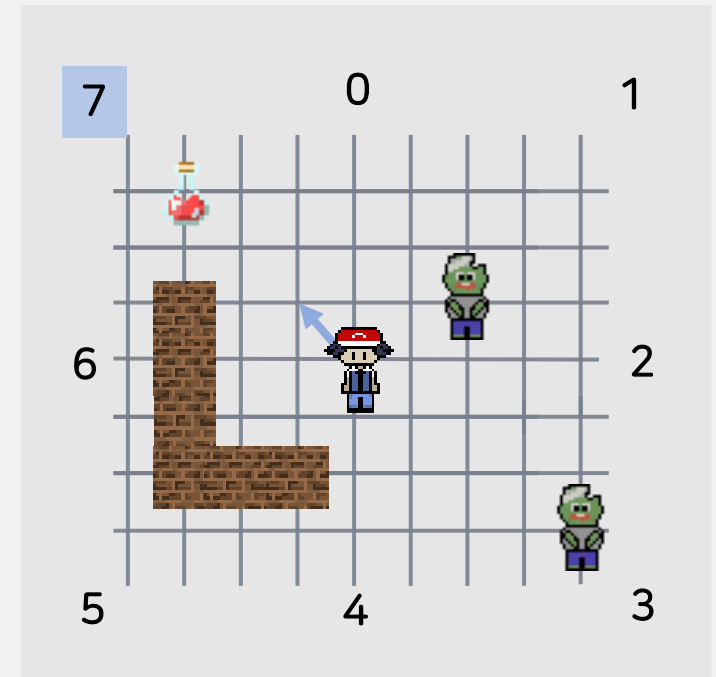
1. player-sensor



2. neural-net



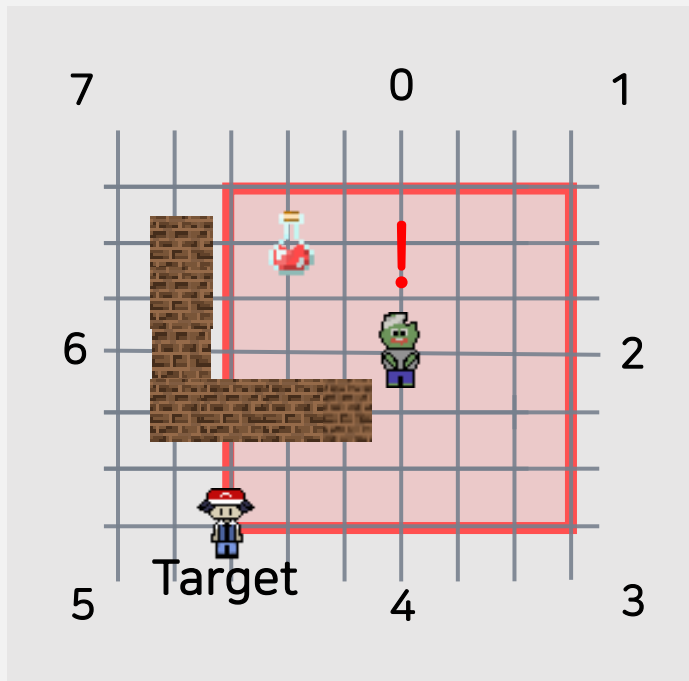
3. step



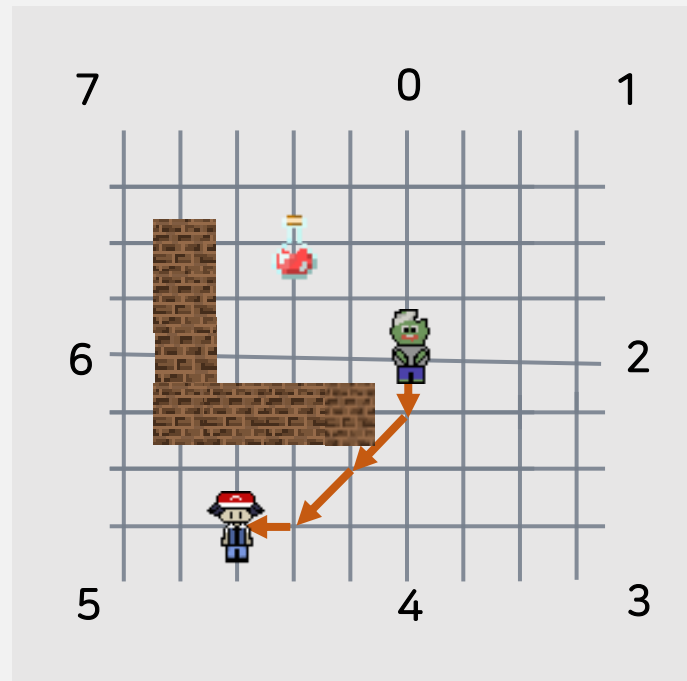
2.1 시스템/GUI 구성

2. 좀비

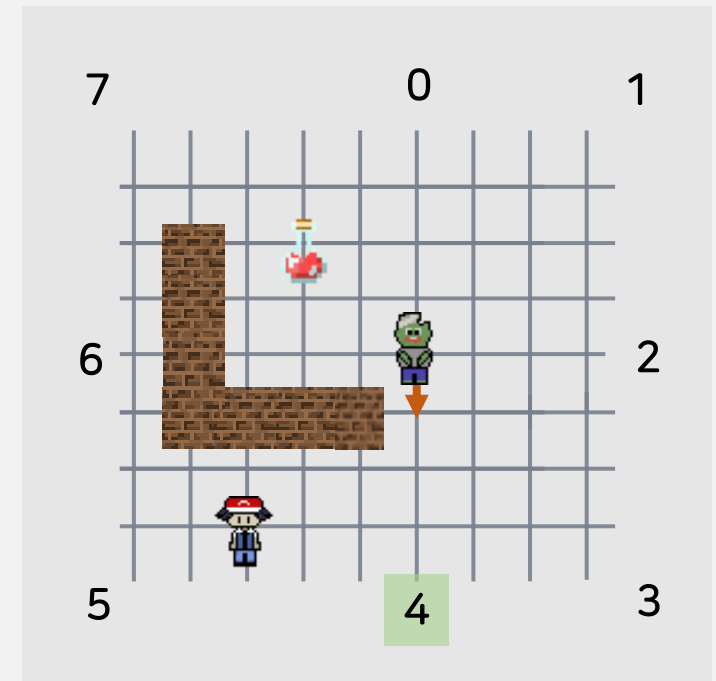
1. zombie-sensor



2. route



3. step



2.1 시스템/GUI 구성

3. 아이템

- 플레이어 AI가 아이템 획득 시 다른 임의의 자리에 아이템 재생성



4. 벽(맵, 오브젝트)

- 플레이어 AI와 좀비가 행동함에 있어 장애물의 역할을 수행

2.2 사용 사례


영상 참조

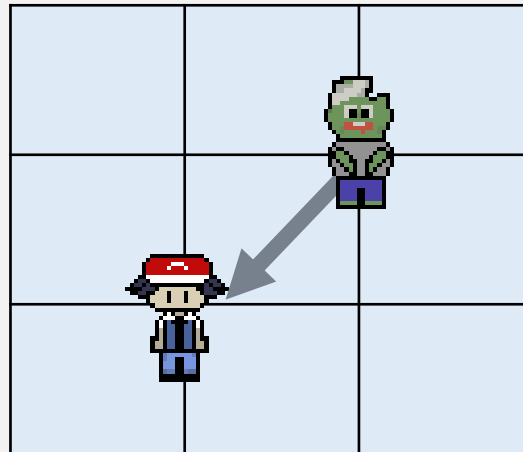
2.3 적용 기법 및 기술

01 체비셰프 거리 (Chebyshev distance)

- 두 지점 사이에서 가장 긴 축 상의 거리를 의미함
- 한 번에 이동할 수 있는 거리는 대각선 1방향과 위,아래,좌,우가 같음

$$\text{Distance} = \text{Max}(|x_a - x_b|, |y_a - y_b|)$$

2	2	2	2	2
2	1	1	1	2
2	1		1	2
2	1	1	1	2
2	2	2	2	2

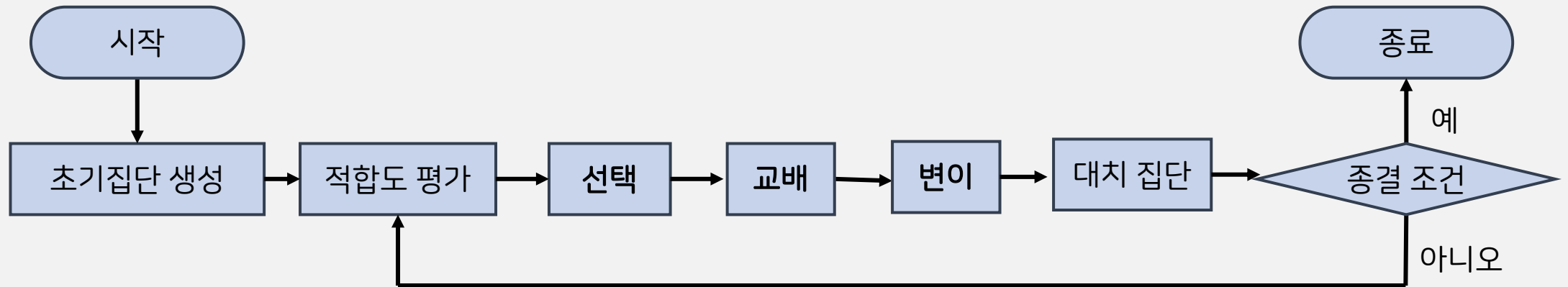


좀비가 플레이어 AI를
대각선 방향으로 추격가능

2.3 적용 기법 및 기술

02 유전 알고리즘

- 진화의 핵심 원리인 자연선택과 유전자의 개념을 이용한 최적화 기법



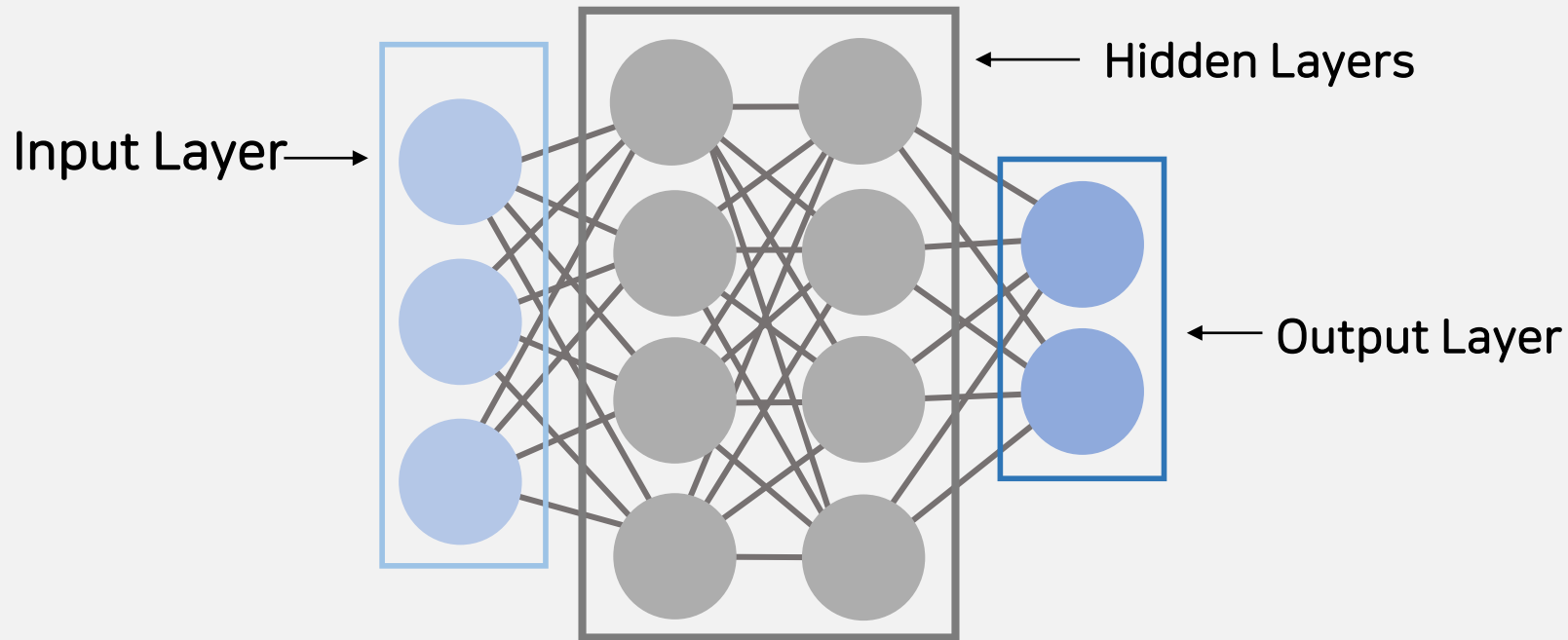
- 선택(Selection) : 적합도에 따라 다음세대에 유전자를 물려줄 후보 개체를 선별하는 연산
- 교배(Crossover) : 부모 양측으로부터 받은 유전자를 혼합하여 새로운 개체를 생성하는 연산
- 변이(Mutation) : 유전자에 직접적으로 변이를 일으켜 해를 변경하는 연산

2.3 적용 기법 및 기술

03 인공 신경망

- 인간의 뇌가 패턴을 인식하는 방식을 모방한 알고리즘

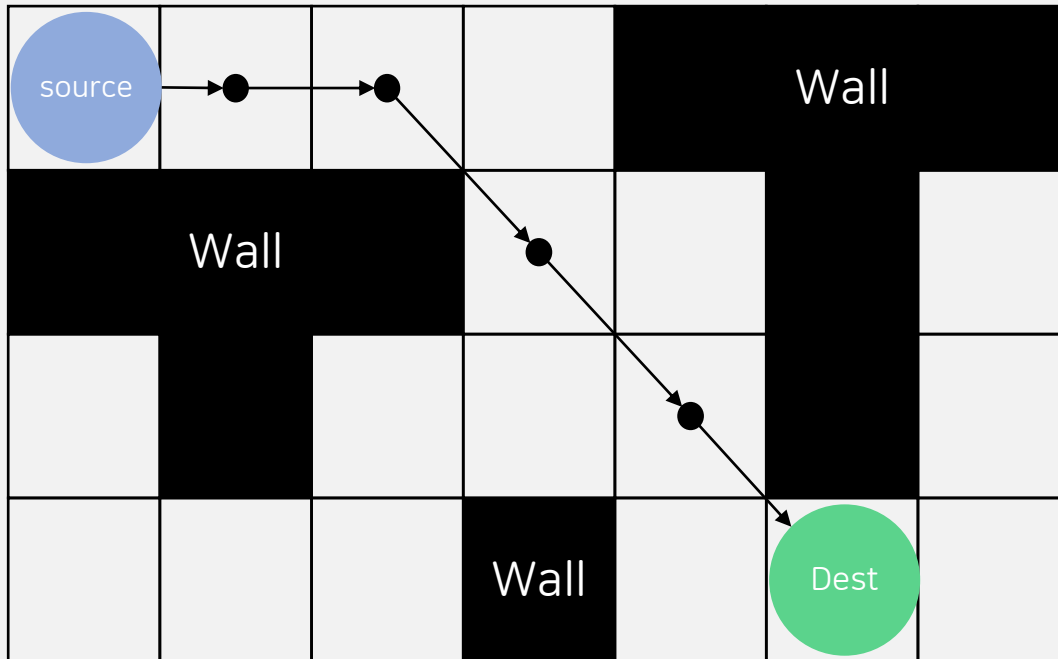
딥 러닝에서 기본적으로 이용되는 **Deep Feedforward Network(DFN)**를 이용



2.3 적용 기법 및 기술

04 A* algorithm

- 두 지점간 최단거리 파악에 활용하는 알고리즘

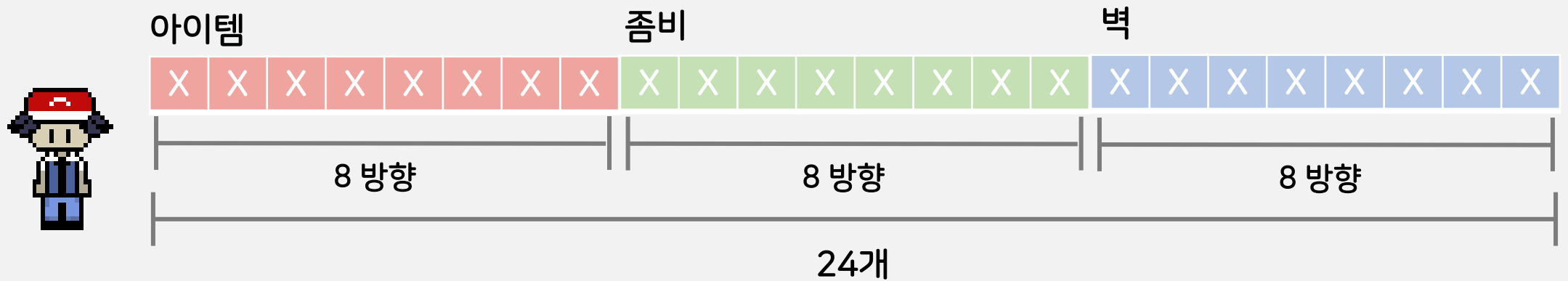


- 장애물이 있는 상태에서 목표지점까지의 cost가 제일 적은 경로를 탐색
- 좀비가 플레이어를 발견하고 쫓아가는 상황에 적용

2.4 핵심 연구/개발 과제

1. 인공 신경망 개발

(1) 뉴런과 각 layer들의 구성

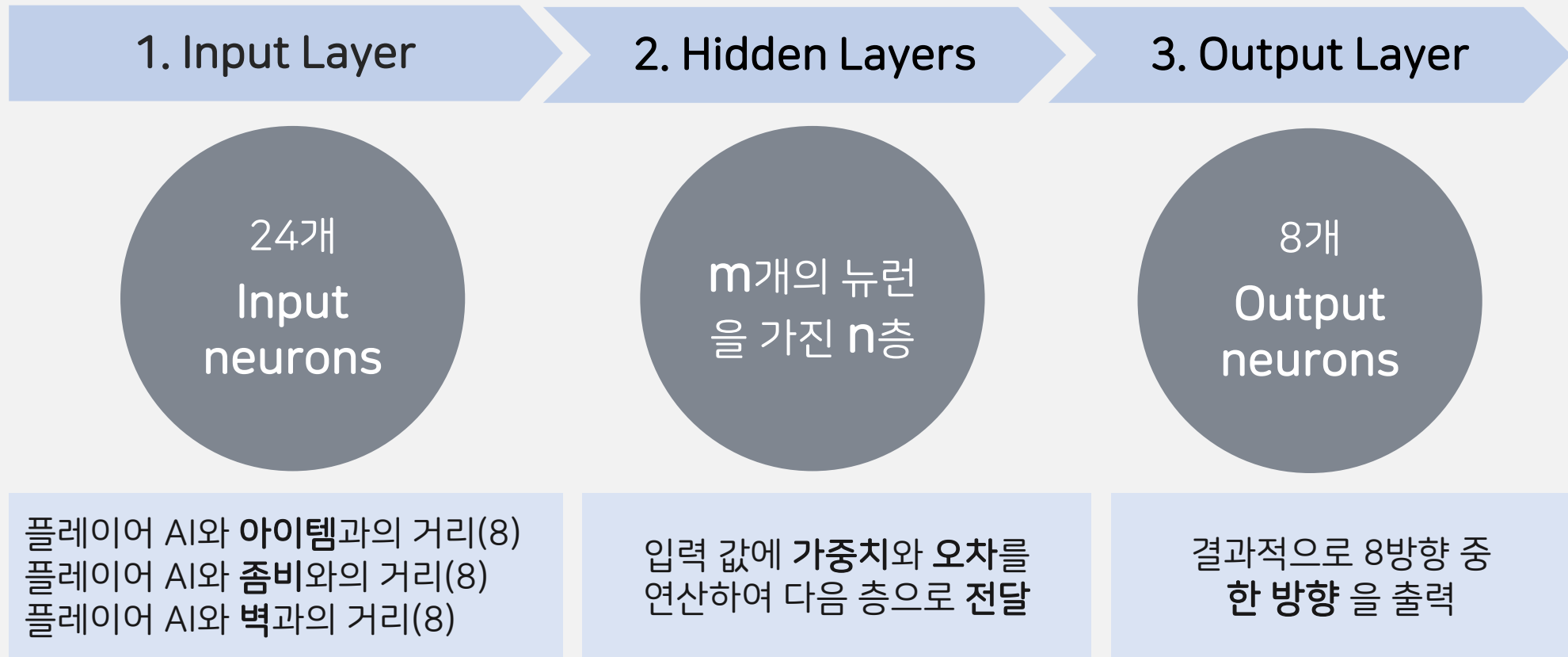


- 플레이어가 외부로부터 받아들이는 정보
- 8방향 * 3개 타겟으로 **신경망의 입력값으로 24개의** 인공신경망 입력 노드를 구성
- Input value = 1 / 플레이어AI와 센서가 인식한 물체와의 거리

2.4 핵심 연구/개발 과제

1. 인공 신경망 개발

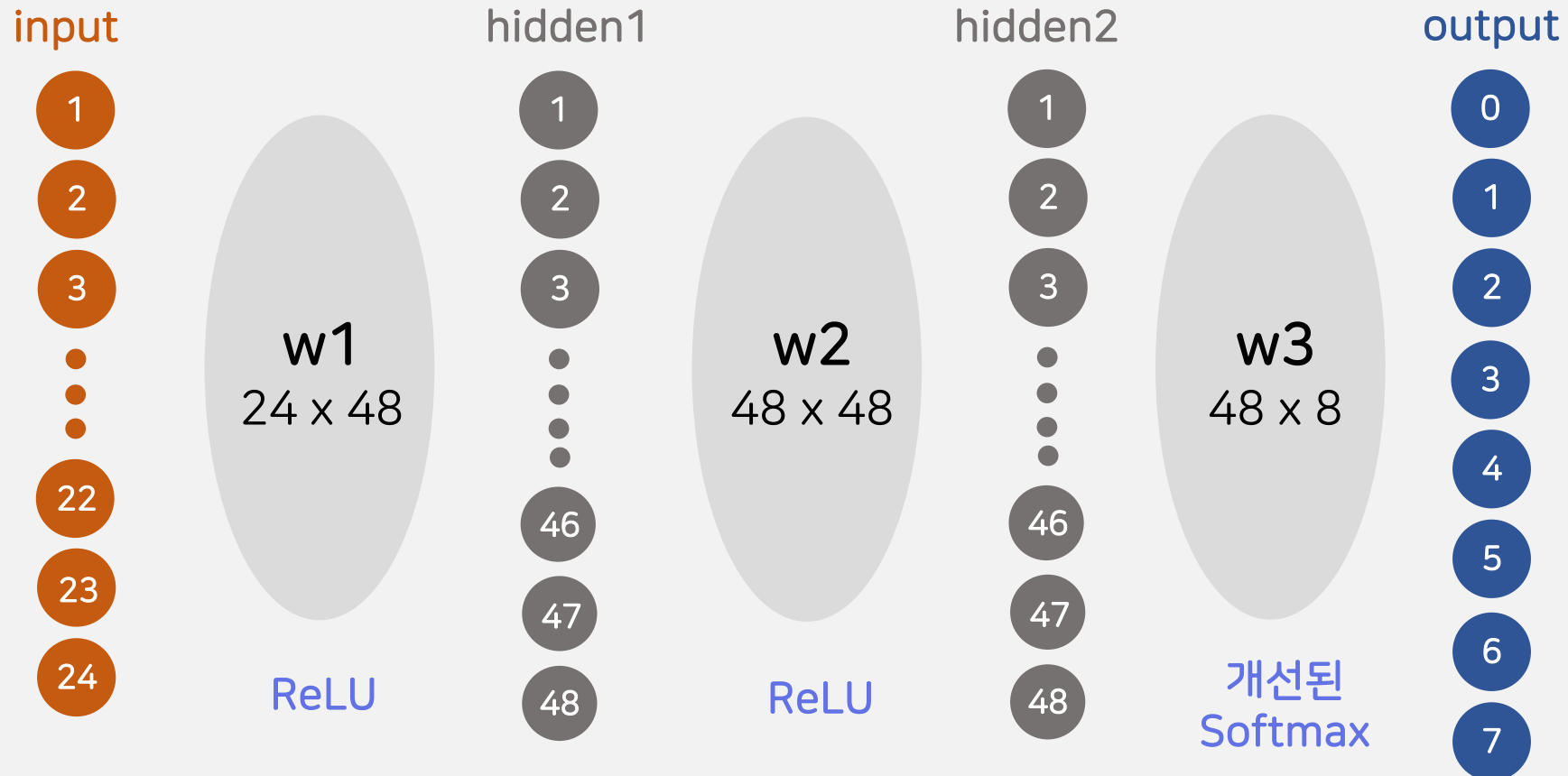
(1) 뉴런과 각 layer들의 구성



2.4 핵심 연구/개발 과제

1. 인공 신경망 개발

(2) 인공 신경망 구조 및 활성화 함수



2.4 핵심 연구/개발 과제

1. 인공 신경망 개발

(3) 초기 가중치 선언

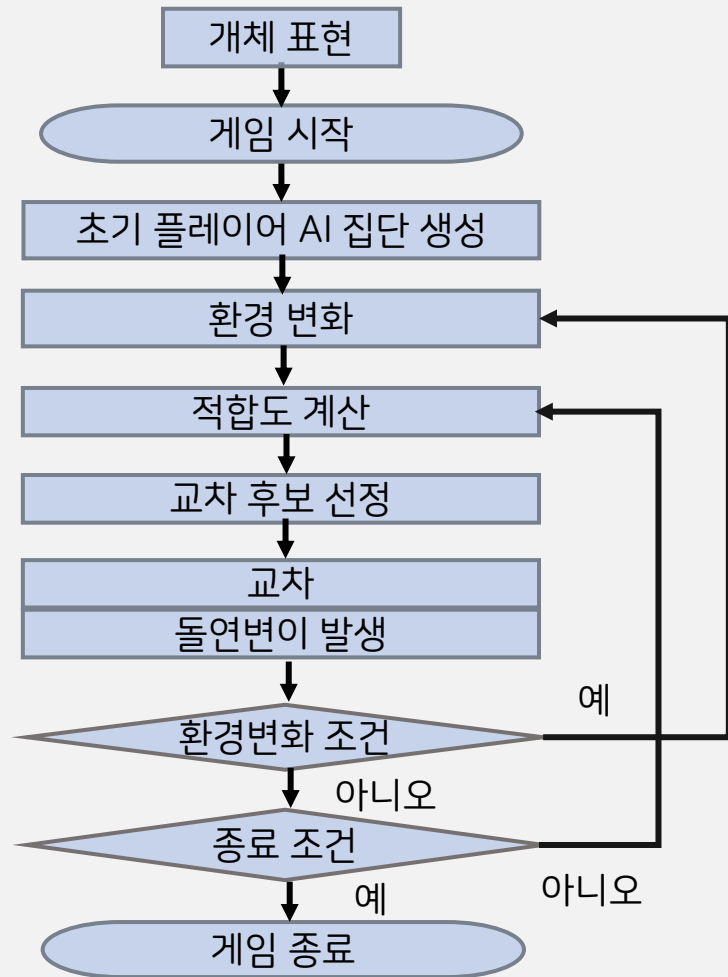


- 가중치는 GA에서 **염색체** 역할을 하며, 입력된 신호가 결과에 주는 영향력을 조절하는 요소로 작용함
- 표준정규분포를 따르는 난수배열이며, ReLU 계열 활성화 함수의 초기화 방식인 **He 초기화** 를 사용

(He initialization: 가중치 / 인풋 노드 개수의 절반의 제곱근)

2.4 핵심 연구/개발 과제

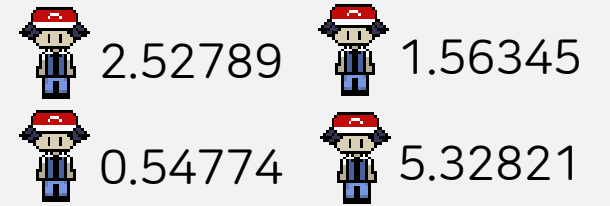
2. 플레이어 AI의 유전 알고리즘 개발



초기 플레이어 AI 집단

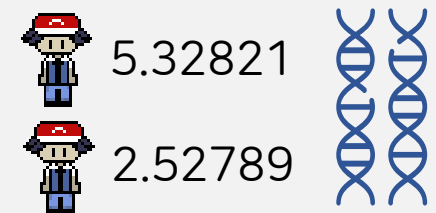


각 플레이어 AI의 적합도 계산



교차 & 돌연변이
+
맵 변화

우수한 유전자 선택

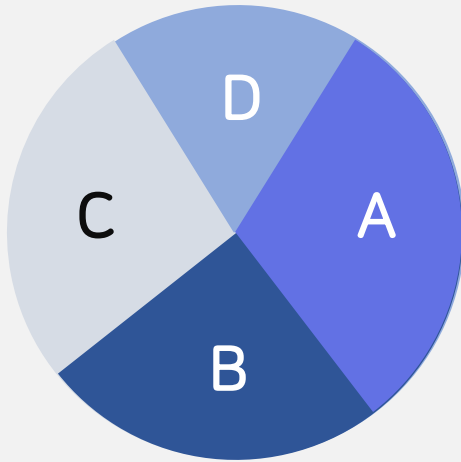


2.4 핵심 연구/개발 과제

2. 플레이어 AI의 유전 알고리즘 개발

(1) 유전관련 변수 설정

1. 선택방식



Roulette wheel selection



내림차순 선택연산

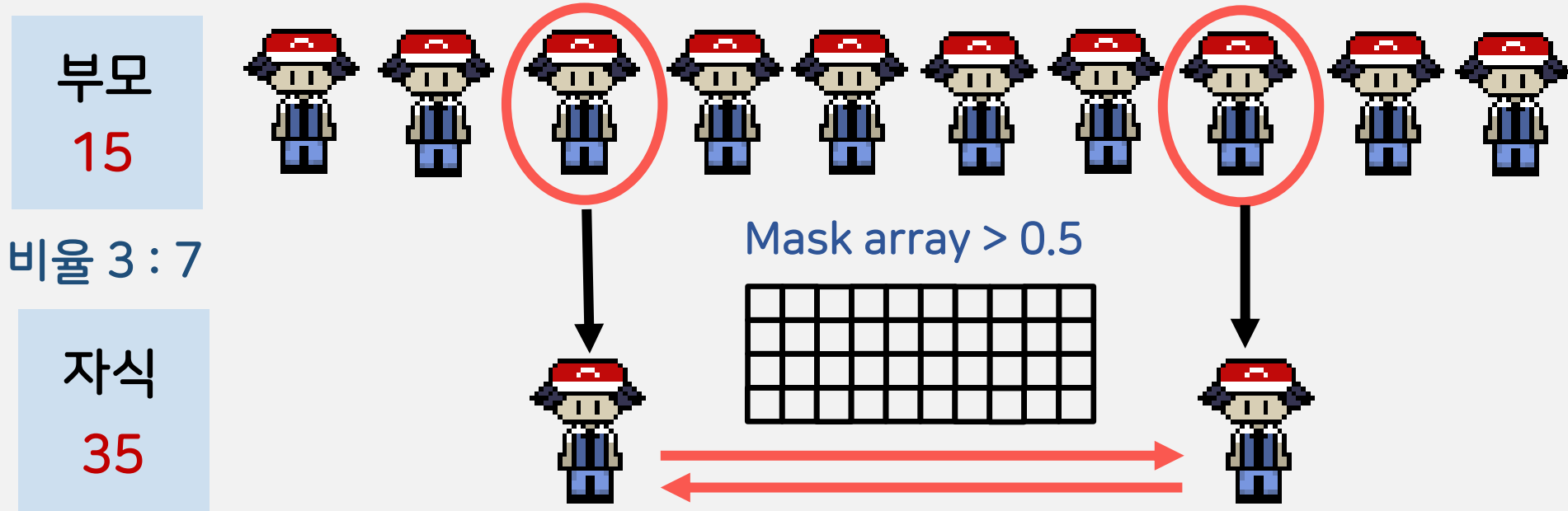
2.4 핵심 연구/개발 과제

2. 플레이어 AI의 유전 알고리즘 개발

(1) 유전관련 변수 설정 - 지역 최적화 최소화 수치 탐색

2. 교차 타입

Uniform binary crossover



2.4 핵심 연구/개발 과제

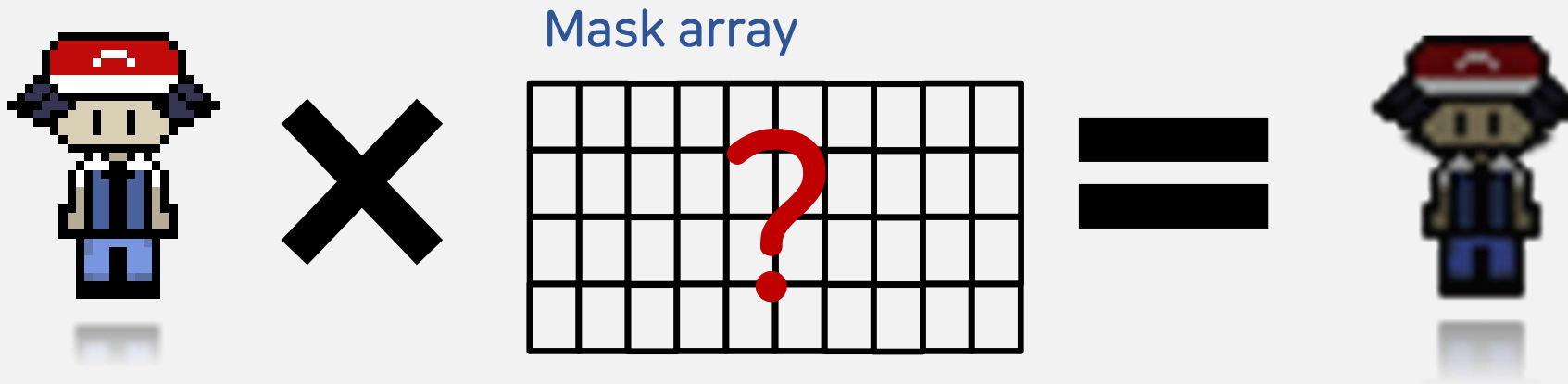
2. 플레이어 AI의 유전 알고리즘 개발

(1) 유전관련 변수 설정

3. 돌연변이율 : 0.15(15%)

4. 돌연변이율 타입 : 'static', 세대를 거듭해도 돌연변이율이 변하지 않음

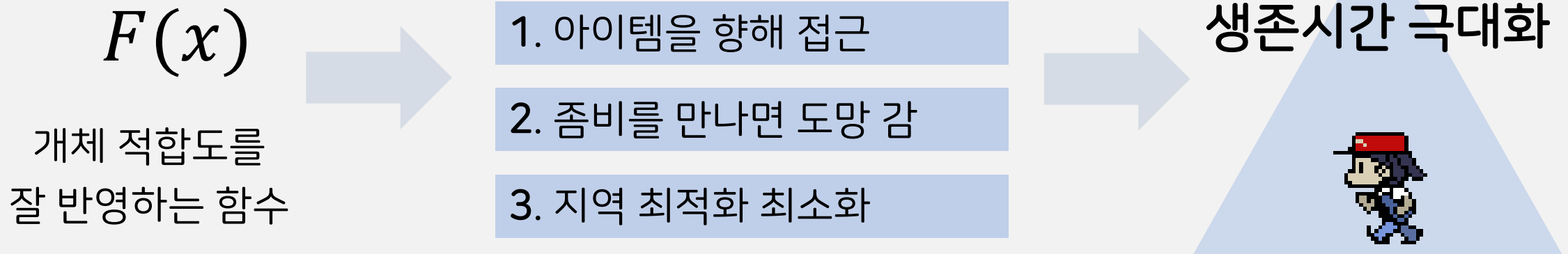
5. 돌연변이 발생 분포 방식 : Probability random uniform (-2, 2, mean=25, std=10)



2.4 핵심 연구/개발 과제

2. 플레이어 AI의 유전 알고리즘 개발

(2) 적합도 함수 설정



2.4 핵심 연구/개발 과제

2. 플레이어 AI의 유전 알고리즘 개발

(2) 적합도 함수 설정

목표

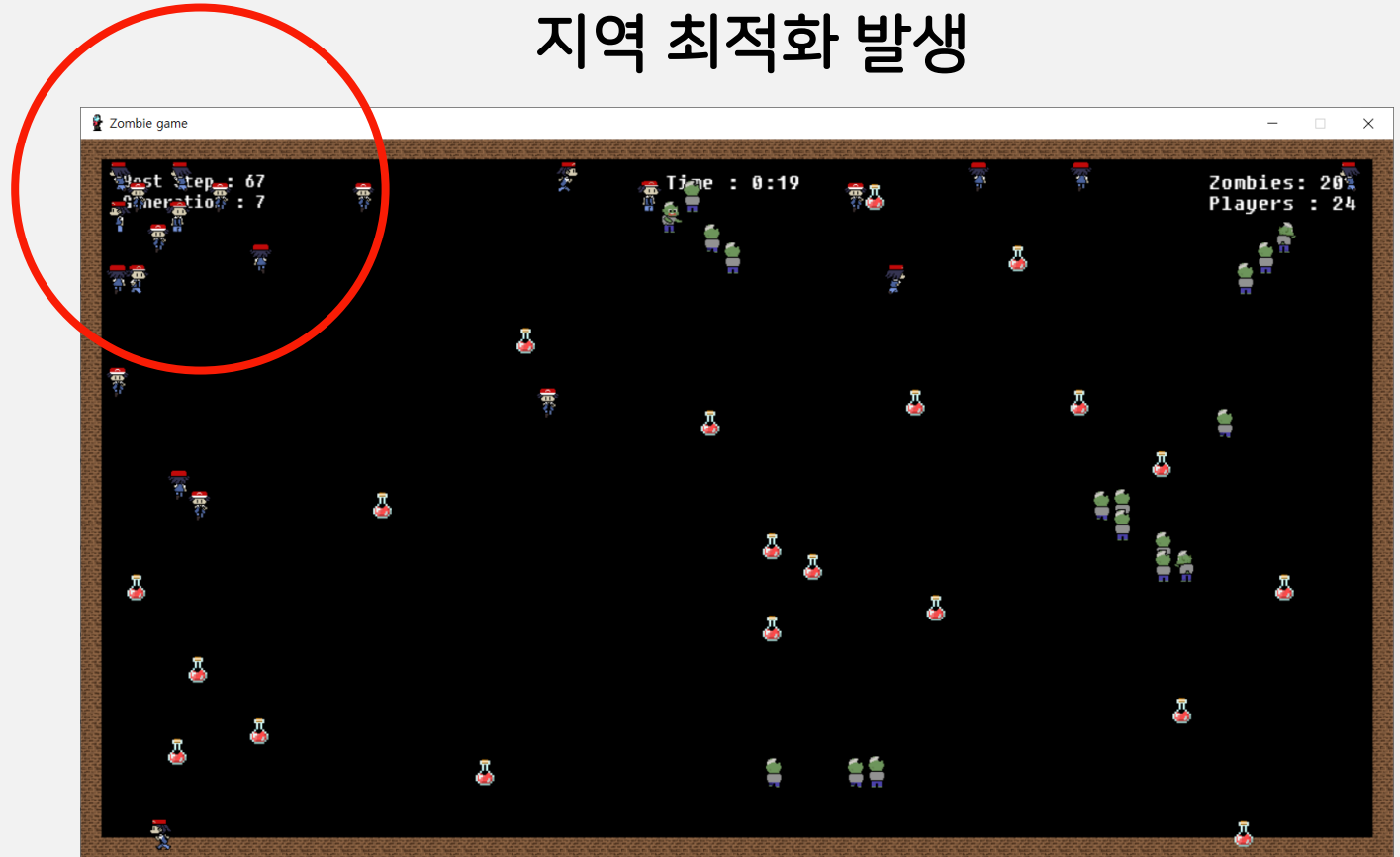
개체 적합도를 잘 반영하는
함수의 설정

문제점 발생

유전형질로 단순히 생존시간을
선택하면 의도했던 선택이
이루어지지 않음 →

좀비와 상관없이 운 좋게
오래 살아남은 개체들이
선택되는 문제 발생

지역 최적화 발생



2.4 핵심 연구/개발 과제

2. 플레이어 AI의 유전 알고리즘 개발

(2) 적합도 함수 설정

좀비에게 쫓기기 시작한 시점으로부터의 생존 시간을 기록하는
'target_step'을 구현함으로써 문제 해결

$$\text{Fitness} = \text{target_step} + (1 + \text{아이템 획득 개수} * 0.2)$$

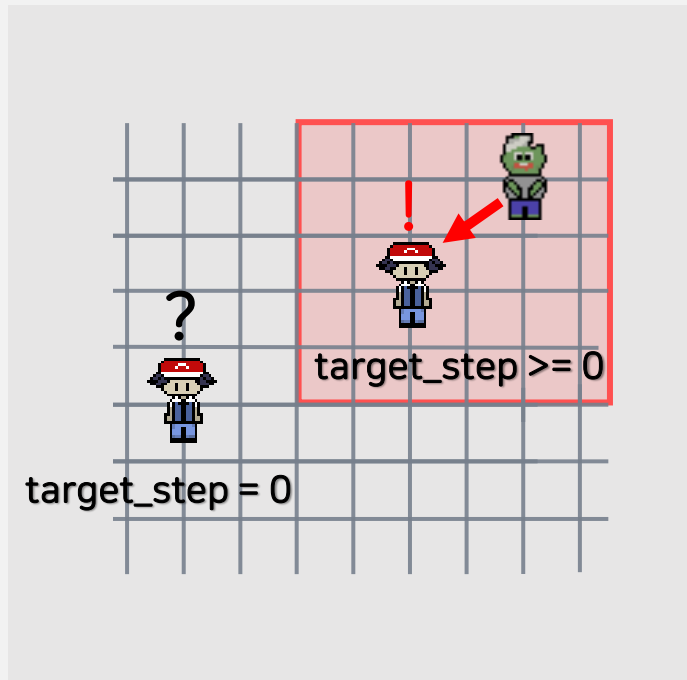
- 결과적으로 학습을 진행한 플레이어AI들은 좀비를 피해다니게 됨
- 모서리에서 맴도는 지역최적화 문제를 완벽히 해결하여 성능 개선함
- 아이템 가중치까지 추가하여 1+ 먹은 아이템 개수 * 0.2 추가

2.4 핵심 연구/개발 과제

2. 플레이어 AI의 유전 알고리즘 개발

(2) 적합도 함수 설정

target_step이란?



- 기본적으로 게임에 시각 센서 범위라는 개념이 존재하기 때문에 필요한 기능
- 좀비의 센서에 특정 플레이어가 포착된 시점을 기준으로 플레이어가 생존한 시간
- 좀비에게 포착되지 않은 플레이어는 시간이 지나면 step은 증가하나 target_step은 증가하지 않음
- 결과적으로 플레이어의 생존 시간이 좀비와 관련된 것인지, 좀비와 상관없이 운 좋게 살아남은 것인지 구별 가능하게 함

2.4 핵심 연구/개발 과제

2. 플레이어 AI의 유전 알고리즘 개발

(3) 최적의 하이퍼 파라미터 탐색

게임의 성능을 극대화할 수 있는 최적의 변수 값들을

수많은 시행착오를 거치며 경험적으로 도출

게임의 성능에
매우 큰 영향을
미쳤던 변인

- 시작 시 좀비의 생성 위치와 수

- 최대 좀비 제한 수

- 플레이어와 좀비의 시야 범위

- 부모와 자식의 개체 수 비율

- 좀비의 이동 속도

- 돌연변이율

- 맵의 크기

2.4 핵심 연구/개발 과제

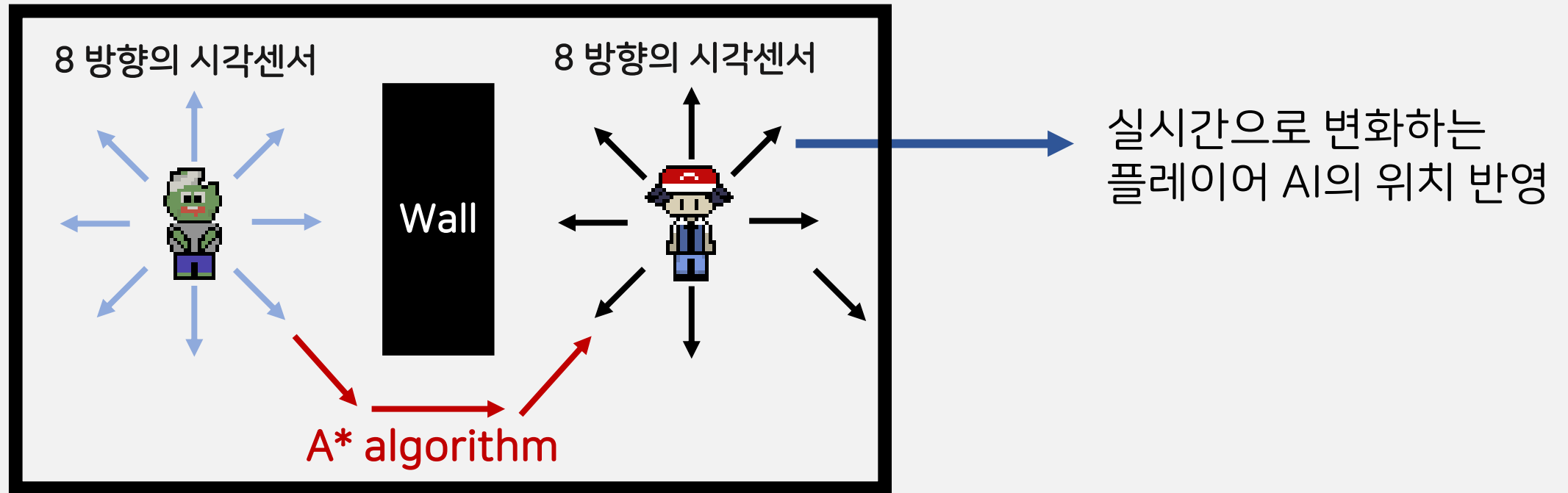
3. 학습용 총 4단계의 맵 구성

numpy 라이브러리를 이용해 벽 좌표 생성



2.4 핵심 연구/개발 과제

4. 좀비의 행동 알고리즘 개발

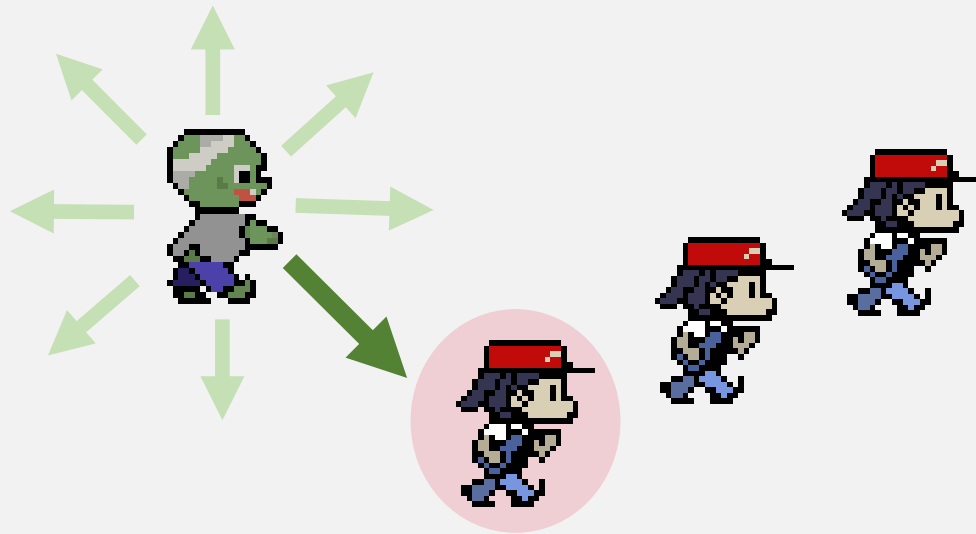


플레이어 AI를 포착하기 전까지는 랜덤 이동을 하다가

플레이어 AI를 포착하면 A* 알고리즘 을 이용해 최단경로로 추적

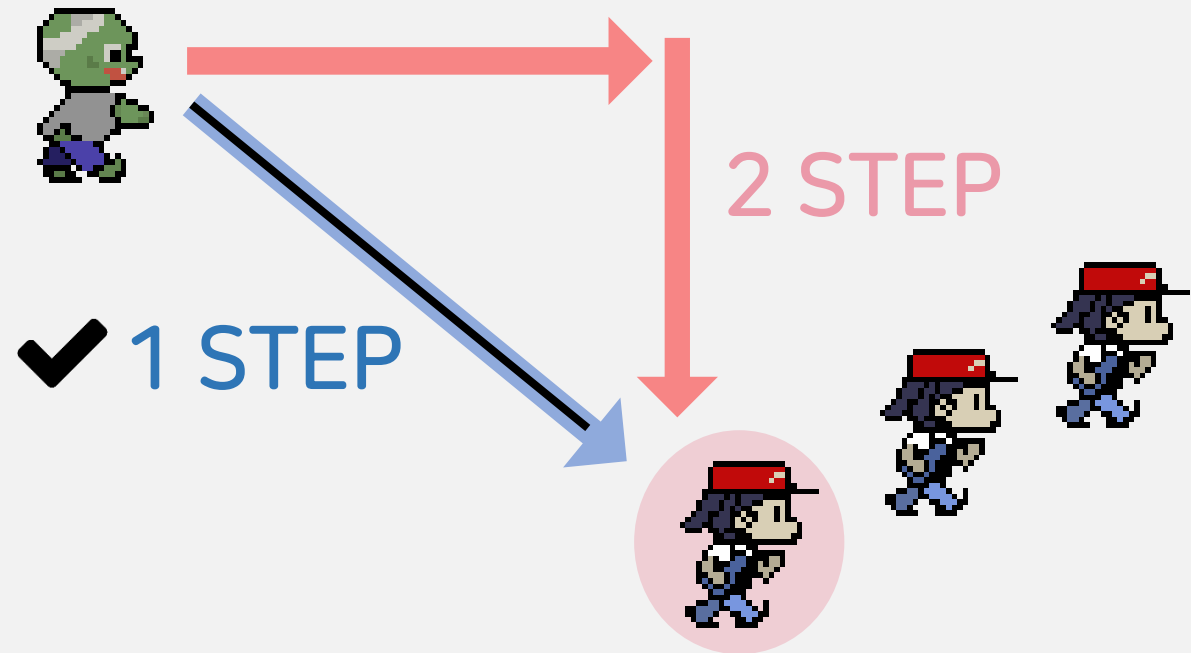
2.4 핵심 연구/개발 과제

4. 좀비의 행동 알고리즘 개발



가장 가까운 거리에 존재하는
플레이어 AI를 대상으로 포착

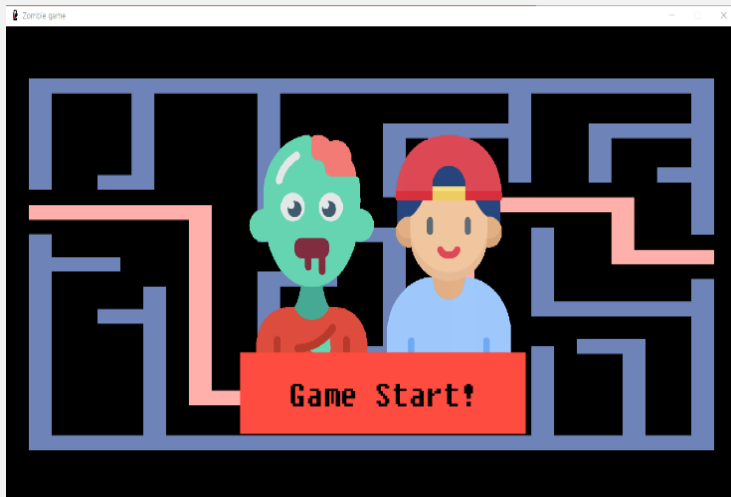
체비셰프 거리를 이용해
최소스텝으로 플레이어 AI에게 이동



2.4 핵심 연구/개발 과제

5. 게임 맵, 아이템, 좀비, 플레이어 AI를 그래픽으로 구현

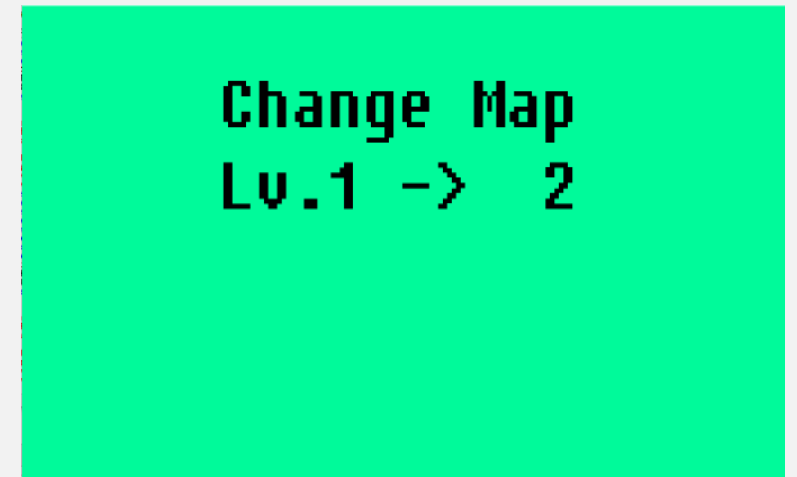
(1) 게임과 유사한 GUI 구성



게임 시작화면



세대 전환 시 출력 화면



맵 변경 시 출력 화면

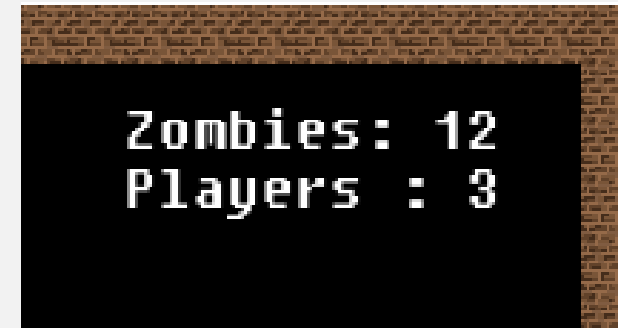
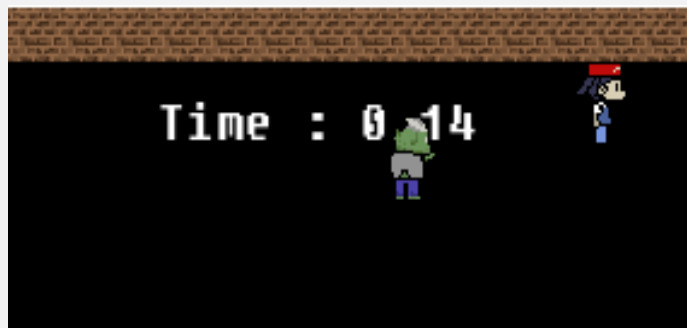
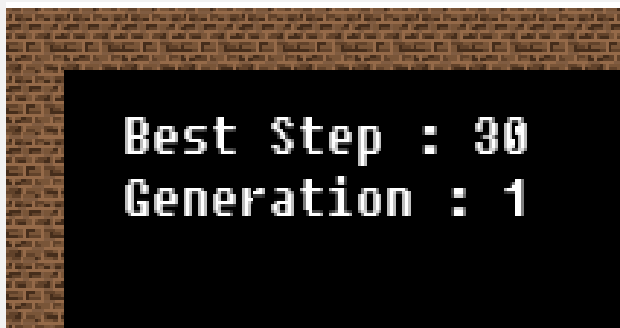
2.4 핵심 연구/개발 과제

5. 게임 맵, 아이템, 좀비, 플레이어 AI를 그래픽으로 구현

(1) 게임과 유사한 GUI 구성

- 현재 점수를 보여주는 전광판

Best Step, 세대, 게임진행 시간, 필드 내 좀비, 플레이어 수 출력



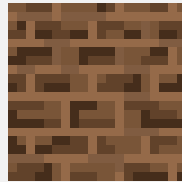
2.4 핵심 연구/개발 과제

5. 게임 맵, 아이템, 좀비, 플레이어 AI를 그래픽으로 구현

(2) 게임 내 정적 요소



아이템 (32x32)



벽 (20x20)



게임 필드 (64x35)

실제 게임 화면은

(게임필드 가로길이 x 벽 그래픽 가로 길이) x (게임필드 세로길이 x 벽 그래픽 세로 길이)

1280x700으로 설정하였다.

2.4 핵심 연구/개발 과제

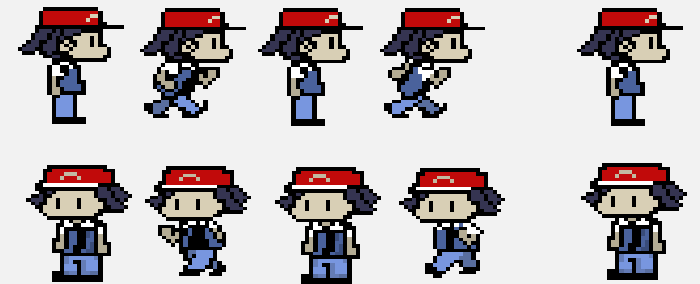
5. 게임 맵, 아이템, 좀비, 플레이어 AI를 그래픽으로 구현

(2) 게임 내 동적 요소

- 움직이는 플레이어/좀비 그래픽
- 플레이어/좀비는 총 8방향으로 움직인다
- 걷는 애니메이션을 자연스럽게 만들기 위한 최소 프레임은 3프레임이 필요
 - 8방향 x 3프레임 = 이미지 총 24
- 각 프레임을 0, 1, 2, 3으로 지정하고 클래스 변수로 인덱싱해 방향이 바뀌어도 자연스럽게 걷도록 함



좀비 애니메이션



플레이어 애니메이션

2.5 업무 분담/일정계획

● 업무분담



남태수

- 플레이어 AI 유전 알고리즘,
인공신경망 구현



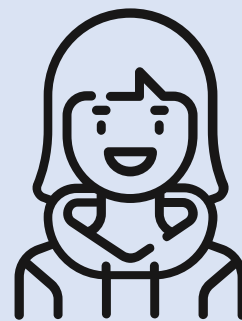
이유진

- 게임 GUI 개발



박지수

- 플레이어 AI 유전 알고리즘,
인공신경망 구현

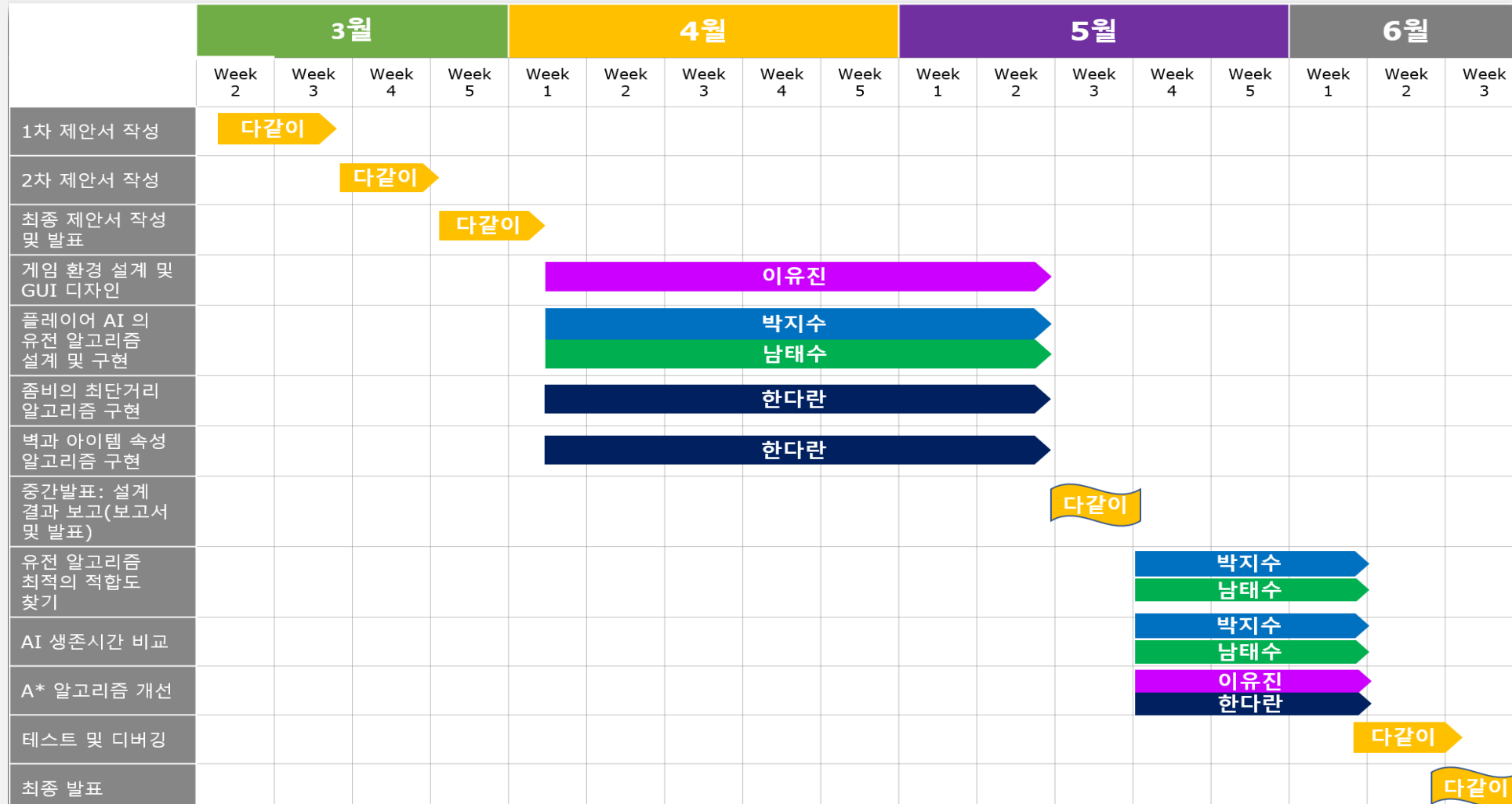


한다란

- 좀비의 최단거리 알고리즘 구현,
벽 및 아이템 속성 알고리즘 구현

2.5 업무 분담/일정계획

● 일정계획



2.6 개발 환경 및 구현



원활한 코드쉐어 / 일관성 유지



전반적인 개발 환경



GUI 구현 시 활용

3. 결론

3.1 달성 목표 및 성과

3.2 프로젝트의 의의

3.1 달성 목표 및 성과

01 세대를 거칠수록 플레이어 AI의 생존시간이 길어지는 유전 알고리즘 구현



02 단계별로 학습한 AI의 생존시간이 그렇지 않은 AI 생존시간의 130% 이상 달성



03 좀비가 장애물을 인지하며 플레이어 AI를 쫓는 최단거리 탐색 알고리즘 구현



04 게임 맵, 아이템, 좀비와 플레이어 AI를 그래픽으로 구현



3.1 달성 목표 및 성과

1. 세대를 거칠수록 플레이어AI의 생존시간이 길어지는 유전 알고리즘 구현 (100%)

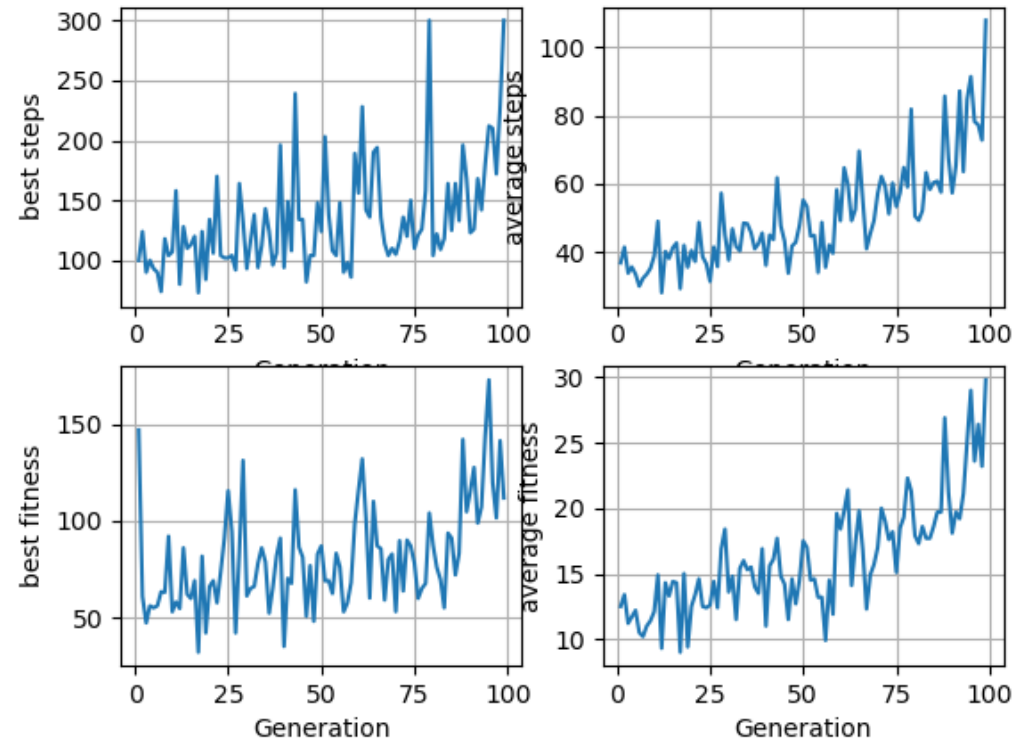
- 플레이어의 행동을 결정하는 다층인공신경망 구현
- 생존성이 좋은 플레이어를 선택하여 유전시키는 알고리즘 구현
- 결과적으로 1~10세대 평균 40 스텝에서, 100~110세대 평균 100 스텝까지 생존성 향상

3.1 달성 목표 및 성과

1. 세대를 거칠수록 플레이어AI의 생존시간이 길어지는 유전 알고리즘 구현 (100%)

- 그림의 우측 상단 그래프는 전체 플레이어들의 생존 시간의 평균을 보여줌
- 초기에 대략 40 스텝 정도에 불과한 생존 시간이 100세대에 도달하자 100스텝을 돌파함
- 이는 초기에 비해 100세대 만에 생존 성이 두배 이상 증가했다는 것을 보여줌

Step = 생존 시간

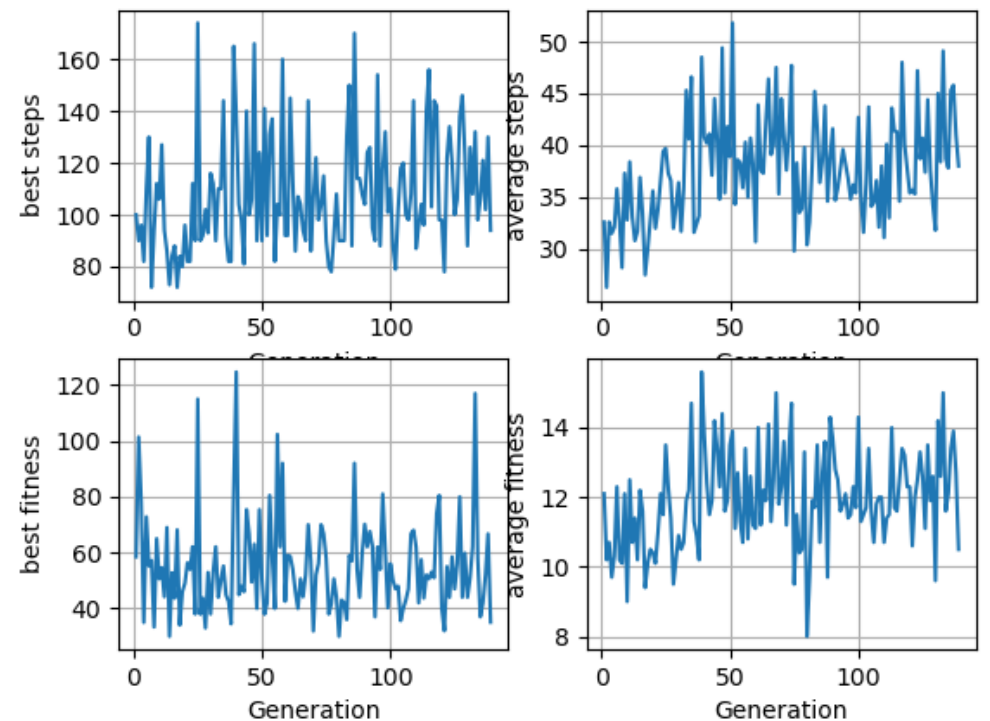


3.1 달성 목표 및 성과

2. 단계별로 학습한 AI의 생존시간이 그렇지 않은 AI 생존시간의 130% 이상 달성 (100%)

- 단계적 학습군과 그렇지 않은 대조군을 독립적인 환경에서 따로 학습을 진행

단계적 학습을 시키지 않은
Stupid 개체군의 테스트 맵에서의
학습 그래프

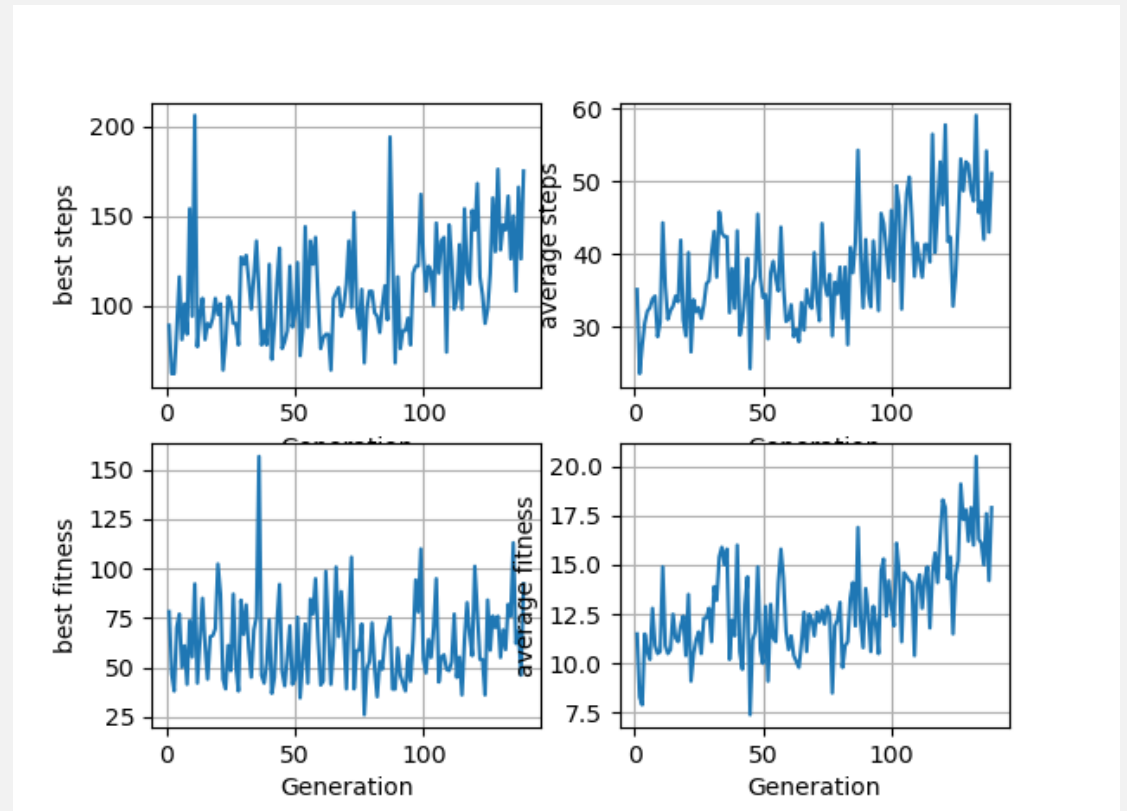


3.1 달성 목표 및 성과

2. 단계별로 학습한 AI의 생존시간이 그렇지 않은 AI 생존시간의 130% 이상 달성 (100%)

- 단계적 학습군과 그렇지 않은 대조군을 독립적인 환경에서 따로 학습을 진행

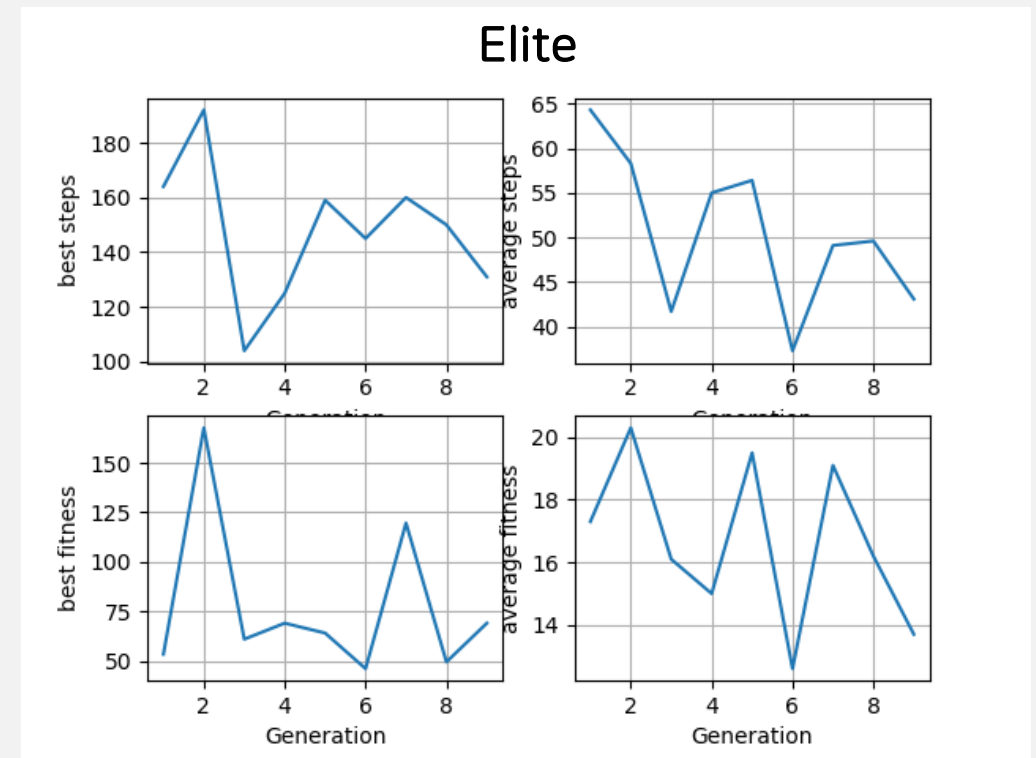
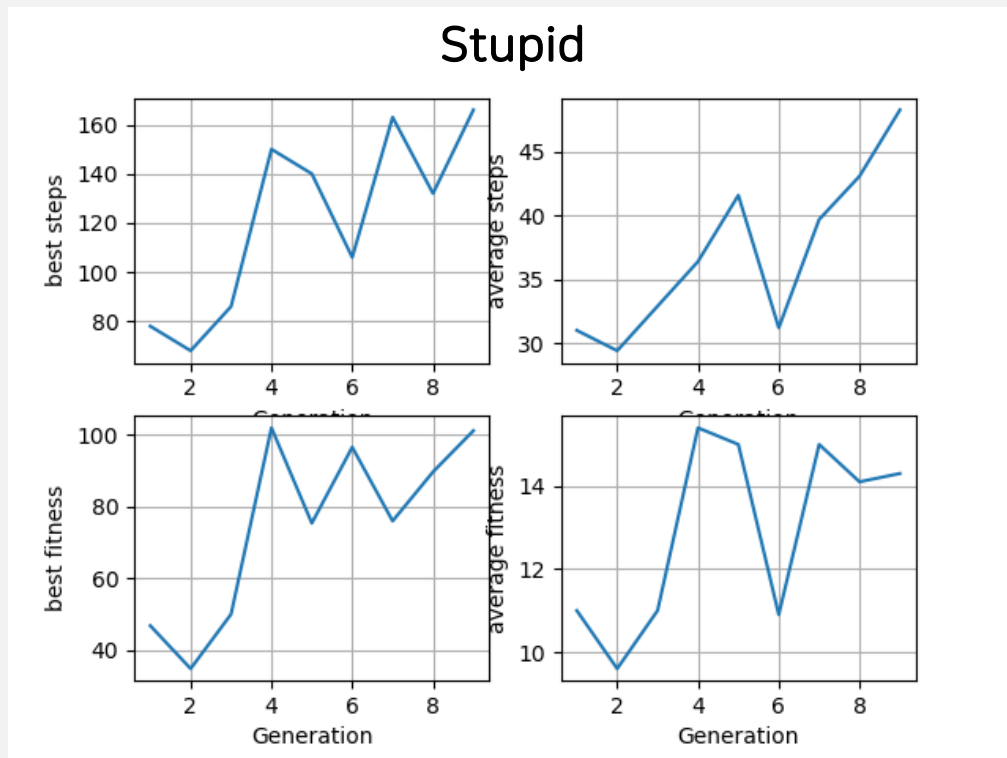
단계적 학습을 진행한
Elite 개체군의 테스트 맵에서의
학습 그래프



3.1 달성 목표 및 성과

2. 단계별로 학습한 AI의 생존시간이 그렇지 않은 AI 생존시간의 130% 이상 달성 (100%)

학습과 유전을 종료 시킨 후 테스트 맵 에서 10세대 동안의 생존 시간을 측정한 그래프 (10회 평균)



3.1 달성 목표 및 성과

2. 단계별로 학습한 AI의 생존시간이 그렇지 않은 AI 생존시간의 130% 이상 달성 (100%)
 - Stupid 그룹의 테스트 맵에서의 10 세대 평균 생존 시간은 37
 - Elite 그룹의 테스트 맵에서의 10세대 평균 생존 시간은 50

단계적 학습 과정을 도입한 결과 평균 생존 시간이 35% 증가 하였다.

3.1 달성 목표 및 성과

3. 좀비가 장애물을 인지하며 플레이어AI를 쫓는 최단거리 탐색 알고리즘 구현 (100%)

- A* 알고리즘을 사용한 최단경로 추적
- 벽을 인식하여 플레이어를 추적



3.1 달성 목표 및 성과

4. 게임 맵, 아이템, 좀비와 플레이어 시를 그래픽으로 구현(100%)

- 실제 게임과 유사한 GUI 구성
- 정적 요소 구현(벽, 아이템)
- 동적 요소 구현(플레이어, 좀비)
- 4단계 맵 제작과 테스트용 맵 2개 제작



좀비 애니메이션



플레이어 애니메이션

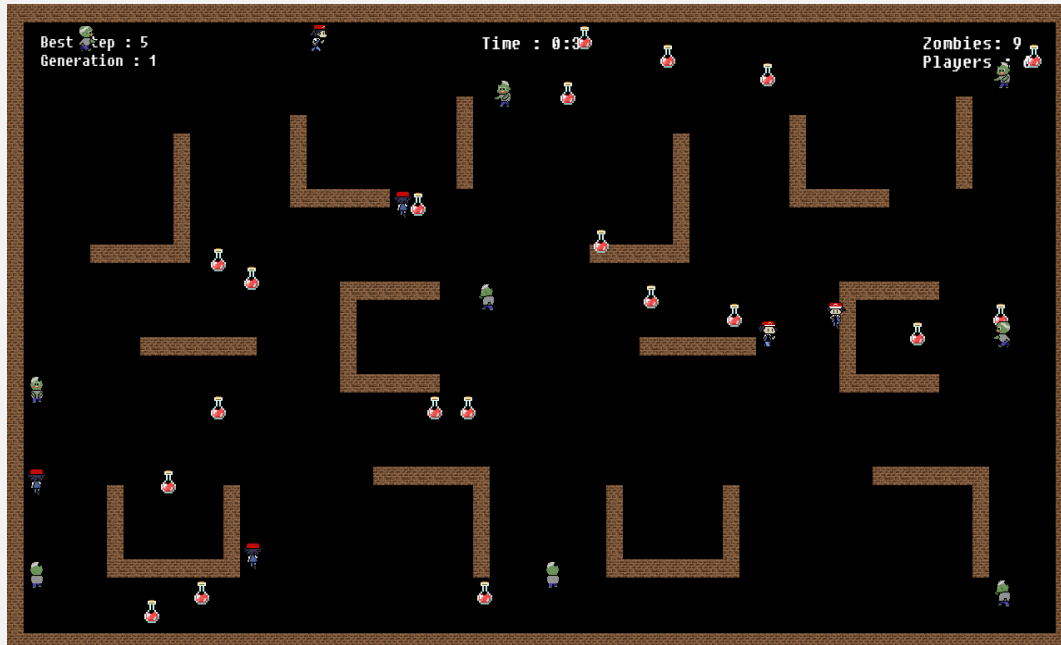


아이템

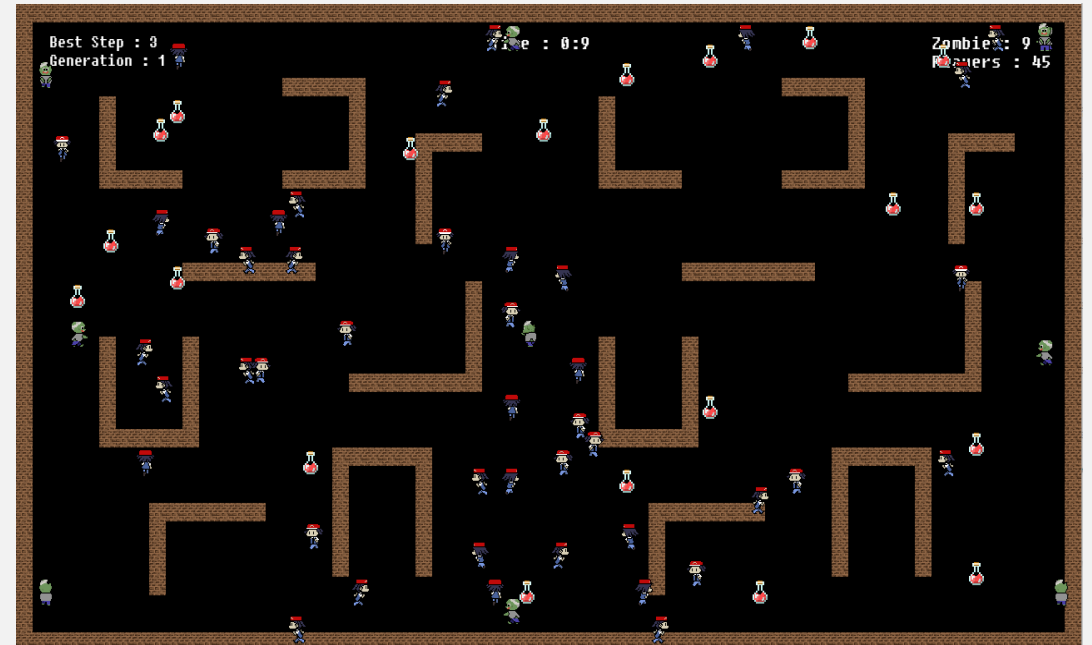
3.1 달성 목표 및 성과

4. 게임 맵, 아이템, 좀비와 플레이어 시를 그래픽으로 구현(100%)

[최종 테스트 맵]



[한가지 맵에서만 학습하는 플레이어의 훈련 맵]



3.1 달성 목표 및 성과

[1단계]



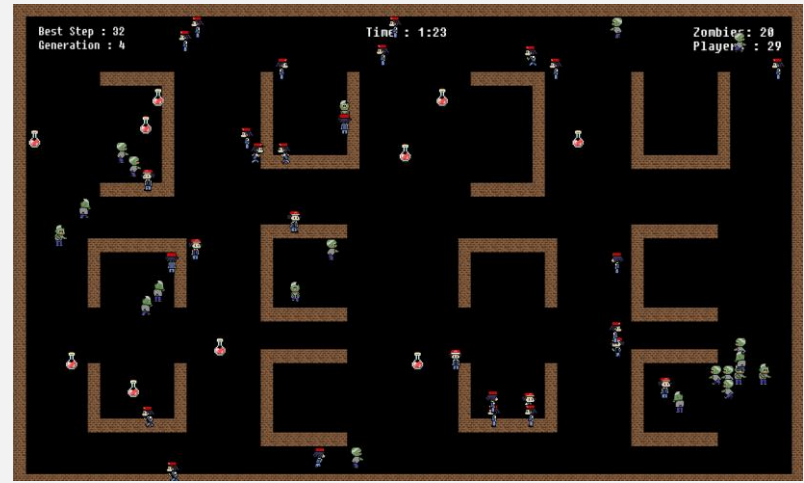
[2단계]



[3단계]



[4단계]



3.2 프로젝트의 의의

첫째, 유전 알고리즘과 인공지능망을 활용한 기존의 게임들은 대부분 문제에서 목표로 하는 것이 한 가지인 경우에 비해 해당 게임은 복수의 목표를 동시에 달성이 가능하게 완성되었다.

둘째, 인공지능망을 활용한 문제풀이에 학습단계의 환경이 점차 변하도록 하는 방법이 한가지 환경에서만 학습시켰을 때 보다 그 효과가 뛰어나다는 것을 증명하였다.

셋째, 유전 알고리즘과 인공지능망의 최적화 과정에서 얻은 경험적 데이터들이 추후 활용될 가능성이 있다.

감사합니다