

유전 알고리즘 파라미터 제어에 따른 게임 NPC의 자율 탐색 성능 및 수렴도 분석

이면재*

백석대학교 컴퓨터공학부 교수

Analysis of Autonomous Exploration Performance and Convergence of Game NPCs Based on Genetic Algorithm Parameter Control

MyounJae Lee*

Professor, Division of Computer Engineering, BaekSeok University

요약 유전 알고리즘(Genetic Algorithm)은 생물학적 자연선택과 진화 원리를 모방하여 최적해를 탐색하는 기계학습 기법으로, 주로 복잡한 최적화 문제나 인공지능 기반의 경로 탐색 등에 널리 활용된다. 본 연구는 해당 알고리즘을 적용하여 게임 내 비플레이어 캐릭터(NPC)의 경로 탐색 성능을 극대화하고, 그 수렴 과정 및 진화적 특성을 다양한 시나리오로 규명한다. 이를 위해 유니티 엔진 환경에서 목표 지점까지의 거리, 개체 생존 시간, 최종 도달 여부를 평가하는 적합도 함수를 구성하여 각 시나리오당 100세대에 이르는 시뮬레이션을 총 30회 반복하였다. 실험 결과, 세대 교체가 진행됨에 따라 목적지까지의 평균 및 최소 거리가 감소하며 안정적인 수렴 경향을 보였고, 목표에 성공적으로 도달하는 개체의 수가 꾸준히 증가하였다. 이를 통해 복잡한 지형 내 유전 알고리즘의 자율적 학습 효과를 실증하였으며, 향후 동적 파라미터 조절 기법을 적용한 적응형 모델로의 발전 가능성을 확인하였다.

주제어 : 유전 알고리즘, 최적화, 세대별 진화, 수렴 성능, 적합도 분석, 경로 탐색

Abstract Genetic Algorithm (GA) is a machine learning technique that searches for optimal solutions by imitating biological natural selection and evolutionary principles, and is widely utilized in complex optimization problems and AI-based pathfinding. This study maximizes the pathfinding performance of non-player characters (NPCs) in games by applying this algorithm, and investigates its convergence process and evolutionary characteristics through various scenarios. To this end, a fitness function evaluating the distance to the target point, individual survival time, and final arrival status was constructed in the Unity engine environment, and simulations of 100 generations per scenario were repeated 30 times in total. The experimental results showed that as generations progressed, the average and minimum distances to the destination decreased, showing a stable convergence trend, and the number of individuals successfully reaching the target steadily increased. Through this, the autonomous learning effect of the genetic algorithm in complex terrains was demonstrated, and the possibility of developing into an adaptive model applying dynamic parameter adjustment techniques in the future was confirmed.

Key Words : Genetic Algorithm (GA), Path Optimization, Generational Evolution, Convergence Analysis
Fitness Function

1. 서론

최근 게임 콘텐츠가 고도화됨에 따라 사용자들은 비플레이어 캐릭터(NPC)의 지능적 상호작용을 요구한다. 기존에 주로 활용된 유한 상태 기계(FSM)와 스크립트 기반 제어는 변화가 잦은 환경에서 사전 정의된 패턴만을 수행하는 단점이 존재한다[1]. 특히 실시간으로 지형 및 장애물이 변동하는 환경에서는 고정된 로직으로 NPC의 자율성을 보장하기 어렵다[2].

전통적 경로 탐색 기법인 A* 알고리즘이나 다익스트라(Dijkstra) 알고리즘은 정적 환경에서는 효율적이나, 이동 궤적이 급변하는 동적 상황에서는 연산 지연과 적응력의 한계를 보인다[3]. 이를 극복하기 위해 진화 연산의 일종인 유전 알고리즘(Genetic Algorithm, GA)을 NPC 인공지능에 적용하여 자가 학습을 유도하는 방식이 대안으로 떠오르고 있다[4]. 유전 알고리즘은 비정형 탐색 공간에서 전역 최적해를 찾는 데 유리하며[5], 목적에 맞는 적합도 함수 설계를 통해 NPC의 행동을 유연하게 제어할 수 있다[6]. 이러한 특성은 비선형적 게임 공간 내 NPC의 상황 판단력을 높이는 데 기여한다[7].

하지만 현재까지의 유전 알고리즘 연구는 주로 정적인 미로 탐색이나 고정된 장애물 회피에 머물러 있다[8]. 센서 데이터의 밀도와 진화 성능 간의 상관성 분석이 부족하며[9], 상용 엔진의 물리 시스템과 연동한 실시간 학습 연구 또한 미흡한 실정이다[10]. 이에 본 논문은 유니티(Unity) 2D 환경에서 레이캐스트(Raycast) 센서와 유전 알고리즘을 결합한 자율 제어 NPC 모델을 설계한다.

연구의 객관성 확보를 위해 동적 및 정적 장애물이 혼재된 '2D Beginner: Adventure Game'[11]을 실험 환경으로 문헌 [12]에서와 같이 채택하였다. 문헌 [12]의 연구에서는 수렴도를 검증하는 것에 집중하였으나, 본 연구에서는 탐색 성능과 수렴 속도에 핵심적인 영향을 미치는 개체군 크기(Population size), 우성인자 선택 비율(Selection rate), 돌연변이율(Mutation rate)을 독립 변인으로 설정하여 실험을 수행한다. NPC가 실시간 탐지를 통해 장애물을 회피하고 목표에 도달하는 과정을 분석하며, 특히 파라미터 조절이 최적 경로 수렴에 주는 효과를 규명한다. 본 연구는 NPC의 자율적 환경 적응력을 극대화하는 차세대 인공지능 설계 방향을 제시한다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련 연구를 고찰하고, 제3장에서는 유니티 엔진 기반의 유전 알고리즘 설계 및 구현 과정을 기술한다. 마지막으로 제4장에서는 실험 결과를 바탕으로 결론을 도출하고 향후

연구 과제를 제시한다.

2. 유전 알고리즘

유전 알고리즘은 생물학적 진화 과정인 적자생존과 돌연변이 현상을 모방하여 복잡한 문제의 최적해를 탐색하는 메타 휴리스틱(Meta-heuristic) 기법이다[2]. 이 알고리즘은 초기화, 적합도 평가, 선택, 교차, 돌연변이의 다섯 단계를 거치며 진화한다. 무작위로 생성된 개체군을 대상으로 문제 해결에 대한 적합도를 수치화한 후, 우수한 개체를 선택하여 유전자를 교환(교차)하고 임의의 변형(돌연변이)을 가하여 지역 최적해(Local Optima)에 빠지는 것을 방지한다[2]. 이러한 반복 연산은 수학적 분석이 어려운 비선형 문제 공간에서 강력한 탐색 성능을 발휘하며, 최근 비디오 게임 산업에서 콘텐츠의 절차적 생성(PCG) 및 동적 난이도 조절 시스템 등에 활발히 응용되고 있다[2].

특히 게임 내 NPC 제어 분야에서 유전 알고리즘의 도입이 두드러진다. 기존의 유한 상태 기계(FSM)와 같은 규칙 기반 인공지능은 동적 환경에서의 대응력 저하와 패턴의 단조로움이라는 한계가 존재했다[1]. 반면 유전 알고리즘을 적용한 NPC는 지형 변화나 동적 장애물에 대응하여 이동 속도와 방향을 최적화하고, 플레이어의 공격에 맞서 다양한 전술 시퀀스를 진화시킨다[1]. 더 나아가 개별 객체를 넘어 다수 NPC의 군집 이동 및 협력 기동까지 최적화할 수 있다[2].

그동안 유전 알고리즘은 축구 시뮬레이션 캐릭터 구현[13], 가변 환경 경로 탐색[14], 인공지능망 결합형 RPG 캐릭터 학습[15] 분야에 폭넓게 적용되었다. 나아가 최근 관련 연구는 적합도 함수를 이용한 기능성 게임 난이도 제어[16] 및 퍼즐 게임 최적 경로 산출[17] 등 동적 게임 환경에 대한 알고리즘의 적응 능력을 향상시키는 데 주력하고 있다. 이처럼 유전 알고리즘은 정해진 스크립트에 의존하지 않고 환경과 상호작용하며 자율적으로 행동 양식을 개선함으로써, 게임 환경의 사실성과 사용자의 몰입도를 크게 향상시키는 대안으로 자리 잡고 있다.

3. 유전 알고리즘 기반 시스템 구현

3.1 NPC의 유전자 구조 및 인코딩 (Genetic Encoding)

진화 연산의 탐색 성능은 개체의 유전자 설계 방식에 크게 의존한다. 문헌 [12]의 기존 유전자 구조는 환경 내 객체와의 거리를 유일한 변수로 삼아 [각도, 속도]를 결정하는 단순 배열 형태를 띤다. 이 구조는 기초적인 자율 제어에는 적합하나, 동적 장애물과 목적지가 혼재된 환경에서는 치명적인 진화적 병목을 유발한다. 본 연구는 이를 극복하고자 NPC의 유전자를 배회, 목표, 위험의 세 가지 독립된 형질로 분리하여 재설계하였다. 이러한 복합 구조체 방식은 알고리즘의 탐색 공간을 논리적으로 분할하여 유전자 간의 충돌을 방지하며, 결과적으로 복잡한 환경에서 NPC의 생존율과 목적 도달률을 동시에 향상시킨다. 각 인지 상태별로 독립된 행동 변수 쌍을 지니게 함으로써, 특정 상태(예: 위험 회피)의 유전자 변이가 다른 상태(예: 목표 접근)의 형질 보존에 악영향을 미치지 않는다. 따라서 본 제안 모델은 [12]의 구조적 한계를 원천적으로 차단하며, 복잡한 게임 환경에서도 NPC의 생존율과 목적 도달률을 동시에 고도화한다.

3.2 구현

본 연구는 자율형 NPC의 환경 정보를 즉각적인 행동으로 전환하여 최적해 탐색 속도를 가속화하는 핵심 단위로 Gene 구조체를 정의한다(〈Table 1〉). Gene 구조체는 장애물과의 거리에 따라 최적의 궤적을 생성하는 회피 각도(-180도~180도)와 생존율 극대화를 위해 가감 속을 조절하는 이동 속도 정보를 포함한다. 센서를 통해 측정된 정규화 거리(d)를 기반으로 해당 구조체가 즉각 호출되며, 추출된 유전자의 각도와 속도 값은 유니티 물리 엔진에 실시간 회전력 및 추진력으로 반영되어 NPC의 직관적인 회피 기동을 제어한다.

〈Table 1〉 Gene

```
public struct Gene{
    public float defaultAngle;
    public float defaultSpeed;
    public float goalAngle;
    public float goalSpeed;
    public float hazardAngle;
    public float hazardSpeed;
}
```

〈Table 2〉는 본 연구에서 제안하는 유전 알고리즘의 전체 진화 과정을 나타낸다. 단계 (1)에서는 무작위로 생성된 개체들로 초기 집단(P)을 구성한다. 단계 (2)에서는 사전에 정의된 종료 조건이 충족될 때까지 전체 진화 루

프를 반복한다. 단계 (3)에서는 현재 집단 내 모든 개체의 적합도를 계산하여 환경 적응 능력을 수치화한다. 단계 (4)에서는 다음 세대를 구성할 자식 개체들을 담기 위해 빈 리스트를 생성한다. 단계 (5)부터 (10)까지는 새로운 세대의 개체 수(size(Next_P))가 현재 세대의 개체 수(size(P))와 동일해질 때까지 반복하는 자식 생성 루프이다. 단계 (6)에서는 토너먼트 선택 기법을 통해 현재 집단에서 우수한 적합도를 가진 두 부모 개체를 추출한다. 단계 (7)에서는 설정된 교차율에 따라 두 부모의 유전자를 결합하여 새로운 자식 개체를 생성한다. 단계 (8)에서는 설정된 변이율에 따라 자식 개체의 유전자 일부를 무작위로 변경하여 탐색 과정 중 지역 최적해에 빠지는 현상을 방지한다. 단계 (9)에서는 연산이 완료된 자식 개체를 다음 세대 리스트에 추가한다. 자식 생성 루프가 종료된 후, 단계 (11)에서는 기존 세대를 새롭게 구성된 다음 세대로 완전히 교체한다. 종료 조건은 제시된 세션 시간이다.

〈Table 2〉 GA Algorithm

```
01:P ← InitializeRandomPopulation()
02:WHILE termination condition is not met DO
03: EvaluateFitness(P)
04: Next_P ← Empty List
05: WHILE size(Next_P) < size(P) DO
06:   P1(Parent1), P2(Parent2) ← TournamentSelect(P)
07:   Child1,Child2← Crossover(P1, P2, CrossOverRate)
08:   Child1, Child2 ← Mutate(Child1, Child2, MutationRate)
09:   Next_P.Add(Child1, Child2)
10: END WHILE
11: P ← Next_P
12: END WHILE
13: RETURN P
```

〈Table 3〉은 적합도 함수를 보여준다. 제안하는 적합도 평가 함수는 개체의 최종 상태에 따라 차등적인 보상을 부여한다. 목표에 도달한 개체는 기본 점수 10점과 시간 단축에 따른 가점을 받아 가장 높은 적합도를 지닌다(단계 (2)-(3)). 반면, 데미지 존과 충돌하여 사망한 개체는 최소 점수를 부여받아 다음 세대에서 도태된다(단계 (4)-(5)). 목표에 도달하지 못했으나 생존한 개체는 사망 개체보다 높은 5점의 기본 점수를 확보한다(단계 (6)-(8)). 이때 에이전트가 안전 구역에만 머무르는 지역 최적해 현상을 방지하고 목적지 탐색을 적극적으로 유도하기 위해, 목표 접근성과 위험 회피성을 0.7 대 0.3의 비대칭 가중치로 합산하여 최종 적합도를 산출한다.

〈Table 3〉 Fitness of Agent

```

PROCEDURE EvaluateFitness(agent)
01:  $t \leftarrow \text{agent.survivalTime} / \text{maxSessionTime}$ 
02: IF State(agent) == SUCCESS THEN
03:   RETURN  $10.0 + 5.0 * (1.0 - t)$ 
04: IF State(agent) == DEAD THEN
05:   RETURN  $0.001 + 0.1 * t$ 
06:  $G \leftarrow \text{Max}(0, 1.0 - (\text{Distance}(\text{agent}, \text{Goal}) / \text{MaxDistance}))$ 
07:  $H \leftarrow \text{Min}(1.0, \text{Distance}(\text{agent}, \text{NearestHazard}) / \text{SafeRadius})$ 
08:   RETURN  $5.0 + 0.7 * G + 0.3 * H$ 
END PROCEDURE

```

〈Table 4〉는 환경 인지 기반 에이전트의 행동 유전자 선택 절차를 보여준다. 데미지(목표) 존과 같은 위험(성공) 요소가 발견되면 거리에 따른 유전자 인덱스를 계산하여 데미지 존(목표)을 회피(도달)하도록 Gene을 변경한다.

〈Table 4〉 Gene Action Selection Algorithm based on Agent's Environment Recognition

```

PROCEDURE SelectGeneBehavior
01:   IF Hazard is Detected THEN
02:     Calculate index based on hazard distance
03:     Set target to Hazard Avoidance Gene[index]
04:   ELSE IF Goal is Detected THEN
05:     Calculate index based on goal distance
06:     Set target to Goal Approach Gene[index]
07:   ELSE
08:     Set target to Default Gene[middle]
09:   END IF
END PROCEDURE

```

3.3 결과

본 연구의 실험은 Intel Core i7, 16GB RAM, 유니티(Unity) 6.4 버전 환경에서 진행되었다. 유전 알고리즘의 성능을 객관적으로 평가하기 위해서는 각 파라미터가 탐색과 수렴에 미치는 상충 관계를 증명해야 한다. 이를 위해 화면 크기를 고려하여 개체군 크기가 40 이하로 제한된 환경에서는 유전적 다양성이 초기에 소실되어 지역 최적해에 빠질 위험이 크다. 따라서 세션 시간의 제약과 개체군의 크기 한계를 고려하여, 다음 세 가지 유형으로 실험군을 구성한다.

〈Table 5〉는 실험 파라미터를 보여준다. 기준 집단인 그룹 A는 연구[18]에 기반하여 유전적 다양성과 수렴 속도 간의 표준적인 균형을 유지하도록 설정되었다. 둘째, 고속 수렴을 목적으로 하는 그룹 Group B는 연구 [19]

에서 제시된 파라미터를 차용하여, 높은 선택률을 통해 현재 세대의 우수 해를 빠르게 활용하도록 구성했다. 마지막으로 장기 탐색을 위한 그룹 C는 [20,21] 연구를 참조하여 선택률을 낮추고 돌연변이율을 극대화함으로써 조기 수렴을 방지하고 전역 최적해 탐색 능력을 높이도록 설계하였다. 이 알고리즘의 수렴과정은 연구[12]에서 제시되었다.

〈Table 5〉 Experimental Parameter

Group	Population	Selection rate	Mutation rate
A	40	0.8	0.05
B	20	0.9	0.1
C	40	0.5	0.15

각 실험군은 세션 시간 종료 시점에서의 목표 도착 개수(성공 개수), 데미지존에 도착한 개수(죽은 개수), 두 경우를 제외한 에이전트중에서 목표 지점까지의 가장 짧은 거리, 평균 거리로 성능을 비교 분석한다. 무작위성에 의한 오차를 최소화하기 위해 각 실험군은 30회 독립 수행을 거쳐 평균을 구한다. 세션시간 즉 1 세대의 탐색시간은 60초로 설정한다. [Fig. 1]은 시작점과 도착점 그리고 데미지 존, 나무 등의 장애물을 보여준다. 데미지 존을 목표 지점보다 앞에 위치 시켜서 세대수가 거듭될수록 에이전트가 데미지 존을 피해가는 여부를 알수 있도록 하였다.

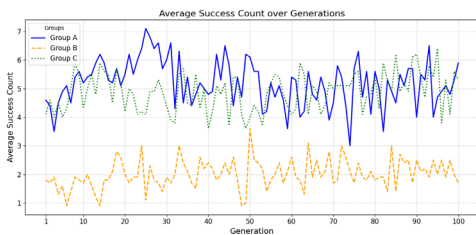


[Fig. 1] Starting Point, Target Point, Damage Zone

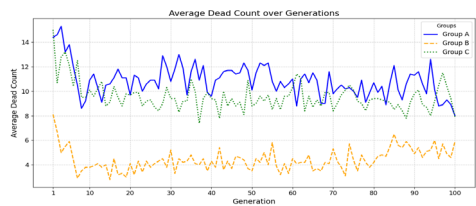
[Fig. 2]는 그룹별 성공 횟수의 평균으로 에이전트가 타겟 지점에 도달한 개수이다. 그룹 A와 C는 100세대 도달 시 높은 성공 횟수(각각 5.9, 5.3)를 기록한 반면, 그룹 B는 1.7에 머물렀다. 이는 그룹 B가 우수 개체의 유전자를 빠르게 교차시켜 수렴 속도를 높이려 했으나, 돌연변이율이 지나치게 낮아 새로운 탐색 경로를 찾지 못하고 지역 최적해에 갇혔기 때문이다. 반면 그룹 C는 초반에는 성공 횟수가 변동성을 보이지만 높은 돌연변이

을 통해 다양한 유전자를 탐색함으로써 후반부(100세대 인근)에 이르러 그룹 A에 근접하는 우수한 전역 최적해 도달 능력을 보여준다.

[Fig. 3]은 그룹별 죽은 개수 비교이다. 즉 데미지 존에 도착한 경우이다. 그룹 A와 C는 전 세대에 걸쳐 뚜렷한 변동성을 보이며 약 8~14 구간의 비교적 높은 실패 개체 수를 유지한다. 이러한 현상은 두 그룹이 높은 돌연변이율과 교차율을 통해 탐색 공간 내 해의 다양성을 지속적으로 확보하려는 특성에 기인한다. 한편, 세대가 증가함에 따라 두 그룹 모두 실패 개체 수가 감소하는 추세를 보인다. 반면 그룹 B는 전 구간에서 가장 낮은 수준의 실패 개체 수(약 3~6 구간)를 기록한다. 이는 알고리즘이 제한된 유전 연산만을 수행하여 개체의 도태 비율을 물리적으로 낮추었으나, 결과적으로 탐색 범위가 축소되어 지역 최적해에 조기 수렴하는 원인으로 작용하였음을 시사한다.



[Fig. 2] Average Success Count



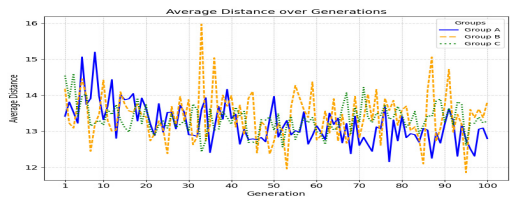
[Fig. 3] Average Dead Count

[Fig. 4]는 그룹별 최단 거리 비교이다. 최단 거리 평균은 에이전트가 목표 지점에 얼마나 가깝게 도달했는지를 나타내는 핵심 최적화 지표이다. 그룹 A와 C는 세대가 진행될수록 최단 거리가 3.0~5.0 구간에서 안정적인 변동폭을 보이며 하향하는 양상을 나타낸다. 이는 두 그룹이 목표지점에 근접하는 최적의 경로를 성공적으로 탐색하고 있음을 의미한다. 그룹 B는 세대가 거듭되어도 최단거리가 5.0~7.5 구간에 머물러 다른 그룹보다 목표 지점과의 거리가 멀게 나타난다 [1]. 이는 해당 알고리즘이 지역 최적해(Local Optima)에 조기 수렴하여 목표 지점까지의 거리를 효과적으로 단축하지 못함을 보여준다.



[Fig. 4] Average Shortest Distance

[Fig. 5]는 평균 거리이다. 세 그룹 모두 평균거리는 12.0~15.0 사이의 구간에 분포하며 세대별로 변동을 반복한다. 그룹 A와 C는 비교적 일정한 변동폭을 보이며 안정적인 탐색 반경을 유지한다. 반면에 그룹 B는 특정 세대(예: 30세대, 80세대 인근)에서 평균 거리가 15.0을 초과하는 등 변동폭이 극심하고 불안정한 모습을 나타낸다. 이는 알고리즘이 안정적인 군집 행동을 형성하지 못하고 탐색 궤적이 무의미하게 흩어지고 있음을 시사한다. 결론적으로 적절한 우성 유전자 선택이 보장된 그룹 A와 C가 효율적이고 안정적인 경로 최적화를 수행하는 반면, 다양성이 제한된 그룹 B는 경로 최적화 및 군집 안정성에 뚜렷한 한계를 보인다.



[Fig. 5] Average Distance

4. 결론 및 추후 연구방향

본 연구의 목적은 유전 알고리즘을 활용하여 장애물 충돌을 회피하고 최단 거리를 탐색하는 최적의 진화 전략을 도출하는 것이다. 이를 위해 교차율과 돌연변이율을 달리 설정한 세 가지 실험군(그룹 A, B, C)을 구성하여 100세대에 걸친 진화 과정과 성능 지표를 분석하였다.

실험 결과, 균형 잡힌 연산을 수행한 기준군(Group A)과 돌연변이율을 높인 장기 탐색군(Group C)은 세대가 거듭될수록 최단 거리를 갱신하며 높은 평균 성공 횟수(100세대 기준 각각 5.9, 5.3)에 안정적으로 도달하였

다. 반면, 교차율을 극대화한 그룹 B는 낮은 돌연변이율로 인해 탐색의 다양성이 상실되어 지역 최적해에 조기 수렴하는 한계를 보였다.

향후에는 유전 알고리즘의 진화 과정에 강화학습(Reinforcement Learning)의 에피소드 보상 체계를 결합한 하이브리드 탐색 기법을 제안하여, 다양성 확보를 위해 소요되는 초기 세대 누적 시간 및 연산 비용을 단축하는 방안을 모색한다.

REFERENCES

- [1] I. Millington and J. Funge, AI for Genetic Algorithms, 2nd ed., Burlington, MA: CRC Press, 2009.
- [2] G. N. Yannakakis and J. Togelius, Artificial Intelligence and Genetic Algorithms, New York, NY: Springer, 2018.
- [3] Sturtevant, N. R., Benchmarks for Grid-Based Pathfinding. IEEE Transactions on Computational Intelligence and AI in Genetic Algorithms, 4(2), 144-148, 2012.
- [4] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Reading, MA: Addison-Wesley, 1989.
- [5] M. Mitchell, An Introduction to Genetic Algorithms, Cambridge, MA: MIT Press, 1998.
- [6] M. Buckland, AI Techniques for Genetic Algorithm Programming, Cincinnati, OH: Premier Press, 2002.
- [7] K. O. Stanley and R. Miikkulainen, "Evolving Neural Networks through Augmenting Topologies," Evolutionary Computation, vol. 10, no. 2, pp. 99-127, Jun. 2002.
- [8] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-Based Procedural Content Generation: A Taxonomy and Survey," IEEE Transactions on Computational Intelligence and AI in Genetic Algorithms, vol. 3, no. 3, pp. 172-186, Sep. 2011.
- [9] S. M. Lucas, "Computational Intelligence and Genetic Algorithms," in Proceedings of the 2008 IEEE Conference on Computational Intelligence and Genetic Algorithms (CIG), Perth, Australia, pp. 21-28, 2008.
- [10] J. K. Haas, "A History of the Unity Genetic Algorithm Engine," M.S. thesis, Dept. Interact. Media Genetic Algorithm Develop., Worcester Polytechnic Inst., Worcester, MA, 2014.
- [11] Unity Technologies. 2D Beginner: Adventure Game. Unity Learn, 2020.
- [12] MyounJae Lee, "Performance and Evolutionary Characteristics of Genetic Algorithm-based Optimal Pathfinding for Genetic Algorithm NPCs in Unity Engine," Journal of KIOTS, Vol.12, No.3, pp.61-66, 2026.
- [13] H.S. No and D.W. Lee, "A Study on The Game Character Creation Using Genetic Algorithm in Football Simulation Games," Journal of Korea Game Society, 17(6), pp.129 - 138, 2017.
- [14] J.W. Koh and D.Y Lee, "Path-finding Algorithm using Heuristic-based Genetic Algorithm", Journal of Korea Game Society, vol.17, No.5, pp.123-132, 2017.
- [15] O.K. Kwon and J.K. Park, "Control of RPG Game Characters using Genetic Algorithm and Neural Network", Journal of Korea Game Society, vol.6, No.2, pp.13-22, 2006.
- [16] V. Vaghani and K. Oyibo, "A Scoping Review of Genetic Algorithms in Serious Games: Applications, Challenges, and Future Directions," Preprints.org, 2024.
- [17] N. H. S. Hasibuan et al., "Using Genetic Algorithm to Solve Puzzle Games: A Review," Journal of Computer Networks Architecture and High Performance Computing, 2024.
- [18] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. Thesis, University of Michigan, Ann Arbor, 1975.
- [19] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," IEEE Transactions on Systems, Man, and Cybernetics, vol. 16, no. 1, pp. 122-128, 1986.
- [20] A. E. Eiben and C. A. Schippers, "On evolutionary exploration and exploitation," Fundamenta Informaticae, vol. 35, no. 1-4, pp. 35-50, 1998.
- [21] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri, and V. B. S. Prasath, "Choosing mutation and crossover ratios for genetic algorithms—A review with a new dynamic approach," Information, vol. 10, no. 12, p. 390, 2019.

이 먼 재(MyounJae Lee)

[종신회원]



■ 2009년 3월 ~ 현재 : 백석대학교
컴퓨터공학부 교수

<관심분야>

사물인터넷, 게임, MPEG