

게임 맵 테스트를 위한 구조적 맵 커버리지

(Structural map coverage for game map testing)

이정준*, 송지영**

(Jeong Jun Lee, Jiyoung Song)

요약

게임의 시각적 요소는 플레이어의 만족도를 높이는 데 중요한 역할을 하며, 특히 맵 디자인은 필수적인 평가 기준으로 작용한다. 이에 개발자들은 몰입도를 높이고, 오류가 없는 최적화된 맵을 구현해야 한다. 최근 게임 맵 크기가 증가하면서, 맵 디자인과 테스트의 복잡성도 커지고 있다. 대규모 맵에서는 크래시나 맵 이탈 같은 오류가 발생할 위험이 크다. 이러한 오류는 플레이어의 몰입감을 저해하고 만족도를 감소시킬 수 있다. 따라서 효과적인 맵 테스트는 플레이어 만족도를 높이기 위해 필수적이다. 본 연구에서는 구조적 그리드 커버리지 기법을 제안하여 맵 테스트의 효과를 분석하고, 효과적인 적용을 제시한다. 구조적 그리드는 맵을 다양한 크기의 그리드로 쪼개어 에이전트가 이동한 그리드의 비율을 측정하는 방법이다. 우리는 구조적 그리드 커버리지를 활용하여 플레이어의 시야와 이동 경로 등을 분석하는 것을 실험적으로 보였다. 실험 결과, 적절한 그리드 크기를 선택하면 오류 검출 정확도가 약 36% 증가함을 실험적으로 확인하였다.

■ 중심어 : 게임 맵 테스트 ; 맵 이탈 오류 ; 구조적 그리드 커버리지 ; 그리드 크기

Abstract

The visual elements of a game play an important role in increasing player satisfaction, and map design in particular serves as an essential evaluation criterion. Developers must implement optimized maps that increase immersion and are free of errors. Recently, as the map sizes of games have increased, the complexity of map design and testing has also increased. In large-scale maps, errors such as crashes or map departures are more likely to occur. These errors can hinder player immersion and reduce satisfaction. Therefore, effective map testing is essential to increasing player satisfaction. In this study, we propose a structural grid coverage technique to analyze the effectiveness of map testing and present effective applications. We experimentally show that we can analyze players' field of view and movement paths using structural grid coverage. The experimental results show that selecting an appropriate grid size can simultaneously improve error detection and player satisfaction.

■ keywords : game map testing ; map exit errors ; structural grid coverage ; grid size

I. 서론

게임의 시각적 요소는 플레이어들의 만족도를 높이는 데 중요한 역할을 한다[1]. 게임의 시각적 요소 중에서도 맵 디자인은 다양한 평가 기준에 필수적으로 포함된다[2,3]. 맵은 단순한 배경 이상의 의미가 있으며, 플레이어가 게임 세계를 탐

험하고 상호작용하는 공간을 제공한다. 따라서, 사용자 연구를 통해 플레이어의 선호와 행동을 이해하고, 이를 바탕으로 최적화된 맵 디자인을 구현하는 것이 중요하다. 예를 들어, 이세연 외의 연구[2]가 제시하는 게임 디자인 가이드는 플레이어의 만족도를 고려한 요소를 포함한다. 요소들의 대표적인 예로 플레이어 외향과 목소리, 배경 사운드, 맵 디자인 등이 존재한다. 디자인 요

* 준회원, 한남대학교 컴퓨터공학과

** 중신회원, 한남대학교 컴퓨터공학과

이 논문은 2024학년도 한남대학교 학술연구비 지원에 의하여 연구되었음

접수일자 : 2024년 10월 21일

수정일자 : 2024년 11월 18일

재확정일 : 2024년 11월 20일

교신저자 : 송지영 e-mail : jysong@hnu.kr

소들의 대부분은 인터페이스와 관련되어있고, 맵은 인터페이스에서 가장 큰 비중을 차지한다.

최근 게임 산업에서는 맵 디자인의 크기가 점점 증가하는 추세이다. 게임 맵의 규모가 커진다는 것은 플레이어의 다양한 경험을 유도할 수 있다는 장점이 있는 반면, 맵 관련 에러가 발생할 수 있는 면적이 넓어지며 에러가 발생할 확률이 높아질 수 있다. 즉, 맵의 크기가 커짐에 따라 게임 맵 테스트의 난이도가 높아진다.

맵 테스트가 제대로 이루어지지 않을 경우, 맵 관련 오류가 발생할 수 있다[4]. 맵상에서 발생하는 에러에는 에이전트가 움직일 수 없는 맵 간헐 에러(stuck-in-map error), 에이전트가 게임 맵 밖으로 이탈하는 맵 이탈 에러, 게임맵의 그래픽이 깨지는 맵 크래시 등이 있다. 이러한 오류는 플레이어의 경험을 저해하며, 게임의 완성도를 떨어뜨린다. 맵의 오류는 플레이어의 만족도에 큰 영향을 미칠 수 있으므로 게임 맵 테스트를 통해 오류를 줄여야 한다. 그러나, 게임 맵 오류들은 테스터들에게 크게 의존하고 있어 체계적인 맵 오류 검출이 어렵다.

본 연구에서는 맵 테스트를 구조적으로 분석하고, 체계적으로 수행할 수 있도록 구조적 맵 커버리지인 그리드 커버리지(Grid coverage)를 제안하고 적용한다. 이 커버리지는 맵 테스트 시 플레이어의 시야, 이동 경로 등의 커버리지를 시각적으로 표기하여, 맵의 각 요소가 얼마나 잘 테스트 되었는지를 평가한다. 우리는 실험에서 다양한 크기의 그리드를 맵 커버리지를 적용하여, 테스트가 이루어지지 않은 영역을 식별했다. 결과적으로 그리드의 크기에 따라 게임 맵 분석이 달라짐을 확인했다. 실험 결과를 기반으로 적절한 그리드 크기로 테스트를 수행하면, 게임 맵상의 오류를 효과적으로 검출하고, 플레이어의 만족도를 높이는 데 기여할 수 있을 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 게임 맵과 관련 있는 연구를 소개한다. 3장에서는 본 연구가 제안하는 구조적 게임 맵 커버리지를 설

명하고, 4장에서는 커버리지의 적용 및 실험을 보여준다. 마지막으로 5장에서 본 연구의 결론, 기대효과 및 향후 연구를 서술한다.

II. 관련 연구

기존 관점에서의 맵은 주체적인 테스트의 대상보다 추상화, 플레이어가 이동하는 공간, 탐색의 대상으로의 연구를 수행해왔다. 본 장에서는 게임 맵과 게임 사용자 연구를 다음과 같이 맵 테스트, 추상화, 플레이어 이동 경로, 탐색, 네 가지로 분류하여 소개한다.

1. 맵 픽셀 기반 게임 테스트

픽셀 기반 자동 게임 테스트 연구[5]는 Microsoft Research Asia에서 진행되었으며, 게임의 스크린샷을 활용해 게임 공간을 탐색하는 자동 테스트 에이전트인 Inspector를 제안한다. Inspector는 스크린샷의 픽셀을 스캔하여 이전 이미지와 유사한 경우 감점을 주고, 다른 경우 점수를 주는 방식으로 학습한다. 이 에이전트는 인간보다 높은 탐색률을 기록하는 성과를 보였다. 픽셀 기반 자동 게임 테스트 연구는 스크린 캡처 방식을 통해 소스 코드에 접근하지 않고도 다양한 게임에 적용할 수 있는 장점이 있다. 본 연구는 맵 픽셀 기반 게임 테스트와 동일하게 소스 코드가 필요 없다. 픽셀 기반 자동 게임 테스트 연구는 고정된 이미지 크기와 그리드 크기를 사용하기 때문에 다양한 환경과 예외를 처리가 어렵지만, 본 연구는 그리드 크기 변경이 쉽다는 장점이 있다.

2. 맵 추상화

Du와 Wang의 연구[6]는 게임 기반 학습을 통한 AI 문해력 함양을 주요 목적으로 한다. 이 연구에서는 사용자 맞춤형 맵을 생성할 수 있는 교육용 게임 Color Conquest를 제시하며, 사용자가 맵의 구조를 직접 설정하고 조작하는 과정을 통해 AI의 추상적 개념을 학습할 수 있도록 한다.

이를 통해 학습자는 맵의 구조적 요소를 추상화하여 탐색하며, 전략적 사고와 문제 해결 능력을 경험할 수 있다. 연구에서 사용되는 맵은 사용자가 설정한 장애물, 경로, 목표 지점 등으로 구성되며, 학습 과정에서 AI 개념 이해를 돕기 위한 다양한 추상화가 반영되어 있다.

Play by Design 연구는 AI 개념 학습과 게임 기반 학습에서 맵의 추상화 과정을 중점적으로 다루지만, 맵의 그리드 크기 조절이 불가능하여 세밀한 탐색 경로 분석에는 한계가 있다. 우리는 본 논문에서 플레이어의 탐험 경로나 위치를 더 세밀하게 분석하기 위해, 그리드 시스템을 조절할 수 있는 커버리지 기법을 제안한다.

3. 맵에서 플레이어 이동 경로 분석

모바일 게임 전략 및 전술 연구[7]은 모바일 멀티플레이어 게임 Treasure의 개발과 사용자 경험을 다룬다. 이 연구의 주요 목표는 사용자가 반복적인 플레이를 통해 전술과 전략을 어떻게 발전시키는지 분석하는 것이다. Treasure는 Wi-Fi 네트워크의 변화를 게임의 핵심 요소로 설계하여, 플레이어가 신호 강도를 게임 전략에 활용한다. Treasure는 Wi-Fi 인프라의 변화를 통해 사용자 참여도를 높였다. 모바일 게임 전략 및 전술 연구와 본 연구는 모두 플레이어 행동 분석에 도움을 준다. 모바일 게임 전략 및 전술 연구는 플레이어의 협력과 경쟁 전략을 관찰하는 데 중점을 두고 있다. 반면에 본 연구는 맵의 오류를 검출하는데 중점이 있으나, 부가적으로 플레이어 행동 분석에도 도움을 줄 수 있다.

4. 게임 맵 탐색

Go Explore Complex Environments 연구[8]은 Go-Explore 알고리즘을 활용하여 3D 시뮬레이션 환경에서 도달 가능성 버그를 자동 탐지하는 접근 방식을 제안한다. 이 알고리즘은 A* 알고리즘을 기반으로 하며, 고유 체크 포인트를 저장하고 유망한 체크 포인트를 탐색한다. 이 알고리

즘을 탑재한 AI 에이전트는 내비게이션 매쉬를 사용해 맵을 탐색한다. 이 연구에서는 작은 맵, 큰 맵, 복잡 구조 맵의 세 가지 난이도의 맵에서 Go-Explore 알고리즘을 실험적으로 평가한다. 실험 결과는 Go-Explore 알고리즘이 AAA 비디오 게임의 복잡한 3D 맵에서도 효율적으로 도달 가능성 테스트를 수행할 수 있음을 보였다.

Go Explore Complex Environments 연구와 본 연구 모두 AI 에이전트를 이용한 게임 테스트에 초점을 두지만, Go-Explore는 체크 포인트 도달 가능성 버그 발견에 중점을 둔다. 본 연구에서는 맵의 체크 포인트 뿐만 아니라 체크 포인트 사이의 모든 오브젝트 주변부가 탐색되었는지 커버리지로 확인 할 수 있다. 따라서 본 연구에서 제안하는 커버리지는 체크포인트들 사이의 오브젝트들의 맵이탈 오류들을 검출하고, Go-Explore 알고리즘은 체크포인트 도달 가능성 버그를 검출함으로써 상호 보완적으로 사용될 수 있다.

III. 구조적 게임 맵 커버리지

구조적 커버리지는 소프트웨어 테스트를 위한 기준 방법론이다. 이 방법론은 게임 맵이 아닌 C, Java와 같은 일반 소프트웨어에서 주로 사용되어왔다. 소프트웨어의 구조적 테스트 커버리지는 프로그램을 구조화했을 때 각 구조가 실행이 되었는지를 확인한다. 구조의 실행 여부가 테스트의 기준이 된다. C와 Java는 제어 흐름도와 같은 구조도를 그리고, 구조도의 지점을 실행시킬 수 있는 테스트 커버리지들이 제안되어왔다.

구문 커버리지는 구조적 커버리지의 대표적인 예이다. 그림1은 예제 프로그램의 구문 테스트 커버리지를 측정하기 위해 제어 흐름도를 그린 예이다. 그림에서 1~8줄의 구문을 그래프의 노드로 변환시키고, 각 구문이 모두 실행되면 구문 테스트 기준 커버리지를 100% 달성했다고 한다. 많은 연구들이 커버리지를 100% 달성하면 프로그램에 결함이 없을 것이라고 가정한다. 원자력 발전소 시스템과 같은 안전 필수 소프트웨어는

```

1: int i;
2: int sum = 0;
3: int product = 1;
4: int w = 7;
5: for (i = 1; i < N; ++i) {
6:     sum = sum + i + w;
7:     product = product * i;
8: }
    
```

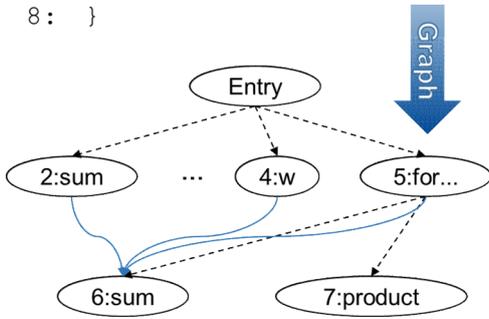


그림 1 예제 프로그램 구조화

구문 커버리지 100% 달성을 필수 요구조건으로 설정하도록 한다.

위와 같이 소프트웨어 결함 테스트도 필요하지만, 게임 맵 상에서도 결함 검출을 위해 테스트가 필요하다. 특히 게임 맵의 크기가 커지고, 맵 오류가 증가하면서 게임 맵을 테스트하기 위한 구조적 테스트의 기준이 요구되고 있다. 이를 위해 우리는 게임에 필수로 모델링되는 게임 맵의 테스트의 기준인 구조적 그리드 맵 커버리지 (Structural grid coverage)를 제안한다.

구조적 그리드 커버리지를 기준으로 테스트를 수행하는 데는 게임 맵을 그리드 단위로 쪼개는 것과 이동하는 에이전트가 필요하다.

구조적 그리드 커버리지에서 맵은 그림 2에서 보이는 바와 같이 그리드 단위로 나뉜다. 게임 테스터는 맵을 에이전트로 이동시키며 테스트를 수행한다. 게임 맵 상에서 발생할 수 있는 크래시나 맵 이탈 에러 없이 에이전트가 이동하면, 해당 그리드는 커버되었다고 기록된다. 구조적 그리드 커버리지 기준으로 테스트한 결과는 (커버된 그리드 개수) ÷ (전체 그리드 개수) × 100을 한 퍼센트 값(%)으로 출력한다. 퍼센트 값이 100%이라는 것은 게임 맵의 모든 구역을 테스트

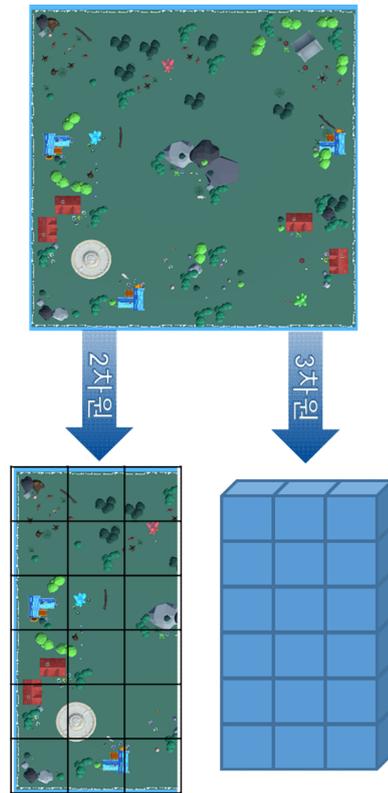


그림 2 예제 게임 맵 그리드 생성

했다는 것이다.

위 수식대로 구조적 그리드 커버리지 결과를 계산하기 위해서는 그리드를 먼저 정의하는 것이 필요하다. 그리드를 정의할 때 다음과 같이 두 가지 고려해야 할 점이 있다.

첫째는 게임 맵의 특성이다. 게임 맵은 이차원과 삼차원 등으로 나눌 수 있으며, 그림 2에서 보듯이 이차원의 그리드는 사각형으로, 삼차원은 육면체로 구성될 수 있다. 3차원 맵의 게임 중 에이전트의 이동 및 맵의 특성으로 인해 높이뛰기가 허용되지 않을 수 있다. 이 경우에는 예외적으로 커버리지 그리드를 2차원으로 평면화하여 테스트를 수행하는 것이 커버리지 기록 및 계산의 효율을 높일 수 있다.

두 번째는 그리드의 크기이다. 그리드의 크기는 테스트 비용을 고려하여 적합한 크기를 정하는 것이 최우선으로 고려되어야 한다. 기본적으로 그리드의 크기가 너무 큰 경우, 에이전트가 짧은 시간 돌아다녀도 쉽게 커버리지를 높일 수 있어서 맵에 오류가 많음에도 높은 커버리지로

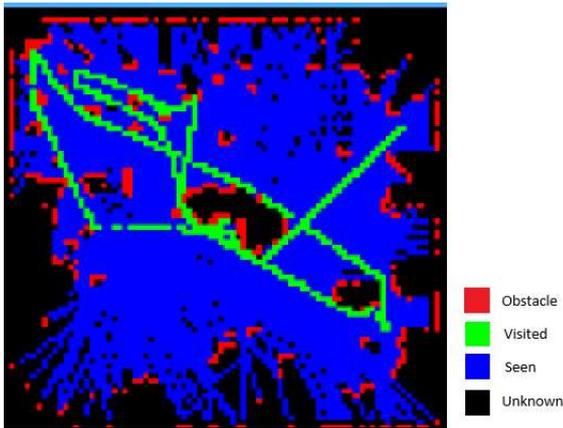


그림 3 예제 맵에서 에이전트 이동 경로 예시

테스트를 통과할 수 있다는 가설을 제시할 수 있다. 그리드의 크기가 작은 경우는 에이전트가 모든 맵을 커버하기 위해 과도한 시간과 비용이 들 수 있다는 가설을 제시할 수 있다.

그리드의 크기에 따른 위 가설들을 증명하기 위해 다음 장에서는 실험을 통해 그리드 크기에 따른 테스트 결과를 분석한다.

IV. 구조적 게임 맵 커버리지 실험

본 장에서는 3장에서 제안한 구조적 그리드 커버리지의 적용 방법과 커버리지 그리드 크기에 따른 게임 맵 분석 결과를 실험으로 보인다. 본 실험은 인공지능 에이전트를 활용하여 게임을 테스트하는 PathOS[9]를 기반으로 구현되었다. 이어지는 절에서는 PathOS의 실험 환경 설정을 설명하고, 커버리지 그리드 변경에 따른 맵 커버리지 분석 결과를 서술한다.

1. 실험 환경 설정

베타테스터와 인적자원을 활용한 반복적 테스트는 비용이 많이 들기 때문에 인공지능 에이전트를 활용하는 PathOS가 제안되었다. PathOS의 에이전트는 성격 설정이나 목적지, 엔티티 태그 등이 설정되어 실제 사람이 게임에서 이동하는 것처럼 이동한다.

우리는 그리드 크기에 따라 맵 커버리지 분석 결과가 다름을 확인하기 위해 PathOS를 기반으

로 실험 환경을 구현하였다. PathOS는 테스트에 걸린 시간, 에이전트의 경로가 전체 맵에서 차지하는 비율, 테스트 성공 여부를 제공하는 기능이 없어서 우리는 분석을 수행하기 위해 별도의 스크립트를 구현하여 테스트를 진행하였다.

본 실험 테스트는 에이전트가 이동하는 시뮬레이션 단위로 수행된다. 시뮬레이션에서는 다양한 성격을 가진 사람의 길찾기와 유사하도록 에이전트에 다양한 성격을 부여하고, 맵을 탐색하도록 설정하였다. 그림 2는 에이전트가 맵을 탐색할 때 사용된 예제 게임 맵이다. 시뮬레이션을 통해 추출된 결과는 에이전트의 성격 값, 테스트 종료 시의 맵 이미지, 목표 도달 여부, 테스트 경과 시간이 포함되었다. 이 결과들을 바탕으로, 이미지를 통해 맵의 방문 경로와 미방문 지역을 분석하고, 작은 그리드 버전(GridSmall, 83x83)과 큰 버전 (GridLarge, 43x43) 대상으로 구조적 그리드 커버리지를 계산한다.

GridSmall/Large를 정한 근거는 PathOS 제공하는 기본 그리드 크기 배율이다. GridLarge는 pathOS가 제시하는 기본 그리드 크기이다. GridSmall은 그리드 크기를 에이전트의 가로, 세로 크기이며, 이것은 기본 그리드의 0.5배율이다. 테스트는 GridSmall/Large 두 가지 환경에서 각각 50번씩 시뮬레이션을 돌려 진행되었다.

그림 3은 그림2를 그리드로 구조화하고, 시뮬레이션을 돌린 결과 나타난 에이전트의 이동 경로를 시각화한 그림이다. 테스트 대상 맵은 PathOS연구에서 사용한 게임 맵으로 대규모 오픈월드 맵의 스토리 단위 구역 크기이다. 대규모 오픈 월드 크기로 확장 시, 대규모 맵을 스토리 단위 구역 크기로 쪼개어 병렬 테스트를 수행할 수 있다. 그림 2는 구역화한 이 시각화는 맵에서 장애물(obstacle, 빨강), 에이전트의 시야(see, 파랑), 탐색한 영역(visited, 초록)과 미 탐색 영역(unknown, 검정)을 나타낸다. 그리드 커버리지는 에이전트가 이동했다고 표시되는 초록색 그리드와 전체 지도의 비율로 계산할 수 있다.

	GridLarge	GridSmall
Obstacle	17.11% (28.61)	9.39% (8.87)
Visited	22.53% (78.54)	13.55% (34.12)
Seen	36.56% (117.80)	54.02% (196.50)
Unknown	23.80% (380.59)	23.04% (380.59)
실행시간	460.08s (44804.18)	476.86s (17686.34)

표 1 GridSmall과 GridLarge의 평균 (소괄호 안의 값은 분산)

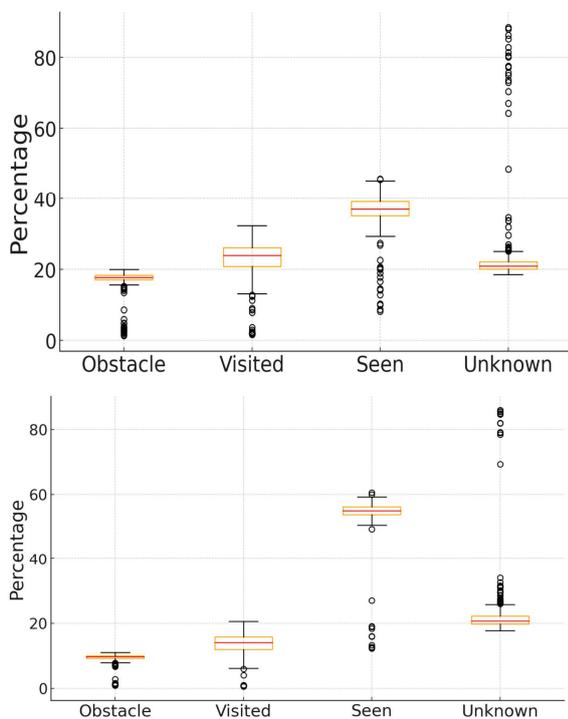


그림 4. GridSmall(위)과 GridLarge(아래) boxplot

2. 실험 결과

표 1은 실험 결과를 요약하여 나타낸 것이다. 세부적인 실험 결과를 보면, 그림 4는 GridSmall과 GridLarge를 대상으로 실험한 결과를 boxplot으로 시각화한 것이다. boxplot의 값들은 에이전트의 카메라가 객체를 확인하고 에이전트 메모리에 저장한 것이다. 그림에서 볼 수 있듯이, 그리드 크기가 달라짐에 따라 맵 커버리지 분석이 차이가 남을 확인할 수 있다.

Obstacle은 에이전트가 전체 맵에서 탐지한 장애물의 비율을 나타낸다. GridLarge와 GridSmall을 비교할 때, 에이전트가 인식한 장애물 수에 차이가 있음을 확인할 수 있다. GridSmall의 평균은 17.11%로 GridLarge의 9.39%보다 높으며, 이는 에이전트가 GridSmall에서 더 많은 장애물을 탐지했음을 의미한다. 또한, GridSmall의 분산은 28.61로 GridLarge의 8.87보다 크며, 이는 에이전트가 탐지한 장애물의 수가 더 다양하게 분포하고 있음을 보여준다.

GridLarge에서는 장애물 탐지가 상대적으로 좁은 분포를 보인다.

Visited는 에이전트가 맵에서 방문한 경로의 비율을 의미한다. 이 항목에서 에이전트가 방문한 경로의 비율은 GridSmall에서 평균 22.53%로 GridLarge의 13.55%보다 높게 나타난다. 이는 에이전트가 GridSmall에서 더 많은 경로를 탐색했음을 의미한다. 또한, GridSmall의 분산은 78.54로 GridLarge의 34.12보다 크며, 이는 탐색된 경로의 비율이 매우 다양하게 분포하고 있음을 나타낸다. 따라서 GridSmall 설정에서는 탐색 패턴의 변동성이 더 크고, GridLarge 설정에서는 평균 주변에 고르게 분포함을 확인할 수 있다.

Seen은 에이전트가 시야 내에서 탐지한 전체 맵의 비율을 나타낸다. GridLarge의 평균이 54.02%로, GridSmall의 36.56%보다 높다. 이는 에이전트가 GridLarge 설정에서 더 넓은 범위를 탐지했음을 보여준다. GridLarge의 분산은 196.50으로 GridSmall의 117.80보다 높게 나타나, 탐지된 영역이 더 넓고 다양하게 분포하고 있음을 의미한다. 이는 GridSmall에서 에이전트의 시야가 더 제한적임을 나타낸다.

Unknown은 에이전트가 미탐색하거나 확인하지 못한 맵의 비율을 의미한다. 이 항목에서는 GridLarge과 GridSmall의 평균이 각각 23.04%와 23.80%로 유사하게 나타나, 에이전트가 탐색하지 못한 영역의 비율은 두 설정 간에 큰 차이가 없음을 보여준다. 그러나 GridSmall의 분산은

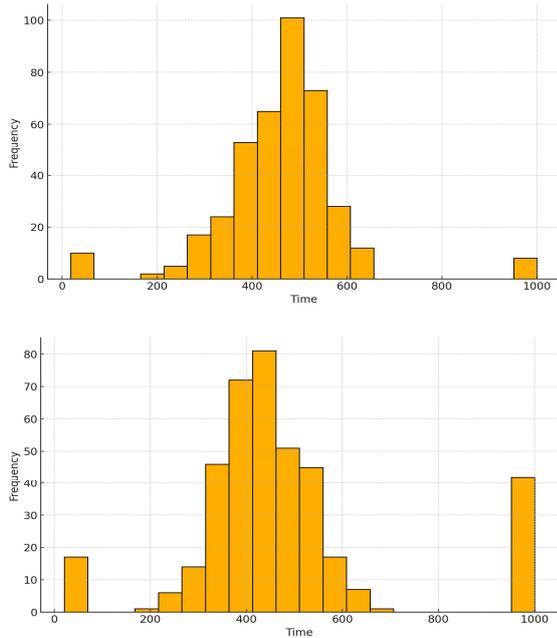


그림 5. GridSmall(위)과 GridLarge(아래) 시간 분포도

380.59로 GridLarge과 동일한 수치를 보이나, 분포가 약간 더 넓어져 에이전트가 파악하지 못한 영역의 변동성이 다소 크다는 것을 나타낸다.

그림 5는 GridSmall과 GridLarge의 시간분포도이다. GridSmall의 경우 평균 소요 시간은 476.86초가 소요되었고 최소시간은 20.70초 표준 편차는 211.67초이다. GridLarge의 경우 평균 소요 시간은 460.08초가 소요되었고 최소시간은 18.11초, 표준 편차는 132.99초이다. 둘을 비교했을 때 평균 시간은 GridSmall이 17.17초 더 많이 소요되었다. 표준 편차의 경우 GridSmall이 시간 소요에서 더 큰 변동성을 보였으며 GridLarge가 더 적은 시간을 소요할 수 있음을 보여준다.

3. 분석 및 활용

boxplot과 분포도를 분석한 결과 GridSmall과 GridLarge 일 때 각각 장단점이 있었다. 그림 6은 GridSmall과 GridLarge의 에이전트가 맵을 탐색한 경로를 시각화시킨 그림이다. GridLarge의 경우 탐색 시간이 짧아지고, 데이터 세트가 작아 분석 및 시각화가 쉽다. 반면, GridSmall의 경우 셀의 증가로 데이터 포인트가 많아지며, 이

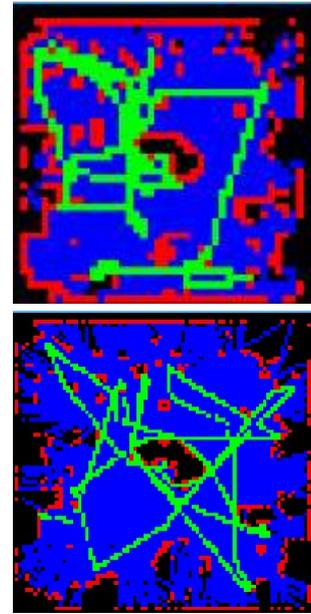


그림 6. GridSmall(위)과 GridLarge(아래) 에이전트 탐색 경로

는 미세한 경로 감지와 통계적 유의성 증가로 이어져 더 상세한 분석이 가능하다. 그러나 데이터 세트가 1/4로 줄어든 것에 비해 실행시간은 1/20로 근소하게 줄어들었다. 게임 테스트가 출시 스케줄을 크게 변동시키지 않는다면, GridSmall로 커버리지를 측정하는 것이 유리해 보인다.

pathOS 기본 그리드 크기 커버리지로 테스트가 수행된다면, 평균 커버리지 차이는 최소 9.89%에서 최대 36.56%만큼 커버리지 오탐율이 발생한다. 즉, 커버리지 정확도가 36% 증가하는 것이다. 커버리지 정확도가 오류 검출율에 미치는 영향을 분석하기 위해 맵상의 오브젝트 주변에 20개의 에러를 검출하는 시뮬레이션을 수행하였다. GridSmall와 GridLarge는 각각 평균적으로 87.56%와 92.78% 오류 검출률을 보인다. Grid Large가 더 높으나 커버리지 오탐율이 발생해 59%의 오류를 검출한다. 결과적으로 GridSmall이 더 높은 오류 검출률을 보인다.

위 분석 결과는 일반적인 게임 베타테스트에 활용될 수 있을 뿐 아니라, 앞으로 AI 에이전트 커버리지 측정 시 사용할 수 있는 알고리즘 적용 연구 등에 활용될 수 있다. 위 실험 결과에서는 보여주는 것과 같이 pathOS 알고리즘과

GridSmall의 조합이 오류 검출율을 높였으나, 다른 탐색 알고리즘에서는 다른 그리드 크기에 따라 최적의 오류 검출율을 얻을 수 있다. 추후 연구로 여러 탐색 알고리즘과 그리드 크기 조합에 대한 오류 검출율 변화를 진행 중이다.

V. 결 과

본 논문에서는 게임 맵을 효과적으로 테스트하기 위해서 구조적 맵 커버리지인 구조적 그리드 커버리지를 제안하고, 실험을 통해 그리드 커버리지의 효과를 분석하였다. 분석 결과, 그리드 크기가 작을수록 탐색 경로의 정확도는 증가했지만 소요 시간도 증가하였다. 반면, 그리드 크기가 크면 탐색 시간은 줄어들고, 경로의 세부적인 분석은 어려웠다. 이 결과는 게임 맵 테스트 뿐 아니라 게임 난이도 조절 시 최적의 커버리지 설정을 선택하는 데 중요한 자료로 활용될 수 있다.

AI 에이전트의 이동 방식에 따라 커버리지 결과 및 오류 검출율이 달라질 수 있다. 게임 맵을 더 많이 탐색할 수 있는 여러 가지 길 찾기 알고리즘과 비교하여, AI 에이전트가 더 높은 커버리지를 달성할 수 있는지에 대한 심도 있는 분석을 진행할 계획이다. 이를 통해 맵 테스트 과정에서 여러 선택지를 제공하고, 다양한 전략을 적용할 수 있는 기반을 마련할 수 있을 것이다. 이러한 연구는 맵 디자인의 최적화를 돕고, 게임 플레이 경험을 더욱 향상하는 데 이바지할 것이다.

REFERENCES

- [1] Lavoué, É., Monterrat, B., Desmarais, M., & George, S. (2020). "Adaptive gamification in education: A literature review of current trends and developments," *User Modeling and User-Adapted Interaction*, vol. 30, no. 3, pp. 401 - 425.
- [2] Lee, Seyeon, et al. "Mobile game design guide to improve gaming experience for the middle-aged and older adult population: user-centered design approach," *JMIR Serious Games*, vol. 9, no. 2, 2021.

- [3] Gerling, K., Mandryk, R. L., & Linehan, C. (2020). "Player Experience and Game Design in Health Games: A Systematic Review," *JMIR Serious Games*, vol. 8, no. 3, 2020.
- [4] Touns Dugas, P. O., LaLone, N., Spiel, K., & Hamilton, B. (2020). "Paper to Pixels: A Chronicle of Map Interfaces in Games," *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, pp. 1433-1444. 2020.
- [5] M. Schmidt, N. Nassif, D. A. M. Zaidi, P. A. F. P. Costa, and D. B. Youssef, "Inspector: Pixel-Based Automated Game Testing via Exploration, Detection, and Investigation," 2020.
- [6] Du, X.X., & Wang, X. (2023). "Play by Design: Developing Artificial Intelligence Literacy through Game-based Learning," *Journal of Computer Science Research*, vol. 5, no. 4, pp. 1-12, 2023.
- [7] M. Lopez, B. Olsson, and S. Carlson, "11551201_21," 2021.
- [8] A. Ecoffet, J. Huizinga, A. Lerer, R. Wang, and J. Clune, "Go-Explore: Complex 3-D Game Environments for Automated Reachability Testing," 2020.
- [9] Stahlke, S., Nova, A., & Mirza-Babaei, P. (2020). "Artificial players in the design process: Developing an automated testing tool for game level and world design," In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '20)*, Virtual Event, Canada. ACM, Nov. 2-4, 2020,

저 자 소 개



이정준(준회원)

2019년 한국게임과학고등학교 졸업
2024년 한남대학교 컴퓨터공학과 학사 재학
<주관심분야 : 게임프로그래밍, 게임 엔진>



송지영(정회원)

2014년 이화여자대학교 컴퓨터공학과 학사 졸업.
2000년 한국과학기술원 전산학부 석사 졸업.
2004년 한국과학기술원 전산학부 박사 졸업.

<주관심분야 : 소프트웨어공학, 모델 기반 SW 테스트 및 검증, 시스템 오브 시스템즈>