



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

碩士學位論文

강화학습을 이용한 게임 콘텐츠의
시스템 테스트 자동화

濟州大學校 大學院

컴퓨터工學科

李石基

2018年 8月

강화학습을 이용한 게임 콘텐츠의 시스템 테스트 자동화

指導教授 郭 鎬 榮

李 石 基

이 論文을 컴퓨터工學 碩士學位 論文으로 提出함

2018 年 06 月

李石基의 工學 碩士學位 論文을 認准함

審査委員長 이 상권
委 員 김 도현
委 員 김 호영

濟州大學校 大學院

2018 年 06 月

System testing automation of game content
using reinforcement learning

Sukki Lee

(Supervised by professor Ho-Young Kwak)

A thesis submitted in partial fulfillment of the requirement for
the degree of Master of Science in Computer Engineering

2018. 06.

This thesis has been examined and approved.

Thesis director, Gyung-Joon Lee
Thesis director, Do Hyeon Kim
Thesis director, Ho-Young Kwak

June 2018

Department of Computer Engineering

GRADUATE SCHOOL

JEJU NATIONAL UNIVERSITY



목 차

I. 서론	1
1.1 연구배경 및 필요성	1
1.2 연구 내용	2
II. 관련 연구	4
2.1 게임 개발	4
2.2 게임 테스트	6
2.2.1 테스트 유형	6
2.2.2 테스트 단계	8
2.3 게임 테스트 자동화	10
2.4 기계학습과 게임	11
III. 강화학습을 이용한 게임 테스트 자동화	13
3.1 자동화 시스템 개요	13
3.2 인공 신경망의 게임 조작	16
3.3 인공 신경망의 게임 상태 감지	17
3.4 테스트 보상함수와 행동 선택 정책	20
3.5 테스트 에이전트 학습	22
3.6 테스트 수행 및 결함 발견	23
3.7 기능 변경에 기존 테스트 자동화 방법과 비교	25
IV. 실험 및 분석	28
4.1 사례1 : 게임 상점 기능 테스트	28
4.1.1 실험 대상	28
4.1.2 강화학습을 이용한 테스트 자동화 실험	30
4.1.3 절차형 테스트 자동화 실험	32
4.1.4 테스트 자료 변동량 비교	33
4.2 사례2 : 게임 던전 입장 테스트	34
4.2.1 실험 대상	34
4.2.2 강화학습을 이용한 테스트 자동화 실험	37
4.2.3 절차형 테스트 자동화 실험	38
4.2.4 테스트 자료 변동량 비교	39

4.3 사례3 : 게임 스킬 장착 테스트	41
4.3.1 실험 대상	41
4.3.2 강화학습을 이용한 테스트 자동화 실험	42
4.3.3 절차형 테스트 자동화 실험	44
4.3.4 테스트 자료 변동량 비교	45
4.4 종합 분석	47
V. 결론	49
VI. 참고문헌	50

그림 목 차

그림 1. 나선형 개발 모델	5
그림 2. 테스트 V모델	8
그림 3. 강화학습 시스템 구성도	12
그림 4. 강화학습 테스트 자동화 시스템 구성도	13
그림 5. 게임 테스트 자동화 시스템 전체 흐름	15
그림 6. 아재 온라인 게임 수정 상점 화면	16
그림 7. 아재 온라인 게임 수정 상점 화면, 마우스 좌표 행동 집합	17
그림 8. 아재 온라인 게임 수정 상점 화면, 화면 직접 입력	18
그림 9. 아재 온라인 게임 수정 상점 화면, 상태 선별 통신	19
그림 10. 테스트용 기계 학습 특수성	22
그림 11. 시스템 테스트 활동 흐름	23
그림 12. 기계학습 테스트 에이전트 테스트 활동 흐름	25
그림 13. 게임 상점 원본	28
그림 14. 게임 상점 구매 절차 변경	28
그림 15. 게임 상점 구매 절차 변경 없음	28
그림 16. 게임 상점 공통화면	29
그림 17. 정상동작 기록	31
그림 18. 던전 입장 원본	35
그림 19. 던전 입장 절차 변경1	35
그림 20. 던전 입장 절차 변경2	35
그림 21. 던전 입장 절차 변경 없음	35
그림 22. 던전 입장 공통화면	36
그림 23. 스킬 장착 원본	41
그림 24. 스킬 장착 절차 변경	41
그림 25. 스킬 장착 절차 변경 없음	41
그림 26. 스킬 장착 공통화면	41

표 목 차

표 1. 게임 구조 특수성	6
표 2. 테스트 유형 정리	8
표 3. 테스트 단계 정리	9
표 4. 강화학습 인경신경망을 위한 최소 테스트 정보	19
표 5. 기능 동작 기준 결함 명세	20
표 6. 아재 온라인 명예 상점 구매기능 테스트 케이스	21
표 7. 아재 온라인 명예 상점 구매기능 테스트 보상함수 설계	21
표 8. 소프트웨어 제품 결함, 심각도 기준 분류	23
표 9. 절차형 테스트 자동화, 상점 구조 변경 시 테스트 자료 수정	26
표 10. 강화학습 테스트 자동화, 상점 구조 변경 시 테스트 자료 수정	26
표 11. 테스트 자동화 기법별 게임 적용 비용	27
표 12. 게임 상점 테스트 제어변수별 게임 동작	29
표 13. 게임 상점 강화학습 테스트 자동화 적용	30
표 14. 게임 상점 강화학습 테스트 자동화 보상함수	31
표 15. 게임 상점 강화학습 테스트 자동화 적용 결과	32
표 16. 게임 상점 절차형 테스트 자동화 적용	32
표 17. 게임 상점 강화학습 테스트 자동화 변동내용	33
표 18. 게임 상점 절차형 테스트 자동화 변동내용	33
표 19. 게임 상점 테스트 자동화 기법별 변동내용 비교	34
표 20. 던전 입장 테스트 제어변수별 게임 동작	36
표 21. 던전 입장 강화학습 테스트 자동화 적용	37
표 22. 던전 입장 강화학습 테스트 자동화 보상함수	38
표 23. 던전 입장 강화학습 테스트 자동화 적용 결과	38
표 24. 던전 입장 절차형 테스트 자동화 적용	39
표 25. 던전 입장 강화학습 테스트 자동화 변동내용	39
표 26. 던전 입장 절차형 테스트 자동화 변동내용	40
표 27. 던전 입장 테스트 자동화 기법별 변동내용 비교	40
표 28. 던전 입장 테스트 제어변수별 게임 동작	42
표 29. 스킬 장착 강화학습 테스트 자동화 적용	43
표 30. 스킬 장착 강화학습 테스트 자동화 보상함수	44
표 31. 스킬 장착 강화학습 테스트 자동화 적용 결과	44
표 32. 스킬 장착 절차형 테스트 자동화 적용	45
표 33. 스킬 장착 강화학습 테스트 자동화 변동내용	46

표 34. 스킬 장착 절차형 테스트 자동화 변동내용	46
표 35. 스킬 장착 테스트 자동화 기법별 변동내용 비교	47
표 36. 전체 사례 테스트 자료 변동량 비교	47

강화학습을 이용한 게임 콘텐츠의 시스템 테스트 자동화

컴퓨터공학과 이 석 기
지도교수 곽 호 영

본 논문은 게임 콘텐츠의 시스템 테스트 자동화에 대한 연구이다. 기존의 테스트 자동화 기법들은 테스트 절차를 테스트 전용 자료에 기록하고 재생하는 방식으로, 기능 변경 시 테스트 자동화 자료 유지보수 비용이 발생한다. 이는 변경이 잦은 게임 개발에는 부담이 된다. 이 문제를 해결하기 위해 테스트 목표를 보상함수로 명세하고 테스트 절차를 강화학습을 통해 수행시켰다. 제안한 테스트 자동화 기법과 기존 테스트 자동화 기법의 테스트 자동화 자료 유지보수 비용을 측정하기 위해 게임 콘텐츠 중 상점, 던전 입장, 스킬 장착을 대상으로 실험하였으며, 측정결과 절차형 테스트 자동화 기법 대비 약 7.2%의 테스트 자동화 자료 변경으로 테스트 자동화가 가능하였다.

주제어 : 게임 테스트, 테스트 자동화, 강화학습

abstract

A System Testing Automation of Game Contents using Reinforcement Learning

Lee, Sukki

Department of Computer Engineering

Graduate School

Jeju National University

Supervised by Professor Kwak, Ho-Young

This paper is a study on system test automation of game contents. Well-known test automation techniques are a recording and replaying test procedures in test data. This technique incurs the cost of maintenance of test automation data when the game contents is changed, which is a burden on game development that changes frequently. To solve this problem, the test objectives were specified as reward functions and the test procedure was performed through reinforcement learning. In order to measure the maintenance cost of the proposed test automation technique and the existing test automation technique, we experimented with the store contents, dungeon entry, and skill selection. The proposed automation was possible by changing the test automation data about 7.2% compared with procedural test automation technique.

Key words: Game Test, Test automation, Reinforcement learning

I. 서론

1.1 연구배경 및 필요성

한국콘텐츠진흥원의 자료에 따르면 2016년 한해 전 세계 게임시장 매출이 910억 달러(약109조원)에 이르는 것으로 집계됐다.[1] 국내에도 넥슨, 엔씨소프트, 넷마블 등 개발진 500명 이상의 대형 게임 제작사가 다수 존재하며, 해외 게임 제작사를 포함하면 수십 개의 대형 게임 제작사가 경쟁하고 있다. 최근 제작되는 게임은 높은 수준의 다국어 지원과 여러 국가의 동시 서비스를 목표로 제작되고 있고, 중국의 게임 산업에 높은 투자로 경쟁은 더욱 심화될 것으로 보인다. 치열한 경쟁에서 게임 개발사들에게 신속한 개발은 생존과 직결되며, 업무 효율 향상을 위해 멀티 플랫폼 상용 게임 엔진 이용, 애자일 개발 방법론 도입과 테스트 자동화 등 다양한 시도를 하고 있다.

소프트웨어 테스트 기술은 응용 프로그램 또는 시스템의 동작과 성능, 안정성이 요구하는 수준을 충족시키는지 확인하기 위해 개발 전 과정에 걸쳐 결함을 발견하는 행위로 현재 그 중요성이 부각되어 IT산업은 전반에 걸쳐 적용되고 있다.[2] 최근 게임 업계는 고객에게 보다 높은 품질의 제품을 제공하는 것을 목표로 QA 업무의 중요성을 높이 평가하고 있다.[3] 소프트웨어 테스트는 개발에 꼭 필요한 과정으로 품질에 큰 영향을 미친다.

소프트웨어 개발에 소요되는 총 비용의 50% 이상과 총 기간의 50%정도가 개발된 소프트웨어의 테스트 작업에 할당된다.[4] 게임 개발 비용에 테스트는 큰 비중을 차지하며, 테스트 실패 비용은 엄청나다. 최근 출시된 넥슨의 듀랑고와 펠어비스의 검은사막은 장기간 큰 비용을 투자한 프로젝트이나, 출시 초기에 서버 불안정으로 매출 손실과 사용자의 게임에 대한 전반적인 인식 저하 등 큰 피해를 입었다.

어플리케이션 자동화 테스트란 테스트 과정의 반복적이고 시간 소모적인 작업을 효율적으로 자동 처리하여 자원을 절약하고 인적 오류를 줄일 수 있는 효과적인 방법을 말한다.[5] 게임은 예술 창작물인 특수성 때문에 테스트 자동화 적용이 쉽지 않다. 일반적인 소프트웨어는 명확한 기능성이 있고, 합리성을 추구한다. 게임은 재미와 감동 전달을 추구하기에 요구사항부터 추상적이다. 이에 다양한 시도와 많

은 재개발이 발생한다. 일반적인 소프트웨어는 개발 완료된 기능의 전면 재개발이 드문 일이지만, 신규 게임 프로젝트에서는 종종 일어나는 일이다. 테스트 자동화가 적용된 상태에서 기능 재개발을 할 경우 테스트 자동화를 위한 데이터와 시스템을 함께 수정해야한다. 이로 인해 기능이 언제 변경될지 예측할 수 없으므로 개발 시작과 동시에 자동화 대상에 포함시키는 것은 개발 비용을 증가시키는 결과가 될 수도 있다.

본 연구에서는 이러한 게임 제작 환경을 고려하여, 기계학습 도입으로 자동화를 위한 데이터와 시스템의 유지보수를 최소화하는 방안을 제안한다.

1.2 연구 내용

게임 테스트 활동 중 빠질 수 없는 부분이 기능 명세를 분석하여 테스트 케이스를 생성하고, 게임을 실행시켜 기능 동작을 확인하는 것이다. 사용자가 재미를 위해 게임을 실행시키고 기능을 이용하면 게임 플레이지만, 테스트 담당자가 기능 검증을 위해 게임을 이용한 활동은 테스트다. 즉, 테스트 활동 중 게임을 실행시켜 기능을 검증하는 것은 게임 플레이와 동일하다.

강화학습에 기반을 둔 인공 신경망 모델들이 게임 플레이에서 좋은 성과를 거두고 있다. Playing FPS Games with Deep Reinforcement Learning[6]에서 강화학습 인공신경망 인공지능은 간단한 아케이드 게임이지만 높은 수준의 게임 진행을 보여 주었으며, Playing FPS Games with Deep Reinforcement Learning에서는 FPS 장르의 고전인 게임 Doom2를 플레이해 높은 점수를 기록했다.

강화학습은 보상함수를 이용하여 현재 점수를 계산하고 결과를 인공신경망에 전달한다. 점수를 통해 신경망은 현재 상태와 현재까지 한 행동들의 가치를 재조정하고 다음 시도를 한다. 이 보상함수는 강화학습의 가치기준이라 여길 수 있으며, 계산 공식을 조정하면 에이전트의 행동을 유도할 수 있다. 가령 에이전트가 한발 내딛을 때마다 1점을 가산해 주면 에이전트는 쉬지 않고 움직일 것이며, 상점에서 새로운 항목을 구입할 때마다 1점을 가산하면 에이전트는 새로운 항목을 계속 구매할 것이다. 따라서 강화학습은 게임의 본 기능을 이용하는 것 이외에 다양한 기능을 이용하도록 학습시킬 수 있다.

기존 자동화 방식은 테스트 자동화에 필요한 자료가 많아 제작 및 유지보수 비용이 큰 문제가 있다. 이 문제는 기존 테스트 자동화 방식은 테스트 자동화 자료에 테스트 과정을 모두 지정해줘야 하는 것이 원인이다. 인공 신경망은 입력과 출력을 연결해 주면 중간 과정은 학습으로 대체 가능하다. 그리고 테스트에 인공 신경망을 적용 한다면 학습 과정도 유의미하다. 테스트에는 기능의 정상 동작 외에 게임의 오동작을 감지하기 위해 스트레스 테스트를 수행해야 하는데, 인공 신경망의 학습 과정은 이 스트레스 테스트를 대체할 수 있다.

본 연구에서는 게임 테스트의 일부가 게임 플레이인 것과 강화학습 인공 신경망 모델들이 게임에서 좋은 성과를 거둔 것에 착안하여 테스트 비용을 절감할 테스트 자동화 방법을 제안하고, 게임 기능 일부를 대상으로 실험하여 그 효과를 검증했다.

II. 관련 연구

게임 테스트 자동화의 어려움은 게임의 특수성과 밀접한 관련이 있다. 게임은 개발 과정에서 많은 요구사항 변경과 다수의 중간 개발 단계가 있다. 그리고 타 소프트웨어와 달리 방대한 양의 콘텐츠가 있으며, 이를 모두 테스트하는 데에는 많은 비용이 필요하다.

이 장에서는 게임 개발과 게임 테스트의 특수성을 설명한다. 그리고 게임 테스트 자동화 기법을 설명한 뒤 한계점을 지적한다. 강화학습에 대해 설명하고 게임 테스트와 연관성을 살펴본다.

2.1 게임 개발

대부분 응용 프로그램은 사용자 편의 향상이나 명확한 기능적 서비스 제공을 목표로 한다. 이는 일상생활의 일부를 개선하는 것으로, 설문조사나 핵심 대상 사용자 표본의 행동 관찰 등 여러 방법으로 요구사항을 수집하고 분석할 수 있다. 반면, 게임은 사용자에게 재미와 감동을 주는 것이 목표로 요구사항을 분석하는데 어려움이 많다. 그리고 시장의 불규칙성으로 인해 개발 완료된 기능도 빈번한 변경이 일어난다.

게임 개발은 나선 개발 모형(Spiral Model)을 많이 사용한다. 나선형 모델을 도식화하면 그림 1.과 같다.

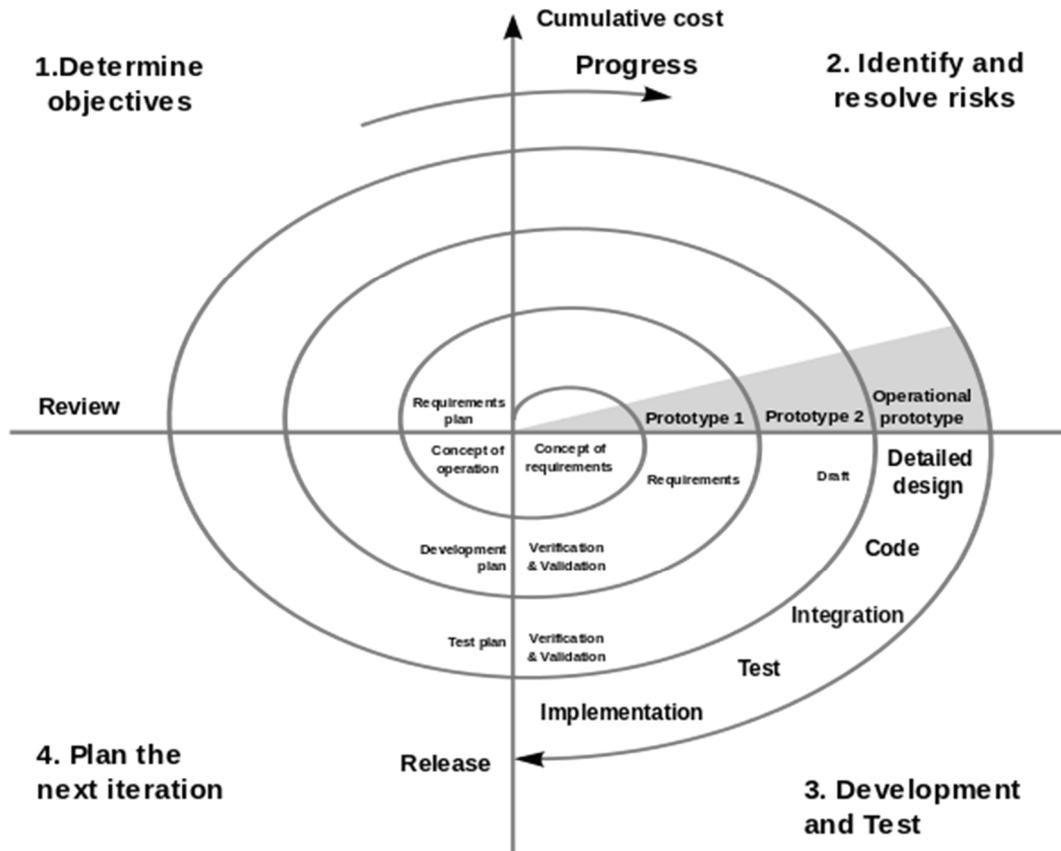


그림 1. 나선형 개발 모델[7]

게임 개발 중 출시 이전인 제품 제작 과정에서는 한 주기는 짧게는 1주, 길게는 분기로 진행하며 드물게 반기나 그 이상의 경우도 있다. 각 주기의 끝에는 예상 핵심 사용자층의 표본과 프로젝트 관련자와 개발자가 제품의 품질을 평가한 후 다음 목표를 수립한 뒤 개발한다. 그리고 출시 이후에도 PC온라인 게임이나 대형 모바일 게임은 지속적으로 개발하여 소프트웨어 수명을 연장한다. 국내 유명 PC 온라인 게임 검은사막의 최근 변경내역을 보면 매주 100건 이상 기능 변경이 있었다.[8] 게임의 개발의 특수성은 한 주기가 끝나고 체험자들의 평가를 수집하여 다음 목표를 수립하는 데에서 나타난다. 재미와 감동은 사용자의 기호에 따라 크게 달라지며, 수집된 의견도 추상적인 경우가 대부분이다. 그리고 경쟁사의 유사 게임 출시나 사용자들의 감성 변화, 게임시장의 성장 및 사용자 연령과 성별의 확대 등 시장 상황이 끊임없이 변화한다. 이에 게임은 제작 중에 핵심 대상 사용자층의 변경이나

게임 장르가 바뀌기는 등 프로젝트의 목표 자체가 변경되는 경우도 종종 있으며, 개발이 완료된 기능도 목표 변경에 따라 재개발한다.

높은 생산성을 위해 게임 개발에 분업은 불가피하다. 음향, 영상은 각 담당 전문가가 제작하며, 기능 골격과 규칙은 프로그래머가 담당하고 상세 속성 값을 콘텐츠 기획자가 작성하여 완성한다. 예를 들면, 온라인 게임 리니지의 경우 캐릭터에 장착하는 장비 시스템이 있으며, 이를 이용하는 수백종류의 장비가 존재한다. 퍼즐 게임인 캔디 크러시 사가의 경우 스테이지의 골격을 이루는 시스템이 있고, 세부 규칙을 정한 많은 스테이지가 존재한다. 이처럼 게임은 소수의 거대 시스템에 다수의 콘텐츠가 있는 양상으로 일반적인 응용 프로그램과 차이가 있다. 이를 정리하면 표 1.과 같다.

표 1. 게임 구조 특수성

	응용 프로그램	게임
기능	소수, 다수	소수, 다수
콘텐츠	소수	소수, 다수

2.2 게임 테스트

게임은 개발 결과와 시스템 구성만을 보면 타 소프트웨어와 크게 다른 점이 없다. 때문에 테스트 전체 골격은 일반적인 소프트웨어 테스트와 같다. 하지만 테스트 진행 양상은 기능이 자주 변경되며 높은 생산성을 추구하는 개발 과정과 콘텐츠 양이 방대한 것에 영향을 받아 특수성을 띤다.

2.2.1 테스트 유형

다양한 목적으로 테스트를 수행하지만, 일반적인 목적은 남아있는 결함 발견, 명세 충족 확인, 사용자 및 비즈니스의 요구 충족 확인, 결함 예방이다.[9]

중점을 두는 테스트 목적에 따라 테스트 유형을 나눌 수 있다. 소프트웨어가 수행하는 기능을 확인하는 기능 테스트, 호환성이나 신뢰성 및 안전성과 같은 기능과 무관한 품질을 확인하는 비기능 테스트, 소프트웨어나 시스템의 아키텍처에 대한 구조적 테스트, 결함의 정상 수정 혹은 의도하지 않은 결함의 발생을 감지하는 것이 확인/회귀 테스트이다.

(1) 기능 테스트

기능 테스트는 의도한 대로 기능이 정상적으로 동작하는지 확인하는 테스트이다. 가령 ATM에서 5만원 인출버튼을 눌렀을 때, 출금 금액으로 5만원이 선택되고 비밀번호 입력으로 화면이 전환되는 것이다. 가장 기초적인 테스트이지만, 테스트 초기에는 기능 테스트에서도 많은 문제가 발견된다.

(2) 비기능 테스트

비기능 테스트는 ATM사례로 대입해 보면, 인출 과정 중에 정전이 발생할 경우이다. 인출 비밀번호를 모두 입력하고 돈을 세는 중간에 정전이 발생할 경우 은행은 잔고를 유지하고 ATM기기는 전원이 복귀된 후 계산하는 현찰을 다시 저장고로 되돌리는 정책을 취할 수 있다. 이에 대한 결정은 제품 설계자가 하는 것이고, 설계대로 정상 동작하는지 테스트 하는 것이 비기능 테스트이다.

(3) 구조적 테스트

구조적 테스트는 소프트웨어의 구조와 설계를 분석하여 구조적 문제가 없는지 확인하는 것이다. 가령 데이터 캐싱 정책이나 서버 구성도, 데이터베이스 설계 검토 등이 될 수 있다.

(4) 확인/회귀 테스트

확인/회귀 테스트는 소프트웨어의 기능 개선 및 변경 혹은 버그 수정이 올바르게 이루어졌는지 확인하고, 이 수정으로 인해 기존에 정상 작동한 기능에 문제가 발생하지 않았는지 확인하는 것이다. 이를 정리하면 표 2와 같다.

표 2. 테스트 유형 정리

테스팅 유형	검 증 대 상
기능 테스트	기능이 명세대로 동작하는지 확인
비기능 테스트	기능이 소프트웨어 외부/환경의 문제에 명세대로 대처하는지 확인
구조적 테스트	기능과 연관 시스템에 구조적 결함이 없는지 확인
확인/회귀 테스트	기능의 문제가 정상적으로 수정되었는지 확인 정상 동작 하는 기능들이 여전히 정상 동작하는지 확인

2.2.2 테스트 단계

각 개발단계에 따라 테스트를 단계 구분이다. 이는 테스트 V모델이라고 하며 그림 2.이다.

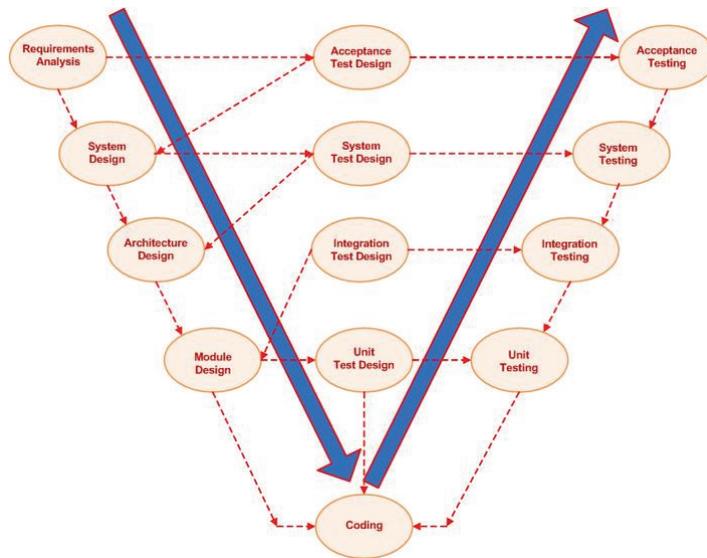


그림 2. 테스트 V모델[10]

(1) 단위 테스트

단위 테스트는 클래스, 함수 등 구현사항에 문제가 없는지 점검하는 것이다. 경계 값 확인 등을 통해 각 함수나 클래스에 오작동 가능성을 사전에 검수한다.

(2) 통합 테스트

통합 테스트는 각 기능들이 연계된 시스템 혹은 기능들과 상호작용의 정상 수행 여부를 확인한다. 통합 테스트 방법에는 하향식 방법과 상향식 방법이 있는데, 하향식 방법은 최상위 모듈에서 하위 모듈을 하나씩 통합해가며 테스트 하는 것이며, 상향식 방법은 반대로 하위 모듈에서 상위 모듈로 통합해 나가는 것이다.

(3) 시스템 테스트

시스템 테스트는 통합을 완료하고 완성된 제품을 테스트하는 것이다. 제품 동작시켜서 각 기능이 명세대로 동작하는지 결함은 없는지 확인한다. 이 단계에서 제품의 내부를 살펴보지 않고, 제품을 실행시켜서 기능 동작을 살펴본다.

(4) 인수 테스트

인수 테스트는 사용자가 제품의 품질을 검수하는 베타 테스트이다. 내용을 정리하면 표 3.이다.

표 3. 테스트 단계 정리

테스트 단계	검 증 대 상
단위 테스트	함수/클래스의 정상 동작
통합 테스트	함수/클래스 간 상호작용의 정상 동작
시스템 테스트	완성된 제품의 기능 정상 동작
인수 테스트	제품의 품질 평가

게임 개발에서 단위 테스트와 통합 테스트는 코드 리뷰와 정적 분석 도구를 통해 이루어진다. 시스템 테스트는 전문 테스트 인력이 수행하며, 인수 테스트는 프로젝트 개발자 전체와 프로젝트 연관자, 핵심 사용자 표본이 진행한다.

게임은 타 소프트웨어와 달리 콘텐츠가 방대한 것이 특징이다. 기능에 결함 발생 시 연관 콘텐츠 전체에 영향을 주기에 매우 치명적이다. 반면 콘텐츠의 경우 속성 값에 문제가 있어도 파급은 해당 콘텐츠에 한정되어 상대적으로 파급력이 적다. 계

임 테스트의 특수성은 테스트 대상이 매우 많으며, 테스트 대상의 중요도 차이가 명확한 것이다. 가령 PC온라인 게임 던전 앤 파이터의 경우 장비 콘텐츠는 만개가 넘는데 장비의 주요 기능이 변경되면 이를 전체 재검수해야 한다. 때문에 출시 전 개발 중간 단계에선 장비 기능을 점검하기 위한 표본을 선정하고 이들을 대상으로 기능 정상 동작을 점검한다. 이 표본에 선택되지 못한 콘텐츠도 장비의 성능과 같은 속성 값이 모두 확인되어야 하지만, 개발 여건이 좋지 않을 경우 전부 확인하지 못하는 경우도 있다.

2.3 게임 테스트 자동화

소프트웨어 테스트 자동화는 상당히 연구가 진척된 분야이다. 테스트 수준과 테스트 유형별로 여러 기법이 있지만, 목표인 시스템 테스트 중 기능 테스트를 대상으로 한정하여 설명하겠다. 이 테스트 자동화에는 Selenium, Appium, Monkey talk 등 상용 제품도 다수 존재한다.

테스트 자동화 기법 중 대표적인 것으로 테스트 절차를 기록하고 수행결과를 감독하는 절차형 테스트 자동화가 있다. 이 외에도 유한상태머신 인공지능을 이용한 테스트 자동화 기법이 있으며, 소프트웨어 오동작을 유발하기 위해 임의의 입력을 발생시키는 테스트 도구가 Monkey talk이다.

절차형 테스트 자동화 기법은 테스트 수행의 행동을 모두 기록하고 정상 작동을 확인하는 것이다. 국내 게임사의 적용 사례로 NC소프트의 모바일 게임 테스트 자동화 사례[11]가 있으며, Appium[12]에 기능을 추가하여 여러 해상도에서 정상적으로 수행되도록 개선한 것이다.

이 방법의 장점은 상용제품이 많아 시스템 도입이 편리하고, 절차를 기록하여 재생하는 방식이기에 적용 불가능한 대상이 거의 없는 것이다. 단점은 테스트 절차를 모두 기록해야만 정상적으로 작동하는 것이다. 테스트 대상이 많을 경우 변경사항이 발생할 경우 모든 변경에 맞추어 테스트 자료를 수정해야 한다. 게임은 기능의 수도 많고 콘텐츠 수도 많아 테스트 자동화를 위한 자료가 매우 많다. 때문에 테스트 자동화 자료 유지보수 비용이 많이 발생한다.

유한상태머신 인공지능 테스트 자동화는 뚜렷한 테스트 목적을 위해 인공지능을 별도 제작하는 것이다. 국내 연구 사례로 에이전트에 의한 온라인게임 서버 테스트 자동화[14]와 온라인 게임서버를 위한 FSM 기반 테스트 자동화 시스템[15]이 있다. 이 사례들은 서비스 안전성을 확인하기 위해 사용자와 유사한 행동을 하는 인공지능을 제작한 후 대량의 테스트 에이전트를 생성하여 시험 운전한 것이다.

이 기법의 장점은 테스트 대상에 맞춤 제작이기에 목표를 확실히 이룰 수 있는 것이다. 단점은 테스트하기 위해 기능을 분석하고 인공지능을 설계해야하며, 에이전트 제작까지 직접 해야 하는 것이다. 때문에, 적용 대상이 제한적이며 제작비용이 많고 유지보수 비용도 크다. 게임 제작 과정 중에 이 자동화를 적용할 때는 높은 비용을 지불할 만큼 테스트 목적이 중요할 경우이다. 앞선 사례의 경우가 대표적이다. 서비스의 안전성이 보장되지 않을 경우는 큰 매출 손실이 발생하며 제작사의 이미지 실추 등 많은 악영향이 있다. 따라서 게임의 전반적인 기능 테스트에 적용에는 다른 자동화 방법이 필요하다.

테스트 자동화는 초기 투입 비용의 증대를 포함하고 있기에 테스트 자동화를 도입하는 조직에서는 적합한 대상에 효율적으로 테스트 자동화를 추진하는 것이 중요하다.[13] 테스트 자동화는 자동화에 필요한 비용을 지불하여 이후 테스트 비용을 절감하는 것이다. 절차형 테스트 자동화는 유지비용이 높고 유한상태 머신 인공지능 방식은 인공지능 개발 비용이 발생하며, 기능 변경 시 유지비용이 필요하다. 따라서 게임 개발과정에 적합한 다른 테스트 자동화 방안이 필요하다.

2.4 기계학습과 게임

강화학습은 주어진 환경에서 에이전트가 행동을 취하면서, 보상을 취하게 되고 이 보상으로 이전까지의 행동과 상태에 가치를 평가하는 학습 형태이다. 이를 도식화하면 그림 3.과 같다.

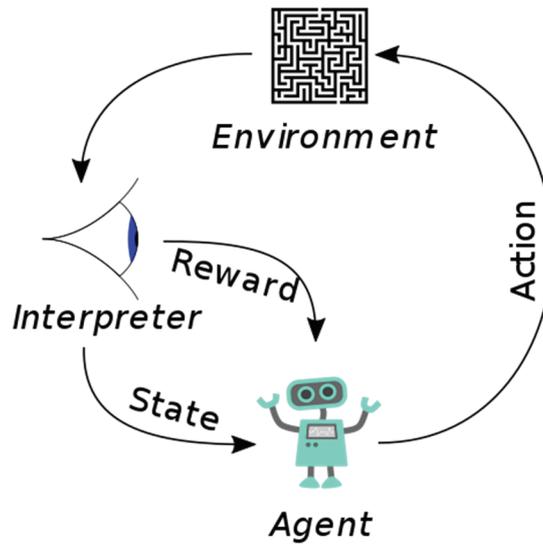


그림 3. 강화학습 시스템 구성도[16]

강화학습 인공 신경망의 학습과 행동에 큰 영향을 미치는 것이 행동 선택 정책과 보상함수이다. 행동 선택 정책의 조정으로 에이전트는 현재 상태와 다음 선택 가능한 행동들의 가치를 확인해보고 그 중의 하나를 선택한다. 이 정책으로 20% 확률로 가장 가치가 낮은 행동을 하도록 설정하여 엉뚱한 행동을 하는 에이전트를 만들 수도 있다.

보상함수는 에이전트의 가치판단 기준이 되며, 보상함수의 계산식을 조정하여 에이전트를 우리가 원하는 방식으로 행동하게 만들 수 있다. 가령 환경이 온라인 쇼핑물이고, 에이전트의 보상 함수로 장바구니에 품목의 종류가 다양할수록 가산점을 주고, 한 항목의 수량이 1이상일 경우 감산한다면 에이전트는 최대한 많은 품목을 단 1개만 추가하기 위해 노력할 것이다. 이처럼 강화학습은 게임의 에이전트 상태에 관한 몇 가지 정보만 제공하고 보상함수의 조절만으로 좋은 결과를 얻을 수 있어서 게임 인공지능으로 유용하다.

게임 테스트 활동은 게임의 각 기능을 이용하여 지정된 행위를 하는 것이다. 가령 상점 기능을 테스트 한다면, 상점에 등록된 모든 상품을 구매하는 것이다. 이는 상점의 상품을 구매할 때 1점씩 가산하도록 강화학습의 보상함수를 작성할 수 있다. 이처럼 강화학습은 게임전반 기능에 적용할 수 있어 테스트 에이전트로 이용할 수 있다.

Ⅲ. 강화학습을 이용한 게임 테스트 자동화

기존 테스트 자동화 기법이 게임 개발에 적용되기 힘든 원인은 기능 변경에 따른 테스트 자동화 유지보수 비용에 있다. 게임은 높은 경쟁으로 빠른 개발과 많은 콘텐츠와 기능 변경은 불가피하다. 때문에 유지보수 비용이 적은 자동화 시스템이 필요하며 강화학습은 좋은 대안이다.

본 장에서는 강화학습을 인공지능으로 사용한 테스트 자동화 시스템을 설명하며, 제안한 기법과 기존 자동화 기법을 비교하여 유지비용의 절감효과를 설명한다.

3.1 자동화 시스템 개요

그림 4.은 본 논문에서 고안한 테스트 시스템 구성도이다.

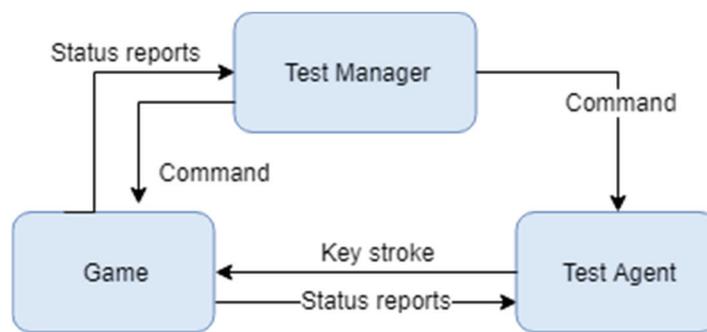


그림 4. 강화학습 테스트 자동화 시스템 구성도

테스트 관리자는 전체 흐름을 관리한다. 게임에게 테스트 준비 명령을 내리고 준비를 마치면 테스트 에이전트에게 테스트 수행을 명령하고, 게임의 변화를 주시하여 테스트 에이전트의 목표 달성 혹은 실패를 판단하며 다음 행동을 지시한다.

게임은 테스트 관리자의 명령을 받으면 테스트 케이스에 명시된 테스트 선행조건을 준비한다. 테스트 진행 중에는 게임 내에 발생한 사건, 게임 승리 혹은 패배 및 물품 구입 등을 테스트 관리자와 테스트 에이전트에게 전달한다.

테스트 에이전트는 게임이 받아들이는 각종 입력을 전달하여 테스트를 수행한다. 각종 입력이란 PC의 경우 키보드나 마우스를 의미하며, 스마트폰의 경우 화면 터

치나 드래그를 의미한다. 테스트 에이전트가 주어진 목표를 달성하지 못하는 경우는 두 가지이다. 하나는 학습이 부족하여 적절한 상황판단을 하지 못하고, 적합 절차를 찾지 못했을 때이다. 테스트 관리자는 테스트 에이전트가 지정시간동안 테스트를 완수하지 못했을 경우 에이전트에게 이번 수행한 것을 학습시키고, 다시 처음부터 테스트를 진행시킨다. 둘째, 테스트 대상에 결함이 존재하여 어떤 방법으로도 과업을 달성할 수 없는 경우이다. 테스트 시스템 담당자는 테스트를 장기간 완수하지 못한 항목에 결함 위치를 파악하기 위해 직접 기능을 점검한다. 이를 정리하면 그림 5와 같다.

점선으로 표기된 영역이 강화학습 테스트 에이전트의 영역이다. 수동 테스트일 경우 인력이 직접 담당하며, 절차형 자동화 테스트는 동일한 영역을 갖는다. 기존 자동화 방법과 달라진 점은 절차형 테스트는 테스트 절차 분석 후 테스트 스크립트를 작성해야 테스트 수행이 가능하지만, 강화학습 테스트 에이전트는 입력, 출력 값을 연결하면 즉시 수행이 가능한 점이다.

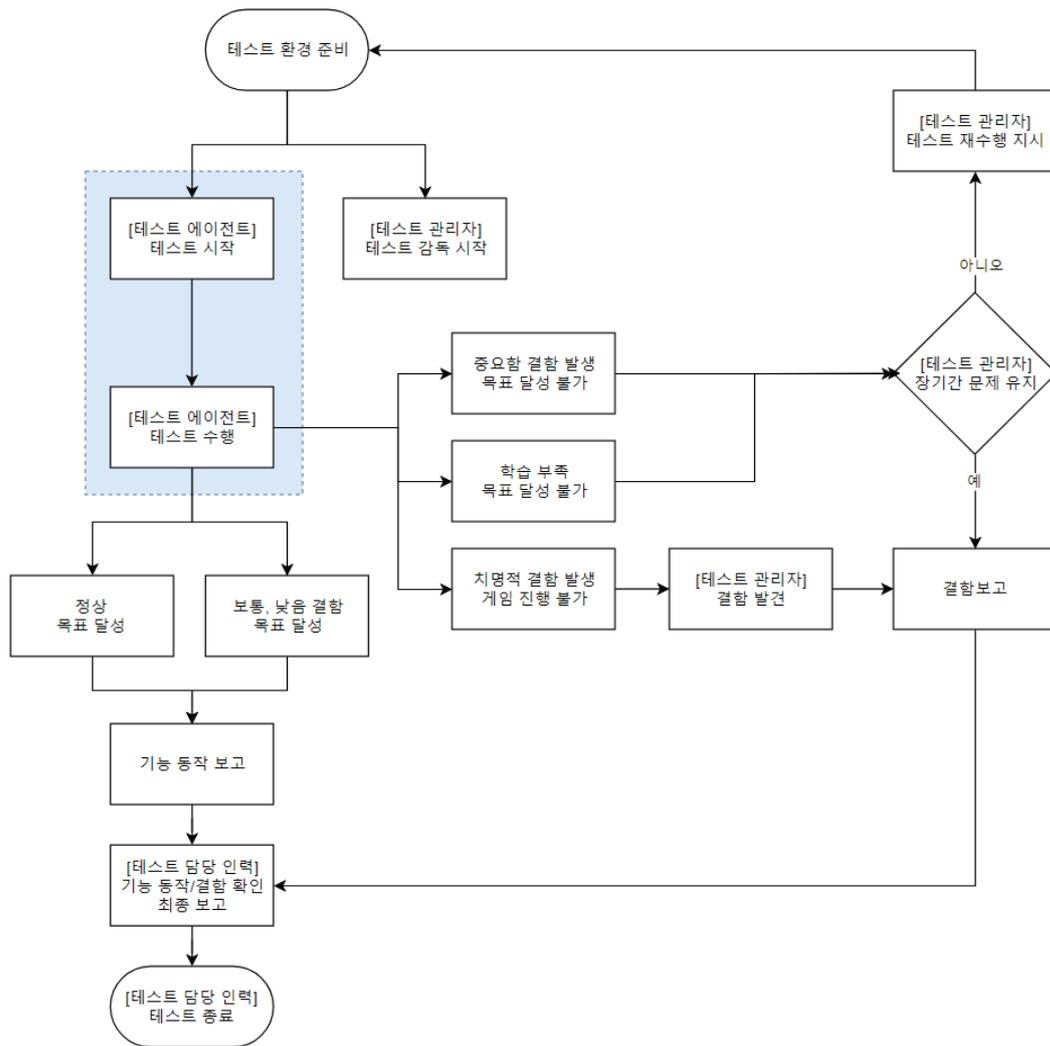


그림 5. 게임 테스트 자동화 시스템 전체 흐름

3.2 인공 신경망의 게임 조작

인공 신경망의 출력은 인공 신경망의 게임 조작이다. 이를 위해 게임의 행동집합 (Action set) 명세가 필요하다. 행동집합이란 게임에 영향을 주는 행동의 집합이며, 화면의 버튼 누름이나 키보드의 지정된 키 누름 등이 있다. 행동집합 명세 후 인공 신경망과 게임을 연결하는 방법에는 두 가지가 있다. 하나는 게임과 사전 정의된 통신 규약으로 명령을 주고받는 것이며, 다른 하나는 키보드와 마우스 입력으로 전달하는 방법이다.

통신 규약으로 명령을 주고받는 방식의 장점은 테스트 전용 기능을 게임에 추가하여 이용할 수 있는 점이다. 가령 테스트에 도움이 되는 캐릭터, 던전 등 지정된 속성 값을 강제로 변경시키는 기능을 추가하여 테스트에 이용할 수 있다. 단점은 통신 프로토콜 및 테스트용 행동 명령의 유지비용이 발생한다. 그림 6.은 아재온라인게임의 수정 상점 화면과 테스트에 필요한 행동집합이다.

게임화면	행 동 집 합
	<p>수정 상점으로 전환 귀속수정 상점으로 전환 명예 상점으로 전환 공헌 상점으로 전환 1~9번 상품 구매 1~9번 상품 정보 보기 명예 받기 기능으로 전환 다음 페이지로 이동 이전 페이지로 이동 처음 페이지로 이동 끝 페이지로 이동</p>

그림 6. 아재 온라인 게임 수정 상점 화면[17]

키보드나 마우스의 입력을 전달하는 방법은 게임에 테스트 전용 기능을 개발하고 관리할 필요가 없는 것이 장점이다. 반면 마우스와 키보드는 입력가능 수가 매우 많아 행동 집합이 매우 크기에 행동집합 작성에 주의가 필요하며, 동일 위치에 서로 다른 조작이 있을 경우 인공 신경망에게 혼선을 줄 수 있다. 그림 7.는 동 게임,

동 기능에 터치 입력으로 행동집합을 작성한 것이다. 빨간 원으로 표시된 곳이 터치 위치이다.

게임화면	행동 집합
	<p>상점전환 터치위치 5개 구매 터치위치 9개 상품정보 터치위치 9개 명예 받기 터치위치 1개 페이지 이동 터치위치 4개</p>

그림 7. 아재 온라인 게임 수정 상점 화면, 마우스 좌표 행동 집합

위 사례의 경우 통신 프로토콜을 통한 직접 입력 방법과 외부입력장치의 입력 전달 방법의 행동 집합이 동일하다. 위와 같이 테스트 특수 기능이 필요 없을 경우 외부입력을 가상으로 발생시키는

테스트 전용기능을 제공하지 않고, 화면이 복잡하지 않을 경우 외부입력 전달 방법이 유지비용이 더 적어 유리하다. 반면, 화면 구성이 복잡하고 특수기능이 필요한 경우 통신을 이용한 직접 입력방법이 유용할 것으로 보인다. 따라서 테스트 목적과 대상 및 상황에 맞추어 적합한 방식을 선택하는 것이 좋다.

3.3 인공 신경망의 게임 상태 감지

인공 신경망의 입력은 게임의 상태이다. 게임 화면을 인공 신경망에 연결하는 방법과 상태 집합 명세 후 지정된 프로토콜로 전달받는 방법이 있다. 게임화면에 직접 연결하는 방법은 유지비용이 발생하지 않는 것이 장점이다. 하지만, 대부분의 게임은 높은 화면 해상도를 사용하고 있어서 인공 신경망의 입력이 매우 커지는 문제가 발생한다. 화질을 낮추는 중간 처리 결과를 인공 신경망에 연결해도 문제

해결에는 부족하다. 그림 8.는 아재온라인 게임 상점화면의 화질을 낮춘 뒤 회색조로 변경한 것이며, 이 이미지의 인공 신경망 입력이다.

게임화면	입력 수
	66,240개

그림 8. 아재 온라인 게임 수정 상점 화면, 화면 직접 입력

화면 해상도가 320X207로 총 화소는 66,240개이다. 회색조로 1개 화소는 1개의 입력으로 전환이 가능하여 총 입력 수는 66,240개이다.

게임 상태 집합 명세 후 지정된 프로토콜로 게임 상태를 전달 받는 방법의 장점은 게임 상태를 선별해서 받을 수 있는 것이다. 단점은 게임 기능 변경에 맞춰 상태를 수정해야한다. 위 아재온라인 상점에 상태를 선별해서 받는 방법으로 정리한 것이 그림 9.이다.

게임화면	입력 수
	<p>잔여명예 1개 현재페이지 1개 전체 품목의 각 잔여수량 9개 총 11개</p>

그림 9. 아재 온라인 게임 수정 상점 화면, 상태 선별 통신[17]

프로토콜은 현재 테스트 정보를 포함할 수 있도록 표 4.의 정보가 포함되어야 한다.

표 4. 강화학습 인경신경망을 위한 최소 테스트 정보

Data1	Data2	Data3
Test ID	State ID	0~1

잔여 명예 수치를 1만으로 나누는 후 1이상은 모두 1로 취급하면 잔여명예는 1개의 인자로 충분하다. 현재 페이지도 끝 페이지 수로 나누면 1개의 인자로 충분하다. 각 상품의 잔여 수량도 같은 처리를 통해 모두 인자 1개로 사용하면 총 11개로 위상점의 상태를 표현할 수 있다.

두 방법은 장단점이 뚜렷하다. 화면을 직접 연결하는 방식은 유지비용이 발생하지 않지만, 상태 입력이 많아 학습 즉, 테스트 비용이 증가한다. 상태집합 명세 후 통신 방법은 인공 신경망이 적어 테스트 비용이 적으나, 게임 변경에 따른 유지비용이 발생한다. 따라서 테스트 자동화용 기기 수량 등 환경에 맞추어 적합한 방법을 선택해야 한다.

3.4 테스트 보상함수와 행동 선택 정책

기능 명세와 실제 동작을 기준으로 결함을 정리하면 표 5.와 같다.

표 5. 기능 동작 기준 결함 명세

상황	결함 여부
명세 기능 동작	정상
명세 기능 동작 하지 않음	결함
명세 외 기능 동작	결함
명세 외 기능 동작 하지 않음	정상

위 결함을 모두 확인하기 위해 명세 기능을 정상 동작하는 테스트 에이전트와 비정상 동작을 확인하기 위한 파괴적 행위를 하는 테스트 에이전트가 필요하다. 먼저 정상 기능 검증에 대해서 살펴보겠다. 표 6.은 아재온라인 명예상점의 구매 기능이 정상 동작 하는지 검증하는 테스트 케이스이다.

표 6. 아재 온라인 명예 상점 구매기능 테스트 케이스

제목	명예 상점 구매
설명	명예 상점에 등록된 상품을 제한 수량까지 정상적으로 구매할 수 있음
사전조건	소지 명예 1만
테스트 절차	<ol style="list-style-type: none"> 1. 첫 번째 상품을 반복 구매한다. 2. 상품이 품절되어 구매가 불가능하면 다음 상품을 구매한다. 3. 9번째 상품까지 모두 구매하였다면 다음페이지로 이동한다. 4. 1번부터 반복한다.
기대 결과	<p>모든 페이지의 모든 상품을 구매할 수 있다.</p> <p>모든 상품을 한계까지 구매하였을 때, 제한 수량과 동일하다.</p>

상점 구매 테스트에서 테스터가 달성해야 하는 과업은 상품의 구매와 모든 상품 구매 시 다음 페이지 이동이다. 보상함수를 정리하면 표 7.와 같다.

표 7. 아재 온라인 명예 상점 구매기능 테스트 보상함수 설계

<p>점수 =</p> <p>상품 수량 점수1 + 상품 수량 점수2 + + 상품 수량 점수9</p> <p>상품 수량 점수 =</p> <p>총 구매 비율 * 100 + 상품 매진 가산점 =</p> <p>(1-잔여 수량 비율) * 100 + (버림(1-잔여 수량 비율)) * 1000</p>

상품 매진 가산점이 있는 이유는 한 품목을 완전히 구매한 후 다음 상품을 구매하도록 유도하기 위함이다. 기능 정상 동작 검증에 행동 선택 정책은 Epsilon Greedy 정책이 적합하다. Epsilon Greedy 정책은 현재 가치판단의 최상위 선택지만 반복하는 것이 아닌 다른 선택지도 선택하여 인공 신경망을 학습 시키는데 유용하다. 다음으로 비정상 동작을 찾는 방안이다. 비정상 기능을 찾는 방안은 기능을 정해진 순서가 아닌 뒤죽박죽으로 조작하는 것이 필요하다. 인공 신경망은 학습 과정에서 많은 시행착오를 겪는 과정에 자연스럽게 이 비정상 기능을 찾는 테스트를 수행하게 된다. 기능이 정상동작하여 목표를 달성한 뒤의 상황에 발생하는 결함이 존재할 수 있으므로, 기능 정상동작 확인 이후에 Epsilon Greedy의 탐험 비중을 최고로 변경하여 추가 테스트를 하는 것이 좋다. 이 시간은 기능별로 다를 수 있으나, 총 테

스트 시간의 절반을 할애하면 충분하다.

3.5 테스트 에이전트 학습

테스트를 위한 인공 신경망은 다른 인공 신경망들에 비해 특수성이 존재한다. 다른 인공 신경망들은 보통 전체 과제를 확보할 수 없지만, 테스트용 인공 신경망은 테스트 대상이 제한되며 과제를 모두 확보한 상태이다. 그리고 다른 인공 신경망은 학습 과정의 결과는 버려지지만, 테스트 에이전트는 학습 과정 중의 과업 수행에 거리가 먼 무분별한 행동도 게임의 오동작을 찾는 중요한 테스트 활동이다. 이 특수성으로 인해 테스트 과제 하나당 하나의 인공 신경망을 생성하고 학습시킨다. 그리고 이미 학습을 마친 신경망이라 하여도 기능 동작에 재검수를 위해 기존 학습 자료를 삭제하고 다시 처음부터 학습시킨다. 내용을 요약하면 그림 10.과 같다.

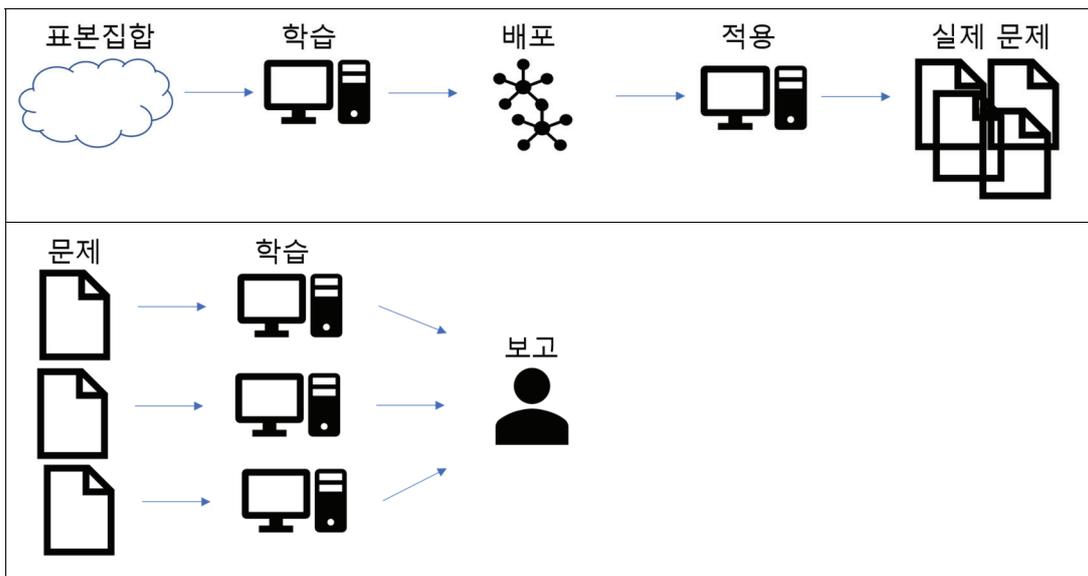


그림 10. 테스트용 기계 학습 특수성

이런 특수성으로 테스트 에이전트는 주어진 문제만을 학습하며, 학습 과정이 테스트이고, 학습을 마치면 테스트의 성공적 수행이다.

3.6 테스트 수행 및 결함 발견

테스트 수행 결과는 각종 결함의 발견 혹은 기능의 정상동작 확인이다. 결함은 표 8.와 같이 심각도에 따라 분류될 수 있다.

표 8. 소프트웨어 제품 결함, 심각도 기준 분류

심각도	설명
치명적	게임 강제 종료 게임 조작 불가 기능 비정상 동작 및 중요 데이터에 영향
중요	기능 비정상 동작 혹은 연관 기능 이용 불가
보통	기능은 동작하지만, 일부 기능이 어색함
낮음	기능이 동작하며, 사용자에게 큰 불편을 주지 않음

테스트 활동과 테스트 중에 발견하는 결함을 도식화 하면 아래 그림 11.과 같다.

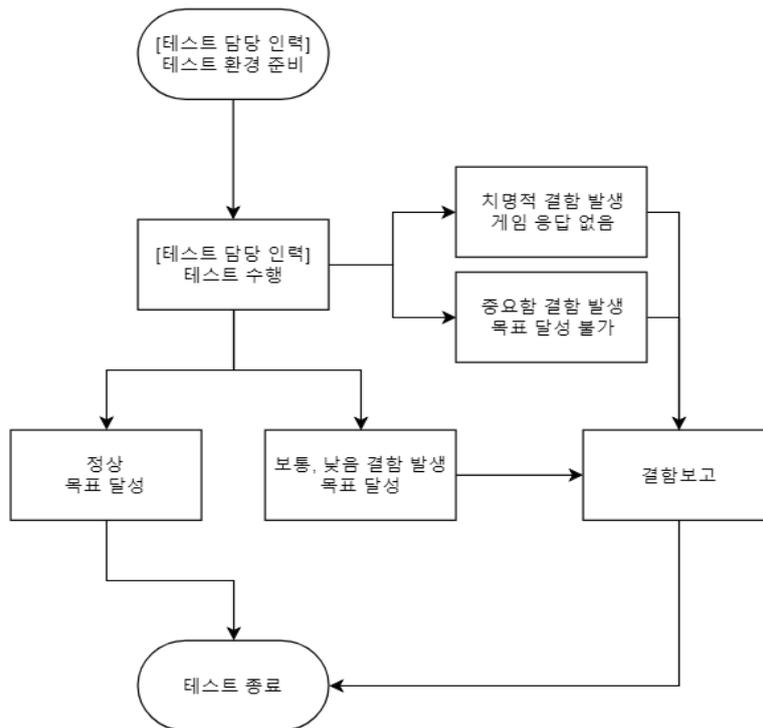


그림 11. 시스템 테스트 활동 흐름

제안하는 테스트 에이전트와 테스트 관리자의 활동에 대입하면 아래 그림 12.와

같다.

테스트 관리자는 달성해야 할 목표에 대한 자료를 가지고 있으며, 게임의 상태 변화 보고를 받아 목표 달성 여부를 판단한다. 지정시간 안에 목표를 달성하지 못할 경우 테스트 에이전트에게 재학습을 지시한다.

치명적 결함이 발생하면 게임은 외부 입력에 응답 불가 상태에 빠지거나, 입력을 받아도 내부 상태 변화가 없다. 테스트 관리자는 테스트 에이전트가 지속적으로 활동함을 전제로 하여, 게임 상태가 일정 시간동안 변화가 없을 경우 치명적 결함이 발생한 것으로 판단하여 보고할 수 있다.

중요 수준의 결함이 발생하면 에이전트는 목표를 달성할 수 없거나 과잉 목표를 달성하게 된다. 테스트 관리자는 과잉 목표 달성 시 즉각 중요 결함으로 보고할 수 있으며, 목표 미달성이 장기간 지속될 경우 테스트 담당자에게 문제 의심을 보고할 수 있다. 테스트 담당자는 문제 의심을 보고 받고 직접 테스트하여 결함 유무를 확인할 수 있다. 결함이 없을 경우 테스트 관리자의 결함 보고 시간을 조정하여 다음에 문제가 없도록 조정할 수 있다.

보통 수준의 결함은 일부 연출이 부자연스러운 것인데, 예를 들면 상품 구매 시 결과화면이 구매 성공화면이 아닌 판매화면이거나 판매 중 상품에 정상적으로 구매가 가능한 상품에 품절 표시가 같이 있는 것이다. 낮은 수준의 결함은 각종 오타자와 같은 문제이다. 이 결함들은 테스트 자동화에서 게임 변화 검수 강도와 연결된다. 이 결함을 발견하기 위해 화면의 특정 구성요소의 변경까지 감독을 하도록 구성할 수 있으며, 이는 전 테스트 자동화에 공통사항이다.

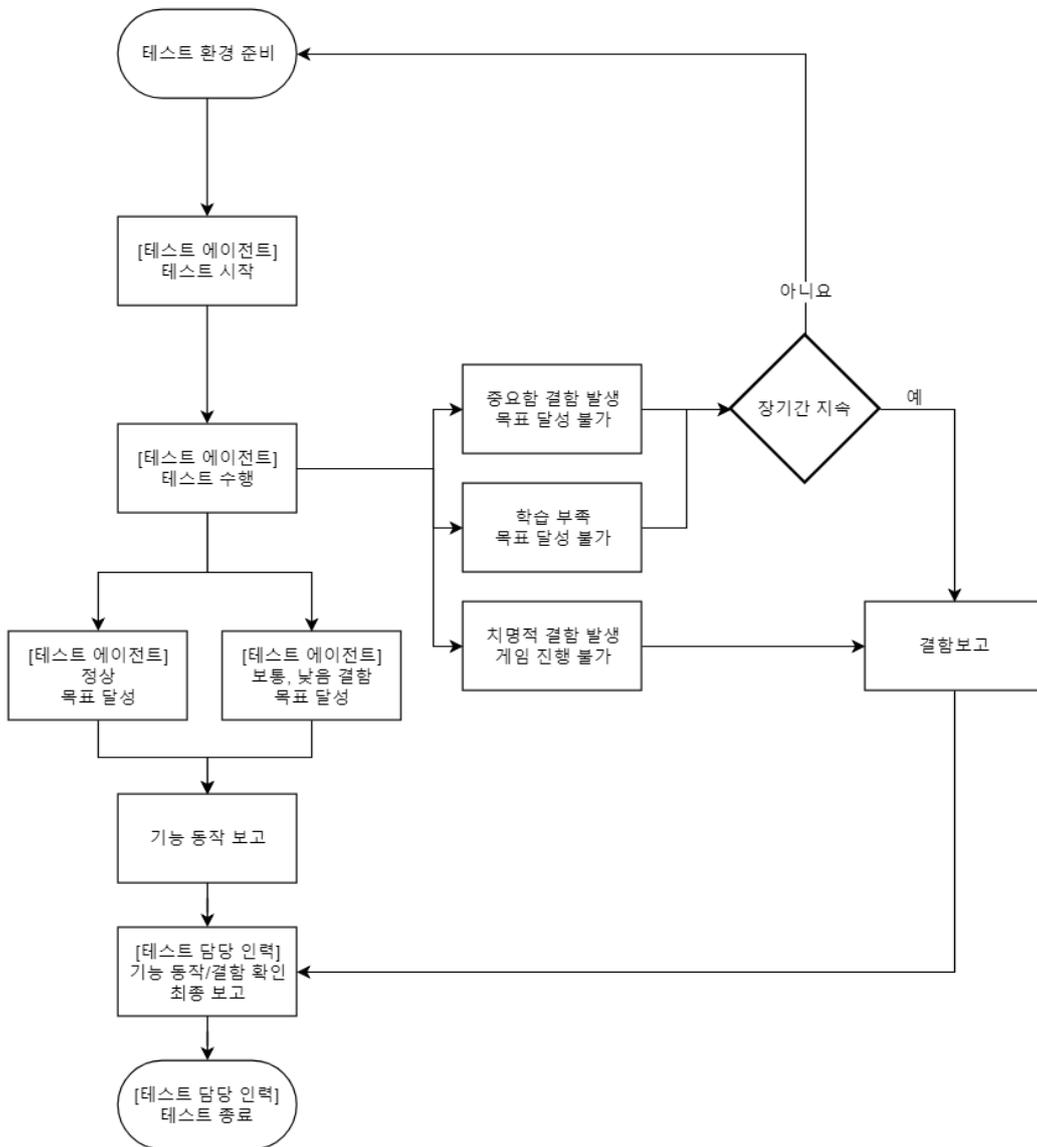


그림 12. 기계학습 테스트 에이전트 테스트 활동 흐름

3.7 기능 변경에 기존 테스트 자동화 방법과 비교

먼저 상점의 구매기능 변경 시 자동화 시스템 유지비용을 예로 차이점을 설명하겠다. 한 화면에 9개의 상품을 전시하고 총 8개의 품목이 비치된 상점에서 화면에 6개의 상품을 전시하고 총 11개의 품목을 비치하도록 변경이 일어나면, 절차형 테스트 자동화는 표 9.의 변경이 필요하다.

표 9. 절차형 테스트 자동화, 상점 구조 변경 시 테스트 자료 수정

기존	변경
1번 품목을 구매한다. 1초 대기한다. 8번 품목까지 1과 2를 반복한다.	1번 품목을 구매한다. 1초 대기한다. 6번 품목까지 1과 2를 반복한다. 2페이지로 이동한다. 1번 품목을 구매한다. 1초 대기한다. 5번 품목까지 5와 6을 반복한다.

위 표는 실제 데이터를 간략화 한 것으로 실제 자료는 더 복잡하고 변경점이 많다. 유한상태머신 인공지능 자동화 방식도 위 표 9.과 대동소이하다. 강화학습 인공지능망을 이용한 인공지능의 변경은 표 10.이다.

표 10. 강화학습 테스트 자동화, 상점 구조 변경 시 테스트 자료 수정

분류	기존	변경
인공신경망 입력	상품 상태 8개 재화 상태 1개 = 총 9개	상품 상태 11개 재화 상태 1개 상점 페이지 상태 1개 = 총 12개
인공신경망 출력	상품 구매버튼 9개 = 총 9개	상품 구매버튼 6개 페이지 전환 버튼 2개 = 총 8개

상점의 상품이 추가되어 상태가 추가되었고, 페이지 전환 버튼이 추가되어 페이지 상태가 추가되었다. 전시 품목이 9개에서 6개로 줄어들어 인공 신경망 출력이 조정되었고, 페이지 전환 버튼이 추가되었다.

유지보수 비용과 시스템 도입 비용 및 자동화 데이터 작성 비용을 정리하면 표 11.과 같다.

표 11. 테스트 자동화 기법별 게임 적용 비용

분 류	절차형	유한상태머신 인공지능	강화학습 인공 신경망 인공지능
시스템 도입 비용	상용 제품 적용 및 기능동작 확인	테스트 전용 기능 제작	강화학습, 인공 신경망 제품 적용 신경망 모델 제작
자동화 데이터 작성	테스트 절차 기록	테스트 절차에 맞춰 인공지능 상태 정의	상황 인지 요인 정의 행동 집합 정의
데이터 유지비용	테스트 절차 수정	테스트 절차 변경에 대한 인공지능 상태 수정	1상황 인지 요인 수정 행동 집합 수정

표 11.을 보면 절차형 테스트 자동화 기법과 유한상태머신 인공지능 테스트 자동화 기법은 테스트 절차에 대한 데이터가 작성되어야하고, 유지보수 되어야 한다. 반면 제안한 강화학습 인공 신경망 인공지능 방법은 테스트 절차를 인공신경망의 학습으로 대체하여 게임의 구성요소 변경에 따른 변경만이 있다. 게임의 구성요소 변경 시 절차는 함께 변경되므로, 이 비용은 전 테스트 자동화 기법에 공통사항이다. 즉, 제안한 방법은 테스트 절차의 변경에 따른 유지비용 없다. 변경사항이 많고, 콘텐츠가 많은 게임은 테스트 절차의 변경도 많이 발생 하므로 게임 개발에 강화 학습 인공신경망 인공지능 테스트 자동화가 더 효율적이다.

IV. 실험 및 분석

본 논문에서 제안한 강화학습 기반 인공지능 테스트 자동화를 검증하기 위해 세 사례를 실험했다. 각 사례에 테스트 성공 여부와 게임 기능 변경 시 제안한 방법의 영향, 절차형 테스트와 테스트자료 변동량 비교했다.

4.1 사례1 : 게임 상점 기능 테스트

4.1.1 실험 대상

아래는 상점 기능 테스트를 위한 해당 기능만 존재하는 실험용 게임이다. 그림 13.에 테스트 자동화를 적용한 뒤, 그림 14.로 수정하고 자동화 필요 자료의 변동을 확인했다. 그리고 그림 13.에서 그림 15.로 수정하고 자동화 자료 변동량을 확인했다. 그림 16.는 모든 실험에서 공통으로 사용된 부기능이다.



그림 13. 게임 상점 원본



그림 14. 게임 상점 구매 절차 변경



그림 15. 게임 상점 구매 절차 변경 없음



그림 16. 게임 상점 공통화면

사용자 입력은 마우스 클릭만 존재하며, 상품 클릭 시 물품 구매 여부를 사용자에게 물으며, 물품 부족 시 경고 메시지를 표시한다. 결함 정상 파악을 확인하기 위해 제어 변수를 추가했다. 변수 값에 의해 게임은 동작이 결정되며, 정리하면 표 12.과 같다.

표 12. 게임 상점 테스트 제어변수별 게임 동작

제어 변수	게임동작
0	정상동작
1	7번째 항목 클릭 시 게임 강제 종료
2	6번째 항목 1개 초과 구매
3	4번째 항목 1개 구매 불가

이 사례의 치명적 결함은 제어변수 1의 게임 강제 종료 사항이고, 중요 버그는 제어변수 2와 제어변수 3이다. 테스트 관리자는 매 초 게임으로부터 보고되는 동작 중임을 보고하는 사항이 없으면 치명적 문제로 파악하고 보고한다. 물품 초과 구매 시 중요 결함 발생으로 즉각 보고하며, 제어변수 3은 달성하지 못하는 과업으로 제한시간 초과 시 테스트를 종료하고 결함 보고를 한다.

비교를 위해 클릭 기능이 존재하는 절차형 테스트 자동화 시스템을 별도 제작하였으며, 동일 조건의 테스트를 수행하는 데이터를 작성했다.

4.1.2 강화학습을 이용한 테스트 자동화 실험

테스트 자동화에 이용한 인공 신경망 시스템과 모델은 아래와 같다.

- 인공신경망 시스템 : Unity ML Agents[18] 사용
- 인공신경망 모델 : Proximal Policy Optimization[19]

강화학습의 행동 집합은 마우스 좌표를 사용했다. 각 경우에 대한 게임 상태 변수와 행동 집합은 표 13.와 같다.

표 13. 게임 상점 강화학습 테스트 자동화 적용

항 목	원 본	구매 절차 변경	구매 절차 변경 없음
조작 대상	상품 위치 18개 상품 구매 확인 버튼 상품 구매 취소 버튼 상품수량부족 확인 버튼 = 총 21개	상품 위치 4개 페이지 전환 버튼 2개 상점 종류 버튼 2개 상품 구매 확인 버튼 상품 구매 취소 버튼 상품수량부족 확인 버튼 = 총 11개	상품 위치 8개 상품 구매 확인 버튼 상품 구매 취소 버튼 상품수량부족 확인 버튼 = 총 11개
게임 상태 변수	상품 수량 상태 18개 소지금 상태 1개 상품구매 화면 표시 여부 1개 상품수량부족 표시 여부 1개 선택한 상품 번호 1개 = 총 22개	상품 수량 상태 18개 소지금 상태 1개 상점 종류 상태 1개 페이지 상태 1개 상품구매 화면 표시 여부 1개 상품수량부족 표시 여부 1개 선택한 상품 번호 1개 = 총 24개	상품 수량 상태 8개 소지금 상태 1개 상품구매 화면 표시 여부 1개 상품수량부족 표시 여부 1개 선택한 상품 번호 1개 = 총 12개

테스트 에이전트의 보상함수는 표 14.과 같다. 매 행동마다 1의 감점을 주어 효율적인 행동을 찾도록 유도하였고, 유효한 동작 시 소량의 점수를 주어 상품 구매에 적절한 행동을 하도록 유도했다. 상품 구매 시 기본 가산점을 10점을 주고, 매진 시 100점의 추가점을 주어 매진까지 상품을 구매하도록 유도했다.

표 14. 게임 상점 강화학습 테스트 자동화 보상함수

<p>점수 = 행동점수 + 동작점수 + 상품 구매점수</p> <p>행동점수 = 매 행동 시 -1</p> <p>동작점수 = 잔여 물품이 있는 상품 클릭 시 +5 잔여 물품이 없는 상품 클릭 시 -5 상품수량부족 화면 표시 시 -5</p> <p>상품 구매점수 = 상품 구매 시 +10 상품 매진 시 +100</p>
--

테스트 에이전트는 총 1천회의 행동을 하며 시간을 100배 가속하였고, 모든 상품 구매 시 성공으로 간주한다. 그림 17.는 정상동작의 학습 기록이다.

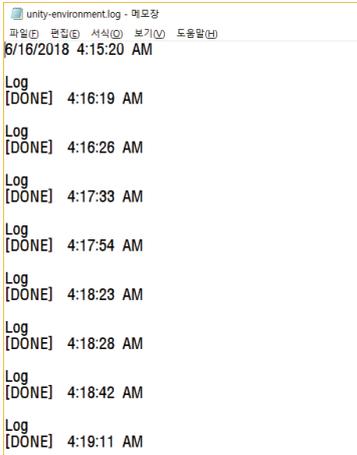


그림 17. 정상동작 기록

상품 전체 구매하였을 경우 “[DONE]”으로 시간과 함께 기록을 남겼고, 총 10회 기록이 남으면 테스트를 종료했다. 이 시간을 테스트 기준 시간으로 선정하고 타 테스트는 이 시간의 두 배 시간을 감독하며, 초과 시 오류로 간주했다. 테스트 수

행 결과는 표 15.와 같다.

표 15. 게임 상점 강화학습 테스트 자동화 적용 결과

항 목	원 본	구매 절차 변경	구매 절차 변경 없음
정상 동작	6분 7초 완료	11분 12초 완료	4분 37초 완료
치명적 결함 사례	11초 발견	4분 31초 발견	9초 발견
물품 수량 과다	1분 51초 발견	5분 1초 발견	49초 발견
물품 소량 과소	제한 시간 초과	제한 시간 초과	제한 시간 초과

4.1.3 절차형 테스트 자동화 실험

절차형 테스트 자동화를 위한 테스트 자료를 정리하면 표 16.와 같다.

표 16. 게임 상점 절차형 테스트 자동화 적용

항 목	원 본	구매 절차 변경	구매 절차 변경 없음
조작 대상	상품 위치 18개 상품 구매 확인 버튼 상품 구매 취소 버튼 상품수량부족 확인 버튼 = 총 21개	상품 위치 4개 페이지 전환 버튼 2개 상점 종류 버튼 2개 상품 구매 확인 버튼 상품 구매 취소 버튼 상품수량부족 확인 버튼 = 총 11개	상품 위치 8개 상품 구매 확인 버튼 상품 구매 취소 버튼 상품수량부족 확인 버튼 = 총 11개
테스트 절차	상품 18개 순차 구매 초과 구매 시도 = 총 21구문	상점 1페이지 상품 순차 구매 페이지 전환 상점 2페이지 상품 순차 구매 상점 전환 상점 1페이지 상품 순차 구매 페이지 전환 상점 2페이지 상품 순차 구매 페이지 전환 상점 3페이지 상품 순차 구매 = 총 68구문	상품 8개 순차 구매 초과 구매 시도 = 총 21구문

4.1.4 테스트 자료 변동량 비교

이 사례에서 강화학습 테스트 에이전트를 이용한 자동화 필요 자료 변동량을 정리하면 표 17.과 같다.

표 17. 게임 상점 강화학습 테스트 자동화 변동내용

항 목	상품 절차 변경의 변동량	상품 절차 변경 없음의 변동량
조작 대상	상품위치 14개 삭제 상품위치 4개 수정 페이지전환 버튼 2개추가 상품종류 버튼 2개추가 = 총 14개 삭제, 4개 수정, 4개추가	상품 위치 10개 삭제, 8개 수정 = 총 10개 삭제, 8개 수정
게임 상태 변수	상점 종류 상태 1개추가 페이지 상태 1개추가 = 총 2개추가	상품 품목 수 수정 1개 = 총 1개 수정
보상합수	없음	

절차형 테스트 자동화에서 수정된 테스트 자료를 정리하면 표 18.과 같다.

표 18. 게임 상점 절차형 테스트 자동화 변동내용

항 목	상품 절차 변경의 변동량	상품 절차 변경 없음의 변동량
조작 대상	상품위치 14개 삭제 상품위치 4개 수정 페이지전환 버튼 2개추가 상품종류 버튼 2개추가 = 총 14개 삭제, 4개 수정, 4개추가	상품 위치 10개 삭제, 8개 수정 = 총 10개 삭제, 8개 수정
테스트 절차	상점 범주 단위 절차 관련 31개 추가 상점 페이지 단위 절차 관련 18 개추가 전시 품목 수 수정 1개 = 총 49개추가, 1개 수정	상품 품목 수 수정 1개 = 총 1개 수정

위 내용을 정리하여 이 사례의 게임 변경에 따른 테스트 자동화 자료 변동량을 비교하면 표 19.과 같다.

표 19. 게임 상점 테스트 자동화 기법별 변동내용 비교

항 목	강화학습 테스트 자동화	절차형 테스트 자동화
구매 절차 변경 공통 수정 사항	조작 대상 = 총 14개 삭제, 4개 수정, 4개추가 (변경 22개)	
구매 절차 변경 개별 수정 사항	게임 상태 변수 2개추가 (변경 2개)	절차 구문 49개추가, 1개 수정 (변경 50개)
구매 절차 변경 없음 공통 수정 사항	조작 대상, 상품 품목 수 수정 = 총 10개 삭제, 9개 수정 (변경 19개)	
구매 절차 변경 없음 개별 수정 사항	없음	

게임 상점 상품 절차 변경에서 공통 수정 사항을 제외하고 절차형 테스트 자동화는 50개의 변경이 필요했고, 강화학습 테스트 자동화는 2개의 변경이 필요했다.

4.2 사례2 : 게임 던전 입장 테스트

4.2.1 실험 대상

아래는 게임 던전 입장 기능 테스트를 위한 해당 기능만 존재하는 실험용 게임이다. 그림 18.에 테스트 자동화를 적용한 뒤, 그림 19.과 그림 20.로 구성되도록 수정하고 자동화 필요 자료의 변동을 확인했다. 그리고 그림 18.에서 그림 21.로 수정하고 자동화 자료 변동량을 확인했다. 그림 22.은 모든 실험에서 공통으로 사용된 부기능이다.



그림 18. 던전 입장 원본

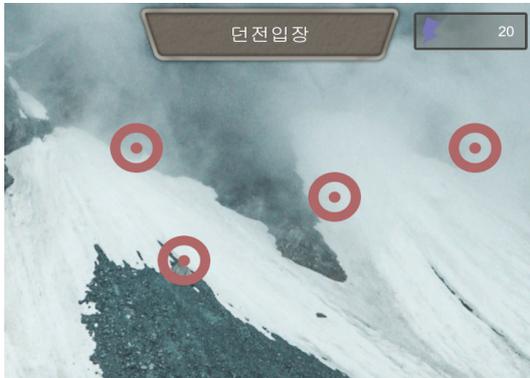


그림 19. 던전 입장 절차 변경1



그림 20. 던전 입장 절차 변경2



그림 21. 던전 입장 절차 변경 없음

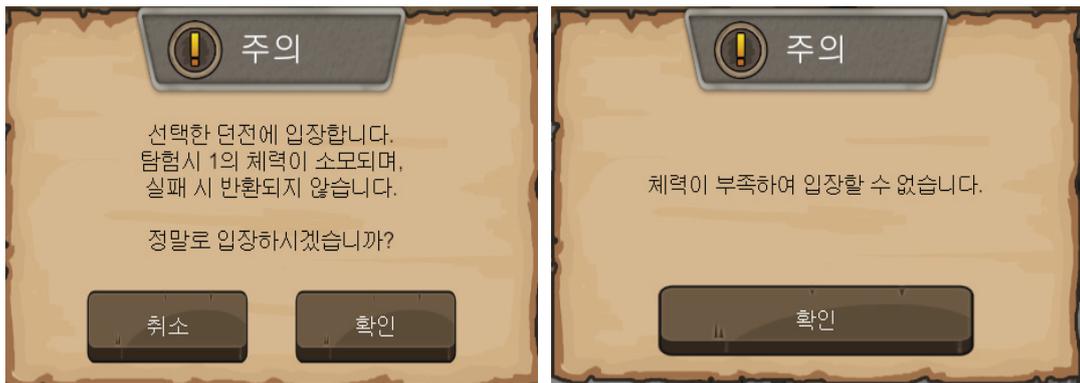


그림 22. 던전 입장 공통화면

사용자 입력은 마우스 클릭만 존재하며, 던전 클릭 시 입장 구매 여부를 사용자에게 물으며, 체력 부족 시 경고 메시지를 표시한다. 결함 정상 파악을 확인하기 위해 제어 변수를 추가했다. 변수 값에 의해 게임은 동작이 결정되며, 정리하면 표 20.와 같다.

표 20. 던전 입장 테스트 제어변수별 게임 동작

제어 변수	게임동작
0	정상동작
1	2-2 스테이지 클릭 시 게임 강제 종료
2	1-2 스테이지 체력이 0이어도 입장
3	3-1 스테이지 항목 입장 불가

절차형 테스트 자동화 시스템은 사례1과 동일하다. 게임은 테스트 관리자에게 보고 항목이 변동되며, 보고 항목은 입장하는 던전 번호와 체력 차감, 현재 체력이다. 이 사례의 치명적 결함은 제어변수 1의 게임 강제 종료 사항이고, 중요 버그는 제어변수 2와 제어변수 3이다. 테스트 관리자는 매 초 게임으로부터 보고되는 동작 중임을 보고하는 사항이 없으면 치명적 문제로 파악하고 보고한다. 체력이 음수로 변경되는 경우와 입장하지 못한 던전이 있을 경우 중요 결함으로 판단하여 보고한다.

4.2.2 강화학습을 이용한 테스트 자동화 실험

인공신경망관련 사항은 사례1과 동일한 조건이며, 강화학습의 행동 집합은 마우스 좌표를 사용했다. 총 1천회의 행동을 하며 시간을 100배 가속하였고, 각 경우에 대한 게임 상태 변수와 행동 집합은 표 21.과 같다.

표 21. 던전 입장 강화학습 테스트 자동화 적용

항 목	원 본	던전 입장 절차 변경	던전 입장 절차 변경 없음
조작 대상	스테이지 위치 12개 던전 입장 확인 버튼 던전 입장 취소 버튼 체력 부족 확인 버튼 = 총 15개	던전 위치 4개 스테이지 위치 3개 던전 입장 확인 버튼 던전 입장 취소 버튼 체력 부족 확인 버튼 = 총 10개	스테이지 위치 24개 던전 입장 확인 버튼 던전 입장 취소 버튼 체력 부족 확인 버튼 = 총 27개
게임 상태 변수	체력 상태 1개 스테이지 입장 상태 12개 던전 입장 확인 화면 표시 여부 1개 체력 부족화면 표시 여부 1개 선택한 스테이지 번호 1개 = 총 16개	체력 상태 1개 스테이지 입장 상태 12개 선택 던전 번호 1개 던전 입장 확인 화면 표시 여부 1개 체력 부족화면 표시 여부 1개 선택한 스테이지 번호 1개 = 총 17개	체력 상태 1개 스테이지 입장 상태 24개 던전 입장 확인 화면 표시 여부 1개 체력 부족화면 표시 여부 1개 선택한 스테이지 번호 1개 = 총 28개

테스트 에이전트의 보상함수는 표 22.과 같다. 사례1과 같이 동작 점수를 주었으며, 각 스테이지의 최초 입장에만 가산점을 주어 모든 던전을 입장하도록 유도했다.

표 22. 던전 입장 강화학습 테스트 자동화 보상함수

점수 = 행동점수 + 동작점수 + 던전 입장 점수 행동점수 = 매 행동 시 -1 동작점수 = 입장 기록이 없는 스테이지 클릭 시 +5 입장 기록이 있는 스테이지 클릭 시 -5 던전 입장 점수 = (올림(입장횟수 / 100)) * 100

모든 던전에 입장하였을 경우 사례1과 같이 “[DONE]”으로 시간과 함께 기록을 남겼고, 총 10회 기록이 남으면 테스트를 종료했다. 이 시간을 기준시간으로 선정하고 타 테스트는 이 시간의 두 배 시간을 감독하며, 초과 시 오류로 간주했다. 테스트 결과는 표 23.와 같다.

표 23. 던전 입장 강화학습 테스트 자동화 적용 결과

항 목	원 본	던전 입장 절차 변경	던전 입장 절차 변경 없음
정상 동작	1분 41초 완료	7분 55초 완료	3분 27초 완료
치명적 결함 사례	5초 발견	1분 28초 발견	2초 발견
체력 0에 입장	12초 발견	3분 4초 발견	39초 발견
입장 불가	제한 시간 초과	제한 시간 초과	제한 시간 초과

4.2.3 절차형 테스트 자동화 실험

절차형 테스트 자동화를 위한 테스트 자료를 정리하면 표 24.과 같다.

표 24. 던전 입장 절차형 테스트 자동화 적용

항 목	원 본	던전 입장 절차 변경	던전 입장 절차 변경 없음
조작 대상	스태이지 위치 12개 던전 입장 확인 버튼 던전 입장 취소 버튼 체력 부족 확인 버튼 = 총 15개	던전 위치 4개 스태이지 위치 3개 던전 입장 확인 버튼 던전 입장 취소 버튼 체력 부족 확인 버튼 = 총 10개	스태이지 위치 24개 던전 입장 확인 버튼 던전 입장 취소 버튼 체력 부족 확인 버튼 = 총 27개
테스트 절차	스태이지 12개 입장 체력 0까지 임의 던전 입장 스태이지 12개 초과 입장 시도 = 총 21구문	던전 1클릭 스태이지 3개 입장 던전 4까지 반복 체력 0까지 임의 던전 입장 던전 1클릭 스태이지 3개 초과 입장 시도 던전 4까지 반복 = 총 34구문	스태이지 24개 입장 체력 0까지 임의 던전 입장 스태이지 24개 초과 입장 시도 = 총 21구문

4.2.4 테스트 자료 변동량 비교

이 사례에서 강화학습 테스트 에이전트를 이용한 자동화 필요 자료 변동량을 정리하면 표 25.와 같다.

표 25. 던전 입장 강화학습 테스트 자동화 변동내용

항 목	던전 입장 절차 변경의 변동량	던전 입장 절차 변경 없음의 변동량
조작 대상	스태이지 위치 9개 삭제 스태이지 위치 3개 수정 던전 위치 4개추가 = 총 9개 삭제, 3개 수정, 4개추가	스태이지 위치 수정 12개 스태이지 위치 추가 12개 = 총 12개 수정, 12개추가
게임 상태 변수	선택 던전 번호 1개 = 총 1개추가	스태이지 수 수정 1개 = 총 1개 수정
보상함수	없음	

절차형 테스트 자동화에서 수정된 테스트 자료를 정리하면 표 26.와 같다.

표 26. 던전 입장 절차형 테스트 자동화 변동내용

항 목	던전 입장 절차 변경의 변동량	던전 입장 절차 변경 없음의 변동량
조작 대상	스태이지 위치 9개 삭제 스태이지 위치 3개 수정 던전 위치 4개추가 = 총 9개 삭제, 3개 수정, 4개추가	스태이지 위치 수정 12개 스태이지 위치 추가 12개 = 총 12개 수정, 12개추가
테스트 절차	던전 선택 절차 관련 13개추가 = 총 13개추가	스태이지 수 수정 1개 = 총 1개 수정

위 내용을 정리하여 이 사례의 게임 변경에 따른 테스트 자동화 자료 변동량을 비교하면 표 27.과 같다.

표 27. 던전 입장 테스트 자동화 기법별 변동내용 비교

항 목	강화학습 테스트 자동화	절차형 테스트 자동화
던전 입장 절차 변경 공통 수정 사항	조작 대상 = 총 9개 삭제, 3개 수정, 4개추가 (변경 16개)	
던전 입장 절차 변경 개별 수정 사항	게임 상태 변수 1개추가 (변경 1개)	절차 구문 13개추가 (변경 13개)
던전 입장 절차 변경 없음 공통 수정 사항	조작 대상, 스태이지 수 수정 = 총 13개 수정, 12개추가 (변경 25개)	
던전 입장 절차 변경 없음 개별 수정 사항	없음	

던전 입장 절차 변경에서 공통 수정 사항을 제외하고 절차형 테스트 자동화는 13개의 변경이 필요했고, 강화학습 테스트 자동화는 1개의 변경이 필요했다.

4.3 사례3 : 게임 스킬 장착 테스트

4.3.1 실험 대상

아래는 스킬 장착 테스트를 위한 해당 기능만 존재하는 실험용 게임이다. 그림 23.에 테스트 자동화를 적용한 뒤, 그림 24.로 구성되도록 수정하고 자동화 필요 자료의 변동을 확인했다. 그리고 그림 23.에서 그림 25.로 수정하고 자동화 자료 변동량을 확인했다. 그림 26.는 모든 실험에서 공통으로 사용된 부기능이다.



그림 23. 스킬 장착 원본



그림 24. 스킬 장착 절차 변경



그림 25. 스킬 장착 절차 변경 없음



그림 26. 스킬 장착 공통화면

사용자 입력은 마우스 클릭만 존재하며, 장착 스킬을 클릭하면 해제되고, 전체 스킬을 클릭하면 장착된다. 장착 최대 수량인 상태에서 전체 스킬을 클릭하면 장착 불가 경고 메시지를 표시한다. 결함 정상 파악을 확인하기 위해 제어 변수를 추가했다. 변수 값에 의해 게임은 동작이 결정되며, 정리하면 표 28.과 같다.

표 28. 던전 입장 테스트 제어변수별 게임 동작

제어 변수	게임동작
0	정상동작
1	공격 전체스킬 2번째 항목 클릭 시 게임 강제 종료
2	방어 장착스킬 수량 1개 초과 장착
3	패시브 전체스킬 3번째 항목 장착 불가

절차형 테스트 자동화 시스템은 사례1과 동일하다. 게임은 테스트 관리자에게 보고 항목이 변동되며, 보고 항목은 장착 스킬 종류와 장착 스킬 번호, 장착 중 스킬 수이다.

이 사례의 치명적 결함은 제어변수 1의 게임 강제 종료 사항이고, 중요 버그는 제어변수 2와 제어변수 3이다. 테스트 관리자는 매 초 게임으로부터 보고되는 동작 중임을 보고하는 사항이 없으면 치명적 문제로 파악하고 보고한다. 장착 가능 수량을 초과하는 경우와 장착하지 못한 스킬이 있을 경우 중요 결함으로 판단하여 보고한다.

4.3.2 강화학습을 이용한 테스트 자동화 실험

인공신경망관련 사항은 사례1과 동일한 조건이며, 강화학습의 행동 집합은 마우스 좌표를 사용했다. 총 1천회의 행동을 하며 시간을 100배 가속하였고, 1 각 경우에 대한 게임 상태 변수와 행동 집합은 표 29.과 같다.

표 29. 스킬 장착 강화학습 테스트 자동화 적용

항 목	원 본	스킬 장착 절차 변경	스킬 장착 절차 변경 없음
조작 대상	공격 장착스킬 위치 3개 공격 전체스킬 위치 6개 방어 장착스킬 위치 3개 방어 전체스킬 위치 6개 패시브 장착스킬 위치 3개 패시브 장착스킬 위치 6개 최대장착 확인 위치 1개 = 총 28개	스킬 범주 버튼 3개 장착스킬 위치 3개 전체스킬 위치 6개 최대장착 확인 위치 1개 = 총 13개	공격 장착스킬 위치 2개 공격 전체스킬 위치 5개 방어 장착스킬 위치 3개 방어 전체스킬 위치 5개 패시브 장착스킬 위치 4개 패시브 장착스킬 위치 5개 최대장착 확인 위치 1개 = 총 25개
게임 상태 변수	공격 스킬 장착 기록 6개 방어 스킬 장착 기록 6개 패시브 스킬 장착 기록 6개 공격 스킬 장착 수량 1개 방어 스킬 장착 수량 1개 패시브 스킬 장착 수량 1개 최대장착 화면 표시 상태 1개 = 총 22개	공격 스킬 장착 기록 6개 방어 스킬 장착 기록 6개 패시브 스킬 장착 기록 6개 공격 스킬 장착 수량 1개 방어 스킬 장착 수량 1개 패시브 스킬 장착 수량 1개 최대장착 화면 표시 상태 1개 스킬 범주 상태 1개 = 총 23개	공격 스킬별 장착 기록 5개 방어 스킬별 장착 기록 5개 패시브 스킬별 장착 기록 5개 공격 스킬 장착 수량 1개 방어 스킬 장착 수량 1개 패시브 스킬 장착 수량 1개 최대장착 화면 표시 상태 1개 = 총 19개

테스트 에이전트의 보상함수는 표 30.와 같다. 사례1과 같이 동작 점수를 주었으며, 각 스킬의 최초 장착에만 가산점을 주어 모든 스킬을 장착하도록 유도했다.

모든 스킬을 장착하였을 경우 사례1과 같이 “[DONE]”으로 시간과 함께 기록을 남겼고, 총 10회 기록이 남으면 테스트를 종료했다. 이 시간을 기준시간으로 선정하고 타 테스트는 이 시간의 두 배 시간을 감독하며, 초과 시 오류로 간주했다. 테스트 결과는 표 31.과 같다.

표 30. 스킬 장착 강화학습 테스트 자동화 보상함수

<p>점수 = 행동점수 + 동작점수 + 스킬 장착 점수</p> <p>행동점수 = 매 행동 시 -1</p> <p>동작점수 = 최대 장착 화면 표시 시 -5</p> <p>스킬 장착 점수(공격, 방어, 패시브) = (올림(장착횟수 / 100)) * 100</p>

표 31. 스킬 장착 강화학습 테스트 자동화 적용 결과

항 목	원 본	스킬 장착 절차 변경	스킬 장착 절차 변경 없음
정상 동작	2분 1초 완료	6분 14초 완료	2분 36초 완료
치명적 결함 사례	1초 발견	2초 발견	2초 발견
초과 장착	4초 발견	1분 12초 발견	7초 발견
장착 불가	제한 시간 초과	제한 시간 초과	제한 시간 초과

4.3.3 절차형 테스트 자동화 실험

절차형 테스트 자동화를 위한 테스트 자료를 정리하면 표 32.과 같다.

표 32. 스킬 장착 절차형 테스트 자동화 적용

항 목	원 본	스킬 장착 절차 변경	스킬 장착 절차 변경 없음
조작 대상	공격 장착스킬 위치 3개 공격 전체스킬 위치 6개 방어 장착스킬 위치 3개 방어 전체스킬 위치 6개 패시브 장착스킬 위치 3개 패시브 장착스킬 위치 6개 최대장착 확인 위치 1개 = 총 28개	스킬 범주 버튼 3개 장착스킬 위치 3개 전체스킬 위치 6개 최대장착 확인 위치 1개 = 총 13개	공격 장착스킬 위치 2개 공격 전체스킬 위치 5개 방어 장착스킬 위치 3개 방어 전체스킬 위치 5개 패시브 장착스킬 위치 4개 패시브 장착스킬 위치 5개 최대장착 확인 위치 1개 = 총 25개
테스트 절차	공격 전체 장착 확인 방어 전체 장착 확인 패시브 전체 장착 확인 공격 초과 장착 시도 방어 초과 장착 시도 패시브 초과 장착 시도 = 총 44구문	공격 범주 선택 전체 스킬 장착 확인 초과 장착 시도 방어카테고리 선택 전체 스킬 장착 확인 초과 장착 시도 패시브 범주 선택 전체 스킬 장착 확인 초과 장착 시도 = 총 46구문	공격 전체 장착 확인 방어 전체 장착 확인 패시브 전체 장착 확인 공격 초과 장착 시도 방어 초과 장착 시도 패시브 초과 장착 시도 = 총 44구문

4.3.4 테스트 자료 변동량 비교

이 사례에서 강화학습 테스트 에이전트를 이용한 자동화 필요 자료 변동량을 정리하면 표 33.와 같다.

표 33. 스킬 장착 강화학습 테스트 자동화 변동내용

항 목	스킬 장착 절차 변경의 변동량	스킬 장착 절차 변경 없음의 변동량
조작 대상	공격 장착 스킬 위치 3개 수정 방어 장착스킬 위치 3개 삭제 패시브 장착스킬 위치 3개 삭제 공격 전체스킬 위치 6개 수정 방어 전체스킬 위치 6개 삭제 패시브 전체스킬 위치 6개 삭제 스킬 범주 위치 3개추가 = 총 18개 삭제, 3개추가, 9개 수정	공격 장착스킬 위치 1개 삭제 패시브 장착스킬 위치 1개추가 공격 전체스킬 위치 1개 삭제 방어 전체스킬 위치 1개 삭제 패시브 전체스킬 위치 1개 삭제 = 총 4개 삭제, 1개추가
게임 상태 변수	스킬 범주 상태 1개추가 = 총 1개추가	공격 장착스킬 수량 1개 수정 패시브 장착스킬 수량 1개 수정 공격 전체스킬 수량 1개 수정 방어 전체스킬 수량 1개 수정 패시브 전체스킬 수량 1개 수정 = 총 5개 수정
보상합수	없음	

절차형 테스트 자동화에서 수정된 테스트 자료를 정리하면 표 34.과 같다.

표 34. 스킬 장착 절차형 테스트 자동화 변동내용

항 목	스킬 장착 절차 변경의 변동량	스킬 장착 절차 변경 없음의 변동량
조작 대상	공격 장착 스킬 위치 3개 수정 방어 장착스킬 위치 3개 삭제 패시브 장착스킬 위치 3개 삭제 공격 전체스킬 위치 6개 수정 방어 전체스킬 위치 6개 삭제 패시브 전체스킬 위치 6개 삭제 스킬 범주 위치 3개추가 = 총 18개 삭제, 3개추가, 9개 수정	공격 장착스킬 위치 1개 삭제 패시브 장착스킬 위치 1개추가 공격 전체스킬 위치 1개 삭제 방어 전체스킬 위치 1개 삭제 패시브 전체스킬 위치 1개 삭제 = 총 4개 삭제, 1개추가
테스트 절차	범주 전환 절차 관련 6개추가, 4개 삭제 = 총 6개추가, 4개 삭제	공격 장착스킬 수량 1개 수정 패시브 장착스킬 수량 1개 수정 공격 전체스킬 수량 1개 수정 방어 전체스킬 수량 1개 수정 패시브 전체스킬 수량 1개 수정 = 총 5개 수정

위 내용을 정리하여 이 사례의 게임 변경에 따른 테스트 자동화 자료 변동량을 비교하면 표 35.와 같다.

표 35. 스킬 장착 테스트 자동화 기법별 변동내용 비교

항 목	강화학습 테스트 자동화	절차형 테스트 자동화
스킬 장착 절차 변경 공통 수정 사항	조작 대상 = 총 18개 삭제, 3개추가, 9개 수정 (변경 30개)	
스킬 장착 절차 변경 개별 수정 사항	게임 상태 변수 1개 추가 (변경 1개)	절차 구문 6개추가, 4개 삭제 (변경 10개)
스킬 장착 절차 변경 없음 공통 수정 사항	조작 대상, 스킬 수량 수정 = 총 4개 삭제, 1개추가, 5개 수정 (변경 10개)	
스킬 장착 절차 변경 없음 개별 수정 사항	없음	

스킬 장착 절차 변경에서 공통 수정 사항을 제외하고 절차형 테스트 자동화는 10개의 변경이 필요했고, 강화학습 테스트 자동화는 1개의 변경이 필요했다.

4.4 종합 분석

위 사례를 종합하면 표 36.와 같다.

표 36. 전체 사례 테스트 자료 변동량 비교

	기능 절차 변경의 변동량 비교 (강화학습 : 절차형)	기능 절차 변경 없음의 변동량 비교
사례1	2 : 50, 기존 대비 4%	동일
사례2	1 : 13, 기존 대비 약7.6%	동일
사례3	1 : 10, 기존 대비 10%	동일
평균	약 7.2%	동일

세 사례에서 공통적으로 기능의 절차 변경에서 테스트 자료 변동량에서 평균 약 7.2%의 차이를 보였고, 기능의 절차 변경이 없을 경우 기존 자동화 기법과 비교해

동일 수치를 기록했다. 표 36.의 결과를 통해 제안한 강화학습을 이용한 시스템 테스트 자동화 기법은 제시한대로 절차에 관한 테스트 관리 비용을 강화학습으로 대체하였음을 알 수 있다.

V. 결론

기존의 테스트 자동화 기법들은 수정이 빈번하고 테스트 대상이 많은 게임에 적합하지 않았다. 절차형 테스트 자동화 기법은 자동화에 필요한 자료가 많았고, 유지보수 비용이 컸다. 유한상태머신 인공지능 자동화 기법은 적용대상이 한정적이며, 개발비용과 테스트 시스템 유지보수 비용이 컸다.

본 논문에서는 기존의 테스트 자동화 방법이 갖고 있는 높은 유지비용, 그리고 테스트 자동화 시스템이 게임 콘텐츠 시스템에 종속성이 높은 문제를 해결하기 위하여 강화학습을 이용한 테스트 자동화를 제안하였다. 기계학습으로 테스트를 수행하기 위한 테스트 자동화 시스템 구조를 고안하였고, 테스트를 수행하도록 인공 신경망이 학습할 방안을 마련하였으며, 인공 신경망의 게임 테스트 수행 방안을 제시했다.

제안한 테스트 자동화 기법으로 게임의 일부 기능을 대상으로 실험을 진행하였고, 절차 변경이 존재할 경우 절차형 테스트 자동화 기법 대비 약 7.2%의 변경으로 테스트 자동화가 가능했다.

본 논문의 의의는 게임 개발의 특수성을 고려한 테스트 자동화 구조를 제시하여 게임 테스트 비용을 절감하고 게임 품질 향상에 기여하는 것이다. 빈번한 기능 개발을 하는 검은사막과 같은 PC온라인 게임이나 모바일 온라인 게임 및 제작 단계에 있는 게임에 적용할 경우, 기존의 테스트 자동화 기법에 비해 테스트 자동화 자료 유지보수 비용 절감 효과를 얻어 개발비용을 절감할 수 있을 것으로 기대된다.

향후 연구 과제로는 본 논문에서 제안한 기법을 기반으로 테스트 적용 가능한 게임 장르의 확장과 복합적인 기능에 테스트 자동화 방법에 대한 연구가 있으며, 절차 변경이 적을 경우에 효율이 높은 자동화 기법 연구가 필요하다.

VI. 참고문헌

- [1] Technical Reports, Korea Creative Content Agency, Vol.1, Jan. 2017.
- [2] Ung-Jin Son, A Study on the Automatic Generation of Test Case Based on Source Code for Quality Improvement, Journal of IEEE Korea Council, June, 2015.
- [3] J.H. Lee, H.C. Jung, S.G. Kang, H.W. Cho, J.M. Lee, M.J. Kang, C.I. Park, H.J. Oh, S.J. Kim, Games QA using Checklists Technique, Proceedings of the Korean Society of Computer Information Conference, July, 2014.
- [4] G. J. Myers, The Art of Software Testing, John Willy & Sons, 1979.
- [5] Y.G. Han, H.N. Lee, C.H. Park, A Study on Technological Level and Cases of Testing Automation for Mobile Applications, Journal of Korea Safety Management & Science, Vol. 17 No. 1, 2015.
- [6] G. Lample, D.S. Chaplot, “Playing FPS Games with Deep Reinforcement Learning,” Proceedings of 31st AAAI Conference on Artificial Intelligence, 2017.
- [7] Spiral model,
https://en.wikipedia.org/wiki/Spiral_model
- [8] Black Desert Online,
<http://black.game.daum.net>
- [9] 권원일, 이현주, 최승희, 이승호, 박은영, 조현길, 개발자도 알아야 할 소프트웨어 테스트 실무, STA, 2010.
- [10] V 모델,
https://ko.wikipedia.org/wiki/V_모델
- [11] J.W. Kim, “Automate testing of mobile games,” Nexon Developer Conference, 2016.
- [12] Appium,
<http://appium.io/>
- [13] 염선화 , 최홍석 , 전미숙. 2011. 사례발표 : C++언어로 개발된 게임 플랫폼의 테스트 자동화 사례. 정보처리학회지, 18(1): 76-83
- [14] Hunjoo Lee, Yong-Woo Jung, Bum-Hyun Lim, Kwang-Hyun Shim, “Automated Testing of Online Game Servers by Agents”, The Journal of the KICS, May, 2006.
- [15] 정훈영, “A FSM based test automation system for online game server”, 고려대학교 대학원 컴퓨터정보통신학과, 2014. 2
- [16] Reinforcement learning,
https://en.wikipedia.org/wiki/Reinforcement_learning
- [17] 아재 온라인 - 게임가이드,
<http://aj.ongate.com>

- [18] Unity Machine Learning Agents,
[https://github.com/ Unity-Technologies/ml-agents](https://github.com/Unity-Technologies/ml-agents)
- [19] Proximal Policy Optimization,
<https://blog. openai. com/openai-baselines-ppo>