

---

저자 (Authors)	변혁, 최은만 Hyuk Byun, Eun Man Choi
출처 (Source)	<a href="#">한국정보과학회 학술발표논문집</a> , 2020.12, 201-203 (3 pages)
발행처 (Publisher)	<a href="#">한국정보과학회</a> The Korean Institute of Information Scientists and Engineers
URL	<a href="http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE10529591">http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE10529591</a>
APA Style	변혁, 최은만 (2020). GVG-AI 프레임워크 기반 에이전트를 활용한 게임 테스트. 한국정보과학회 학술발표논문집, 201-203.
이용정보 (Accessed)	동국대학교 223.39.201.*** 2022/01/17 11:49 (KST)

---

### 저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다. 그리고 DBpia에서 제공되는 저작물은 DBpia와 구독 계약을 체결한 기관소속 이용자 혹은 해당 저작물의 개별 구매자가 비영리적으로만 이용할 수 있습니다. 그러므로 이에 위반하여 DBpia에서 제공되는 저작물을 복제, 전송 등의 방법으로 무단 이용하는 경우 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

### Copyright Information

Copyright of all literary works provided by DBpia belongs to the copyright holder(s) and Nurimedia does not guarantee contents of the literary work or assume responsibility for the same. In addition, the literary works provided by DBpia may only be used by the users affiliated to the institutions which executed a subscription agreement with DBpia or the individual purchasers of the literary work(s) for non-commercial purposes. Therefore, any person who illegally uses the literary works provided by DBpia by means of reproduction or transmission shall assume civil and criminal responsibility according to applicable laws and regulations.

# GVG-AI 프레임워크 기반 에이전트를 활용한 게임 테스트

변혁<sup>o</sup> 최은만

동국대학교 컴퓨터공학

rexs333@naver.com, emchoi@dongguk.edu

## Game Testing Using Agents based on GVG-AI Framework

Hyuk Byun<sup>o</sup> Eun Man Choi

Computer Science and Engineering, Dongguk University

### 요 약

비디오 게임 산업에서 성공의 중요한 요소 중 하나인 게임 플레이 테스트는 비용이 많이 드는 작업이다. 비용을 줄이고 효과적으로 테스트하기 위하여 여러 가지 방법 중 AI를 사용하여 인간처럼 테스트하는 에이전트 기반의 프레임워크를 사용한다. 본 논문은 일반적인 에이전트 프레임워크인 GVG-AI을 통해 에이전트를 구성하여 VGDL(Video Game Description Language)로 작성된 게임 파일을 테스트하는 방법을 제안한다. 에이전트는 Sarsa 알고리즘을 사용하여 학습하고 VGDL로 Map 파일을 작성한 후 게임 파일에 의도적으로 버그인 부서지지 않는 벽을 생성한 뒤 탐지할 수 있는지 에이전트 기반의 테스트를 실험하였다.

### 1. 서 론

세계 게임 시장 규모는 2018년 전년 대비 7.1% 증가한 1783억 6800만 달러이고, 꾸준한 성장률을 보일 전망을 가지고 있다. 게임 시장의 규모가 커져가는 것에 따라 그 개발 과정에서도 많은 변화가 일어나고 있다. 프로젝트의 개발 시간과 비용의 50% 혹은 그 이상을 테스트에 지출하게 된다[1]. 이를 개선하기 위해 테스트의 자동화를 도입하는 데, 그 이유는 인건비의 부담이 크기 때문이며 테스트 직군의 자동화 가능성은 다른 직군들에 비해 높은 것으로 보고되었기 때문이다[2].

본 연구에서는 에이전트를 적용하여 게임을 테스트하기 위해 여러 방법을 찾아본 결과 에이전트의 적용 및 수정이 다른 프레임 워크(Unity 등)보다 쉽고 빠른 General Video Game AI (GVG-AI) 프레임워크를 이용하여 게임을 수정 및 테스트 하였습니다. 테스트 에이전트 선정은 강화학습에서 가장 대표적인 모델(Model) 없이 학습하는 Q-Learning 알고리즘을 사용하는 Sarsa(State Action Reward State Action)가 적용된 기존 에이전트를 이용하며, Video Game Description Language(VGDL)로 구성된 게임을 VGDL 코드를 수정하며, Sarsa 에이전트를 통해 게임 테스트를 진행하고 게임을 진행하며 버그를 탐지할 시 얻는 Reward로 버그의 발견 횟수를 표현한다. 본 연구에서는 Reward의 횟수가 가장 많이 나오는 방법을 찾고 그 방법을 통해 테스트를 진행한다. [3]

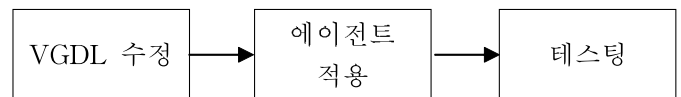


그림 1 테스트 과정

### 2. GVG-AI 프레임워크를 이용한 에이전트 설계

#### 2.1 General Video Game AI (GVGAI)

General Video Game Artificial Intelligence (GVGAI)는 비디오 게임의 플레이를 위한 컨트롤러를 만드는 문제를 탐구하기 위해 주어진 하나의 게임을 플레이할 수 있는 단일 에이전트 혹은 다양한 게임을 할 수 있는 에이전트를 이용하여 VGDL로 작성된 게임을 실행시키면서 에이전트를 테스트할 수 있는 프레임워크이다[4, 5]

#### 2.2 State Action Reward State Action (Sarsa)

본 논문에서 에이전트가 사용하는 Sarsa 알고리즘은 에피소드가 끝난 후 미래 보상의 합을 이용하여 Q-value를 계산하는 Monte Carlo tree search(MCTS)방법 대신 보상(Reward) 값과 다음 스텝의 Q-Value를 이용하여 Q-value를 최적화하는 Sarsa 알고리즘이다.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha (R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t))$$

$\alpha$  : 학습률(Learning rate)

$\gamma$  : 디스카운트(discount factor)

R : 보상(reward)

S : 상태(State)

A : 행동(Action)

Sarsa 알고리즘을 사용하여 에이전트가 게임을 플레이하며 Time-step마다 할 수 있는 행동 중 한 가지 Action을 선택하고 그러한 입력을 통해 보상(Reward)을 받고 State를 갱신하며 현재의 행동이 미래에 영향을 받으므로 다음 스텝의 미래 보상을 이용하여 계산하는 Q-Value를 최적화하는 방법을 사용하여 학습을 진행한다[6].

### 3. 에이전트를 이용한 테스트 실험 특징

#### 3.1 게임 전체 조건 및 특징

본 논문은 에이전트 환경이 잘 갖춰지고, 게임의 조작이 간단한 GVG-AI에서 지원하는 Alien 게임을 선택하였다. Alien 게임은 고정형 2D 슈팅 게임으로 플레이어(Avatar)가 좌우로 이동하며 위에서 나오는 적(Enemy)을 총알을 발사해 처치하는 게임이다.

고정형 2D 슈팅 게임에서 발생할 수 있는 중요한 버그로는 공격 처리 버그가 있다. 공격 처리 버그의 예시로는 플레이어(Avatar)가 정상적으로 상대를 공격하였으나 공격을 받는 대상이 해당 공격을 제대로 처리하지 못하는 경우를 말한다.

따라서 본 논문에서는 임의로 부서지지 않는 벽을 생성한 뒤 에이전트가 공격하며 해당 벽을 탐지할 수 있는지 확인하는 실험을 하였다.

기존에는 사람이 직접 게임을 플레이하며 직접 버그를 찾아야 하였다. 하지만 사람이 테스트하지 않고 에이전트만을 사용하여 테스트에 활용할 수 있는지 확인하기 위해 여러 방법으로 실험을 진행하였다. 실험은 게임의 버그를 알아내기 위한 것이므로 기존 게임 클리어를 목표로 하는 VGDL 코드를 버그를 찾기 위한 테스트를 위해 다음과 같은 전체 조건에 맞게 VGDL 코드를 수정하여 진행하였다.

- Avatar가 죽지 않음
- 버그로 설정된 블록은 사라지지 않음
- Alien은 버그로 설정된 블록을 통과함

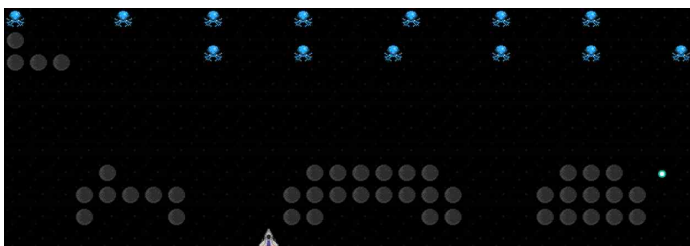


그림2 VGDL의 Alien 게임의 실행화면

#### 3.2 실험 별 설계 특징

본 논문에서는 해당 게임에 있어 어떠한 전체 조건이 버그를 탐지하는 데 있어 가장 효과가 좋은지를 판단하기 위해 여러 실험을 진행하였다.

실험 1은 버그로 설정된 블록을 한곳에 배치하여 에이전트가 움직이며 스스로 버그를 잘 찾을 수 있는지 실험을 진행함

실험 2는 버그로 설정된 블록을 서로 다른 세 곳에 배치하여 에이전트가 버그를 모두 탐지해낼 수 있음을 실험하였다.

실험 3부터 5의 베이스는 실험 2를 기초로 버그 탐지의 정확성을 위해 총알이 무엇도 닿지 않고 빗나가 화면 밖으로 나가는 경우 reward를 0과 -1로 나누어 실험을 진행하였다.

실험 3은 버그 블록에 reward를 각각 다르게 설정하여 중요한 버그일 경우 먼저 찾아내는 것을 유도하였다.

실험 4는 실험 3의 설정에서 에이전트의 활발한 활동을 유도하기 위해 기본 블록에도 reward를 주었다.

실험 5는 버그로 설정된 블록의 reward가 모두 같으며, 기본 블록에는 버그 블록보다 낮은 reward를 주어 버그 블록을 우선적으로 찾는 것을 유도하여 실험하였다.

### 4. 결과

#### 4.1 실험 결과

	레벨 3	레벨 4
실험 1	33.1	87.9
실험 2	52.7	102
실험 3_0	41.3	49.6
실험 3_-1	26.1	3.2
실험 4_0	43.88	74.4
실험 4_-1	42.9	68.5
실험 5_0	42.8	69.2
실험 5_-1	34.5	11.5

표 1 각 실험의 score 값의 평균

각 실험에서 얻은 score 값의 평균은 위와 같다. score는 실험 2에서 가장 큰 값이 나타났고, 실험 3\_-1에서 가장 낮은 값이 나타났다. 전체적으로 X\_0의 실험이 X\_-1의 실험보다 높은 score 값을 나타내는 것을 확인할 수 있었다. 또한, 다음 표 2에서 보이는 것처럼 레벨 4에서의 전체 탄환 발사 수가 많아 레벨 3보다 대체로 높은 점수를 획득하는 것을 확인할 수 있었다.

	레벨 3	레벨 4
--	------	------

실험 1	47.1	107
실험 2	57.5	111.5
실험 3_0	47.8	80.9
실험 3_-1	33.5	38.3
실험 4_0	45.7	95.5
실험 4_-1	46.9	94.8
실험 5_0	48.8	94.9
실험 5_-1	47.8	63

표 2 각 실험 별 에이전트의 발사 횟수의 평균

각 실험 별로 나타난 에이전트의 발사 횟수의 평균은 표 2와 같이 나타났다. 실험 3\_-1과 실험 5\_-1에서는 각각 실험 3\_0과 실험 5\_0에 비해 발사 횟수가 적은 것을 확인할 수 있었다. 하지만 실험 4\_-1은 실험 4\_0와 큰 차이가 없는 결과가 나오는 것을 확인할 수 있었다.

	레벨 3	레벨 4
실험 1	70.16 (0.08)	82.10 (0.03)
실험 2	91.44 (0.10)	91.46 (0.02)
실험 3_0	86.40 (0.03)	55.00 (0.27)
실험 3_-1	69.90 (0.26)	7.72 (0.05)
실험 4_0	95.84 (0.02)	76.69 (0.14)
실험 4_-1	91.40 (0.13)	71.07 (0.12)
실험 5_0	87.72 (0.04)	71.82 (0.11)
실험 5_-1	71.60 (0.21)	18.15 (0.18)

표 3 각 실험 정확도의 평균 (단위: %, 괄호 안은 표준편차)

각 실험에서 버그로 설정된 블록을 맞춘 정확도는 표 3과 같다. score가 높았던 실험에서 정확도도 마찬가지로 큰 값을 보여주었고, 표준편차는 실험 2가 가장 낮은 값을 보여주며 가장 집중되게 버그를 찾아냈다는 사실을 보여주었다. 또한, 레벨 3에서 X\_-1의 실험에서 X\_0의 실험보다 표준편차가 크게 나타나는 것을 확인할 수 있었다.

## 4.2 결과 특징

실험 3\_-1에서는 일반 블록에 리워드가 주어지지 않아 버그를 찾지 못하고 빗나가는 경우 reward가 감소하여 실험을 진행할수록 점점 발사 횟수가 줄어들며 낮은 정확도를 보여주었고, 반면에 실험 4\_-1에서는 일반 블록에 리워드가 주어져 버그를 찾지 못하여도 전체적으로 충분한 발사를 통해 버그를 찾아 버그의 위치를 파악하는 모습을 보여주었다. 또한, 실험 5에서는 각 버그의 리워드가 같아 버그 사이를 이동하며 탄환을 발사하며 빗나가는 탄환이 있어 reward가 감소하기 때문에 정확도가 실험 4에 비해 낮았다.

실험에서 계획할 때 X\_-1의 실험은 에이전트가 버그로 설정된 블록을 더 정확하게 찾아내기 위해 계획한

실험이었으나 실제로는 X\_0의 실험보다 X\_-1의 실험이 더 낮은 정확도를 보였다.

낮은 정확도를 보인 이유는 에이전트가 빗나가면 리워드를 잃는다는 확률이 버그 탐지를 하는 확률보다 높다고 판단하여 탄환을 발사하지 않았다.

## 5. 결론 및 향후 연구

본 논문에서는 기존 에이전트를 이용하여 게임 테스트를 진행하였다. 직접 에이전트를 통해 전제 조건을 만들고 테스트를 해본 결과 Alien 게임에서 실험 2 버그 블록이 3개가 들어있는 게임에서 에이전트 시작 위치에서 가장 먼저 찾은 버그 블록 한쪽으로 치우쳐 한 가지 블록만 때려 정확도는 제일 높지만, 나머지 두 가지 블록은 찾지 못해 해당 실험의 전제 조건을 테스트에 사용하기에는 적절치 않았다. 결과를 통해 기존에 존재하는 에이전트들은 주로 게임의 승률이나 높은 스코어를 위해서 만들어졌으나 이번 연구를 통해서 게임의 Description을 어떻게 설정하는지에 따라 테스트에서도 에이전트가 충분히 활용될 수 있음을 보여주었다. 따라서 규모가 큰 게임에서 에이전트를 사용하면 인력을 많이 줄일 수 있다고 보고 있다. 이번 연구에서는 Aliens 게임에서 한정되어 진행되었으나 향후 작업을 위해서는 다양한 게임에서 동일한 방법을 사용할 수 있는 방법이 만들어져야 할 것이며, 다양한 종류의 버그에 대해서도 테스트가 진행될 수 있어야 할 것이다.

## 6. 참고문헌

- [1] Glenford J. Myers, Corey Sandler, Tom Badgett, The Art of Software Testing, Wiley, 2011.
- [2] 이상규, 게임 생산자의 노동 연구(게임문화 융합연구1). 한국문화관광연구원, 2019.
- [3] VGDL and the GVGAI Framework, <https://gaigresearch.github.io/gvgaibook/PDF/chapters/ch02.pdf?raw=true>
- [4] Tom Schaul, "A Video Game Description Language for Model-based or Interactive Learning", IEEE Conference on Computational Intelligence in Games (CIG), pp. 1-8, 2013
- [5] The GVG-AI Competition, <http://www.gvgai.net/>
- [6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller, "Playing Atari with Deep Reinforcement Learning", Neural Information Processing Systems, 2013.