

GPT 를 이용한 게임의 동적 난이도 조절 연구

오진석¹, 성연식²

¹동국대학교 일반대학원 멀티미디어공학과 석사과정

²동국대학교 AI 소프트웨어융합학부 교수

ojsojs1113@dgu.ac.kr, sung@dongguk.edu

A Study on Dynamic Difficulty Adjustment in Games Using GPT

Jinseok Oh¹, Yunsick Sung²

¹Dept. of Multimedia Engineering, Graduate School,
Dongguk University University-Seoul, Korea

²Division of AI Software Convergence,
Dongguk University-Seoul, Korea

요약

게임의 난이도는 게임 디자인에서 중요한 요소로, 사용자의 몰입과 만족도를 크게 좌우한다. 전통적인 게임 난이도 조절은 일반적으로 기획자의 경험과 판단에 의해 이루어진다. 그러나 기존의 난이도 조절 방식은 다양한 사용자의 개별 실력 수준을 정밀하게 반영하는 데 한계가 있다. 본 연구에서는 GPT 를 활용하여 게임의 난이도를 동적으로 조절하는 방법을 제안한다. 사용자의 게임 플레이 정보를 수집하고 GPT 로 사용자의 게임 실력을 분석하고 평가한다. 평가를 바탕으로 게임의 난이도와 관련된 요소들을 조절한다. 이를 통해 GPT 가 게임 디자인 도구로서 가지는 잠재력을 탐구한다.

1. 서론

게임 난이도는 사용자 경험을 결정짓는 핵심 요소 중 하나로, 사용자의 몰입도와 만족감을 극대화하는데 중요한 역할을 한다. 난이도가 너무 쉬우면 게임이 지루해질 수 있고, 반대로 너무 어려우면 사용자는 좌절하여 게임을 중도에 포기할 가능성이 커진다 [1]. 이러한 이유로 많은 게임 개발자들은 사용자의 실력과 게임 난이도를 조화시키기 위해 다양한 방법을 시도한다.

전통적으로 게임의 난이도 조절은 기획자의 경험과 직관에 의존한다. 하지만 이러한 방식은 모든 사용자의 개별적인 실력 차이를 세밀하게 반영하기 어렵다는 한계가 있다 [2]. 다양한 사용자의 실력 수준을 일관되게 만족시키는 게임 플레이 환경을 제공하는데 제약이 될 수 있다.

본 논문은 대규모 언어 모델(Large Language Model, LLM)인 Generative Pre-trained Transformers(GPT) [3]를 활용하여 게임 난이도를 동적으로 조절하는 방안을 제안한다. GPT 는 인간의 언어를 이해하고 처리하는 능력이 뛰어나며, 다양한 입력 데이터를 분석하여 결

정을 내릴 수 있는 장점을 지닌다. 본 논문에서는 사용자의 게임 플레이 정보를 수집하고, 이를 GPT 에 입력하여 사용자의 현재 실력에 대한 평가를 출력한다. 이를 바탕으로 사용자의 실력에 맞는 난이도 요소 값을 생성하고, 난이도에 맞는 맵을 생성해 사용자에게 최적화된 게임 환경을 제공한다.

2. 관련 연구

2.1 동적 난이도 조절 (Dynamic Difficulty Adjustment)

동적 난이도 조절(DDA)은 사용자의 실력이나 행동에 맞춰 게임의 난이도를 조절하는 기술이다. DDA 는 고정된 난이도를 제공하는 기존 게임 디자인과 달리, 사용자에게 맞게 게임의 난이도를 조절함으로써 게임의 몰입감을 높인다. 하지만 이런 동적 난이도 조절 시스템을 구현하는 것은 복잡하며, 많은 시간과 자원이 요구된다 [4].

2.2 프롬프트 엔지니어링 (Prompt Engineering)

프롬프트 엔지니어링은 대규모 언어 모델, 특히 GPT 와 같은 모델을 활용하기 위해 입력 프롬프트를 설계하고 최적화하는 과정을 의미한다. 이 과정은 모

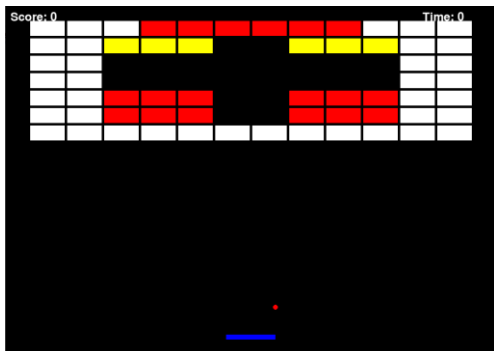
델의 응답을 특정 방향으로 유도하고, 결과를 도출하기 위해 필요하다. 프롬프트 엔지니어링은 질문을 던지는 것을 넘어, 모델이 특정 작업을 수행하도록 유도하는 기법과 전략을 포함한다.

3. 연구 방법

본 연구는 OpenAI 에서 제공하는 LLM 인 GPT[3]를 사용하여 게임의 난이도를 동적으로 조절하는 시스템을 설계한다. 게임은 Pygame 라이브러리[5]를 활용하여 고전 아케이드 게임인 Breakout[6]을 모방한 형태로 구현한다. 이를 통해 사용자의 게임 플레이 정보를 수집하고 GPT 에게 전달하여 난이도를 조절한다. 본 장에서는 구현한 게임과 수집된 게임 플레이 정보를 기반으로 GPT 를 활용한 동적 난이도 조절 방법에 대해서 설명한다.

3.1 Breakout 구현

Breakout 은 사용자가 패들을 조작하여 공을 벽들에 맞춰 제거하는 게임이다. 게임의 목표는 모든 벽들을 제거하는 것이다. 벽들의 종류는 세 가지 종류로 나뉜다. 하얀 벽들은 한 번의 타격으로 제거되며, 노란 벽들은 세 번의 타격, 빨간 벽들은 다섯 번의 타격으로 제거된다. 각 블록은 제거할 때의 점수가 다르며, 패들에 닿기 전에 다수의 벽들을 연속으로 제거하면 추가 점수를 획득한다. 게임 내의 모든 벽들을 제거하거나, 공이 바닥에 닿을 경우, 해당 라운드는 종료되고 성과 데이터를 기록한다. (그림 1)은 구현된 게임의 그림이다.



(그림 1) 구현된 게임 사진.

3.2. 게임 플레이 정보

게임 플레이 정보는 현재 정보와 과거 정보로 나누어진다. 현재 정보는 현재 게임의 성과 지표와 난이도 관련 요소를 포함한다. 과거 정보는 이전 게임의 성과 지표와 난이도 관련 요소, 그리고 GPT 의 평가를 포함한다. 성과 지표는 사용자의 게임 실력을 평가하기 위한 값으로 <표 1>과 같이 점수, 경과 시간, 그리고 제거한 벽돌 수 등이 포함된다. 사용자의 전반적인 성과를 측정하는 지표들로, GPT 가 사용자의 실력을 판단하는 기준이 된다. 난이도 관련 요소는

게임의 난이도를 조절하기 위한 요소들로 맵 난이도, 공의 최대 가능 속도, 그리고 패들의 너비 등이 포함된다. GPT 로 사용자에게 적합한 난이도로 조절하기 위해서, 난이도 관련 요소 값을 결정한다. GPT 의 평가는 현재의 성과 지표와 난이도 관련 요소를 기반으로 사용자를 평가한다. 현재 정보는 사용자가 플레이한 게임에서 성과 지표, 난이도 관련 요소, 그리고 GPT 의 평가는 메모리 모듈에 저장된다. 사용자의 실력 변화 추이를 파악하는 데 활용되며, 지속적인 난이도 조절을 통해 게임이 사용자에게 지속적으로 도전성을 제공할 수 있도록 한다. <표 1>은 성과 지표와 난이도 관련 요소의 종류를 나타낸 표이다.

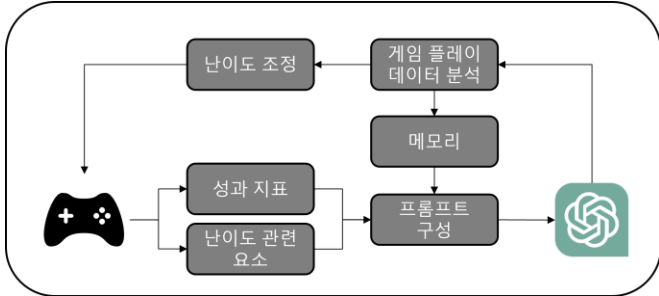
<표 1> 성과 지표와 난이도 관련 요소 종류

구분	항목
성과 지표	점수 (Score)
	제거된 벽돌 수 (Bricks Removed)
	연속 제거 수 (Consecutive Blocks Removed)
	경과 시간 (Elapsed Time)
	패들에 맞은 횟수 (Paddle Hit Count)
	공의 최종 위치 (Ball Position at End)
	평균 반사 각도 (Reflection Angle Average)
	게임 성공 여부 (Game Result)
난이도 관련 요소	맵 난이도 (Map Difficulty)
	공의 최대 속도 (Max Speed)
	공 속도 증가율 (Speed Increase Factor)
	패들 너비 (Paddle Width)
	패들 속도 (Paddle Speed)
	상단 간격 (Top Gap)
	측면 간격 (Side Gape)

3.3. 동적 난이도 조절 시스템 설계

(그림 2)는 동적 난이도 조절 시스템의 전체 구조를 나타낸 그림이다. 게임 라운드가 종료되면 수집된 게임 플레이 정보를 반영해서 GPT 의 입력 프롬프트를 가공한다. 입력 프롬프트는 게임 설명 프롬프트, 게임 정보 프롬프트, 평가 및 조절 지침 프롬프트, 그리고 응답 형식 예시 프롬프트로 구성된다. 게임 설명 프롬프트는 게임의 규칙과 목표를 설명하고 GPT 가 게임의 맥락을 이해할 수 있게 한다. 게임 정보 프롬프트는 현재 게임의 성과 지표와 난이도 관련 요소 값과 과거 정보로 구성된다. 과거 정보는 GPT 의 입력 길이의 제한으로 최근 네 게임의 과거 정보로 구성된다. 평가 및 조절 지침 프롬프트는 GPT 가 게임 플레이 정보를 해석하고, 난이도 관련 요소 값을 조절하는 데 필요한 가이드라인을 제공한다. 마지막으로, 응답 형식 예시 프롬프트는 GPT 의 일관된 응

답을 받기 위해 Few-Shot[7]으로 다수의 예시를 제공한다. GPT는 입력 프롬프트를 기반으로 사용자의 실력을 평가한다. 평가와 함께 사용자에게 적합한 난이도 관련 요소 값을 생성한다. 생성된 요소 값은 게임의 다음 라운드에 적용되어 게임의 난이도를 조절한다.



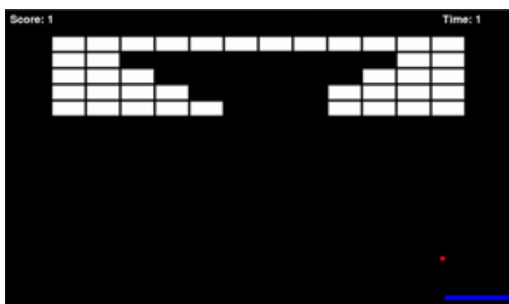
(그림 2) 동적 난이도 조절 시스템 구조도.

4. 실험결과

게임의 초기 난이도 관련 요소 값을 <표 2>와 같이 설정한다. 맵 난이도는 0부터 10까지의 값으로, 0에 가까울수록 쉽고 10에 가까울수록 어렵게 설정한다. 그 후, 다섯 번의 낮은 성과 지표와 라운드를 실패하고, 다섯 번의 높은 성과 지표와 게임을 성공했을 때와 다섯 번의 높은 성과 지표와 라운드를 성공하고, 다섯 번의 낮은 성과 지표와 게임을 실패했을 때의 난이도 관련 요소 값의 변화를 확인한다.

<표 2> 게임의 초기 난이도 관련 요소 값

난이도 관련 요소	값
맵 난이도	5
공의 최대 속도	10.0
공 속도 증가율	1.1
패들 너비	100.0
패들 속도	10.0
상단 간격	30.0
측면 간격	10.0



(그림 3) 다섯 번의 실패 시 생성된 맵.

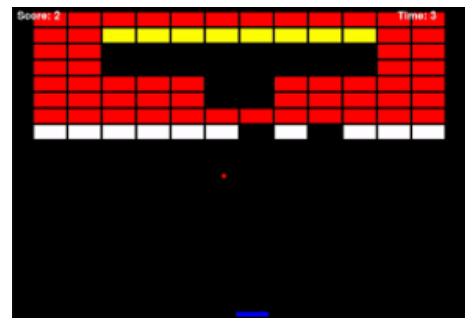


(그림 4) 다섯 번의 실패 후 다섯 번의 성공 시 생성된 맵.

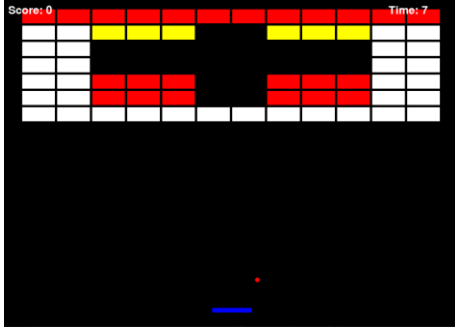
<표 3> 다섯 번의 실패 후 다섯 번의 성공 시 변화표

난이도 관련 요소	초기 값	5번 실패	5번 성공
맵 난이도	5	1	4
공의 최대 속도	10.0	5.0	7.0
공 속도 증가율	1.1	1.02	1.07
패들 너비	100.0	150.0	120.0
패들 속도	10.0	15.0	13.0
상단 간격	30.0	60.0	45.0
측면 간격	10.0	20.0	16.0

다섯 번의 낮은 성과 지표와 라운드를 실패했을 경우, 라운드 마다 현재 난이도는 사용자의 실력에 비해 어렵다고 판단하고, 지속적으로 난이도 관련 요소 값을 하향 조절하고 맵 난이도를 기준으로 (그림 3)과 같은 맵을 생성한다. 그 후 다섯 번의 높은 성과 지표와 라운드를 성공했을 경우, 1~2라운드에서는 현재의 난이도가 적절하거나, 난이도를 높이기에는 과거 게임 기록이 좋지 않았다고 판단하고, 현재의 난이도 요소 값을 유지하는 모습을 보인다. 그 후 지속적으로 성과 지표가 높게 나오에 따라 난이도 관련 요소 값을 상향 조절하고 (그림 4)와 같은 맵을 생성한다. <표 3>은 난이도 관련 요소 값이 초기 값에서 게임이 진행됨에 따라 변화되고 있음을 나타낸다.



(그림 5) 다섯 번의 성공 시 생성된 맵.



(그림 6) 다섯 번의 성공 후 다섯 번의 실패 시 생성된 맵.

<표 4> 다섯 번의 성공 후 다섯 번의 실패 시 변화표

난이도 관련 요소	초기 값	5번 성공	5번 실패
맵 난이도	5	10	6
공의 최대 속도	10.0	15.0	13.0
공 속도 증가율	1.1	1.5	1.3
패들 너비	100.0	70.0	80.0
패들 속도	10.0	7.0	9.0
상단 간격	30.0	15.0	25.0
측면 간격	10.0	5.0	8.0

위의 실험과 반대로, 다섯 번의 높은 성과 지표와 라운드를 성공했을 경우, 라운드 마다 현재 난이도는 사용자의 실력에 비해 쉽다고 판단하고, 지속적으로 난이도 관련 요소 값을 상향 조절하고 맵 난이도를 기준으로 (그림 5)와 같은 맵을 생성한다. 그 후 다섯 번의 낮은 성과 지표와 라운드를 성공했을 경우, 1~2 라운드에서는 현재의 난이도가 적절하거나, 난이도를 낮추기에는 과거 게임 기록이 좋았기에 실수 일수도 있다고 판단하고, 현재의 난이도 요소 값을 유지하는 모습을 보인다. 그 후 지속적으로 성과 지표가 낮게 나옴에 따라 난이도 관련 요소 값을 하향 조절하고 (그림 6)와 같은 맵을 생성한다. <표 4>은 난이도 관련 요소 값이 초기 값에서 게임이 진행됨에 따라 변화되고 있음을 나타낸다.

위 두 실험의 결과를 바탕으로, 동적 난이도 시스템이 사용자의 실력 변화에 따라 난이도 관련 요소를 조절하는 것을 확인할 수 있다. 초기에는 사용자의 성과 지표에 따라 난이도가 상향되거나 하향되었으며, 과거 게임 기록이 추가됨에 따라, 이를 참고하여 사용자의 실수나 우연한 성공을 고려해 난이도를 유지하는 모습을 확인할 수 있다.

5. 결론

본 연구에서는 GPT를 활용하여 게임의 난이도를 동적으로 조절하는 시스템을 제안했고, 이를 게임에 적용해 LLM이 게임 디자인 도구로서의 잠재력을 탐

구하였다. 실험을 통해 GPT가 사용자의 게임 플레이 정보를 기반으로 실력을 평가하고, 적절한 난이도를 제안하여 게임의 난이도를 조절할 수 있음을 확인하였다. 향후 연구에서는 복잡한 게임 장르로의 적용과 성과 지표와 난이도 관련 요소 간의 인과관계를 상세히 분석하고 보다 세밀한 난이도 조절이 가능하도록 동적 난이도 조절 시스템을 개선하는 연구를 수행할 것이다.

사사문구

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 인공지능융합혁신인재양성사업 연구 결과로 수행되었음 (IITP-2024-RS-2023-00254592).

참고문헌

- [1] Xue, S., Wu, M., Kolen, J., Aghdaie, N., & Zaman, K. A. "Dynamic Difficulty Adjustment for Maximized Engagement in Digital Games." In Proceedings of the 26th International Conference on World Wide Web Companion, 2017, pp. 465-471.
- [2] Hagelback, J., & Johansson, S. J. "Measuring Player Experience on Runtime Dynamic Difficulty Scaling in an RTS Game." In 2009 IEEE Symposium on Computational Intelligence and Games, 2009, pp. 46-52.
- [3] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., & Aleman, F. L. "GPT-4 Technical Report." arXiv preprint arXiv:2303.08774, 2023.
- [4] Ang, D., & Mitchell, A. "Comparing Effects of Dynamic Difficulty Adjustment Systems on Video Game Experience." In Proceedings of the Annual Symposium on Computer-Human Interaction in Play, 2017, pp. 317-327.
- [5] McGugan, W. "Beginning Game Development with Python and Pygame: From Novice to Professional." Apress, 2007.
- [6] Chen, A., Dewan, T., Trivedi, M., Jiang, D., Aditya, A., & Mohammed, S. "The Use of Reinforcement Learning in Gaming: The Breakout Game Case Study." Authorea Preprints, 2023.
- [7] Brown, T. B. "Language Models are Few-Shot Learners." arXiv preprint arXiv:2005.14165, 2020.