

유전알고리즘을 이용한 Match-3 게임 레벨 자동 생성

박인화, 오경수
송실대학교 미디어학부
{pih0304, oks}@soongsil.ac.kr

Automatic Generation of Match-3 Game Levels using Genetic Algorithm

InHwa Park, KyoungSu Oh
Dept. of Media, Soongsil University

요 약

본 논문은 유전 알고리즘을 통한 Match-3 게임의 레벨 자동 생성 방법을 제안한다. 적절한 난이도의 게임 레벨을 만들기 위하여 사람이 일일이 게임 레벨을 조절해야 한다면, 사람은 그것을 위한 많은 시간과 노력이 필요하다. 본 논문에서는 유전 알고리즘을 적용하여 Match-3 게임에서의 난이도에 맞는 블록조합을 생성한다. 각 게임 레벨의 블록조합이 진화되는 개체이다. 유전자인 정수로부터 블록조합 개체를 생성하고, 주어진 성공확률과 컴퓨터가 플레이했을 때의 성공확률이 가까워질수록 적합도가 높아지도록 설정하여 해당 블록조합을 진화시킨다. 이를 통해 게임 난이도에 맞는 적절한 블록조합들을 생성하는 데 성공하였으며, 실험한 결과 컴퓨터가 결정한 난이도가 실제 사람이 게임했을 때의 결과에 영향을 준다는 것을 확인하였다.

ABSTRACT

This paper proposes a automatic generation method of Match-3 game levels through genetic algorithm. It takes a lot of time and effort if persons have to control the level in the game. In this paper, the genetic algorithm is applied to create an appropriate block combination. We create block combination from integer DNA. Fitness is high if success probability played by computer is closer to given probability. Experiments have shown that computer-determined levels of difficulty have a significant impact on the results of game played persons.

Keywords : Genetic Algorithm(유전 알고리즘), Game Level Generation(게임 레벨 생성), Automatic Game Difficulty(게임 난이도 자동 생성)

Received: May. 13. 2019 Revised: Jun. 7. 2019
Accepted: Jun. 12. 2019
Corresponding Author: In-Hwa Park(Soongsil University)
E-mail: pih0304@soongsil.ac.kr

ISSN: 1598-4540 / eISSN: 2287-8211

© The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서 론

Match-3 게임이란 일정한 게임 레벨에 해당하는 블록조합에서 특정 블록을 상하좌우로 움직여 같은 종류의 블록을 가로, 세로로 세 개 이상 한 줄로 만들어 없애는 형태의 게임을 말한다[1].

적당한 난이도의 게임 레벨을 만들기 위해서 해당 레벨에 맞는 블록조합을 사람이 일일이 조절하여 생성한다면 사람은 많은 시간과 노력을 필요로 한다. 특히 애니팡과 같은 Match-3 게임에서 난이도를 제어하기 위하여, 블록을 하나하나 생성해야 한다면 더욱 그럴 것이다.

본 논문에서는 유전알고리즘을 적용하여 그리드와 블록조합을 기반으로 하는 Match-3 게임의 레벨을 자동으로 생성한다.

게임을 비롯한 콘텐츠의 자동 생성에 관한 많은 연구들이 있었다[2,3,4,5,12,13,14,15]. 게임 분야에서는 게임 인공지능의 자동 생성[6], 캐릭터나 몬스터의 자동 생성[2,4], 지형의 자동 생성[5], 전투 게임의 전략 자동 생성[3] 등의 연구들이 있었다. 특히 게임 레벨을 자동으로 생성하는 것에 관한 많은 연구들이 있었는데[7,8,9,10], 전투 게임의 레벨 자동 생성[7], 비디오 게임의 레벨 자동 생성[8,10] 등이 있었다. 2-D 플랫폼의 레벨 자동 설계[9], 진화를 통한 새로운 게임 자동 생성[11]에 관한 연구도 있었다. 본 연구에서는 퍼즐 게임인 Match-3 게임에 레벨을 적용할 수 있도록 수정하여 난이도에 맞는 레벨을 자동 생성하는 방법을 제안하였다.

본 논문에서는 Match-3 게임에 레벨 개념을 넣어, 블록조합이 랜덤으로 생성되는 것이 아닌 게임 레벨 별로 고정되는 방식으로 수정했다. 대부분의 Match-3 게임은 블록조합이 랜덤으로 생성되며 난이도는 그 외의 아이템이나 추가 장치를 통하여 조절한다. 난이도를 조절하기 위한 블록조합의 생성 과정은 컴퓨터가 자동으로 처리하고, 그 후에 사람이 심미성 등의 다른 기준들과 같이 고려하여 어느 블록조합을 사용할지 판단한다면, 시간과 노

력이 훨씬 단축될 것이다.

유전알고리즘(Genetic Algorithm)이란 자연계에 존재하는 생물의 진화 과정을 모방하여 문제를 해결하는 방법이다. 문제에 대한 해를 정해진 형태의 자료구조로 표현하고 그 해가 얼마나 적합한지를 적합도 함수(Fitness Function)를 통해 계산하여, 여러 해들을 진화시켜 점차적으로 적합도가 더 좋은 해를 만들어 낸다.

본 논문에서는 유전알고리즘을 이용한 적절한 난이도를 가지는 게임 레벨을 생성하는 방법을 제안한다. 게임에서 생성되는 블록조합을 유전자로 정하고 이것을 진화시킨다. 게임 결과의 점수를 바탕으로 적합도 함수를 계산하여 높은 점수가 나올 수 있는 더 좋은 해를 진화를 통해 점차적으로 생성한다.

주어진 블록조합을 가지고 컴퓨터가 여러 번 플레이하고 이 점수의 평균과 표준 편차를 구한다. 게임의 합격 점수를 고정하고 정규분포를 가정하면 그 점수를 넘을 확률을 계산할 수 있다. 난이도에 따라 이 확률을 조절하여 적합한 블록조합을 생성한다.

본 논문의 기여 내용을 정리하면 다음과 같다.

Match-3 게임의 블록조합 생성을 위한 유전자를 제안하였다.

주어진 성공확률에 근접하느냐에 기반을 둔 블록조합의 적합도 계산 방법을 제안하였다.

유전알고리즘을 이용한 난이도에 적합한 Match-3 게임에서의 블록조합의 생성 방법을 제안하였다.

본 논문의 구성은 다음과 같다. 2장에서는 Match-3 게임의 방법과 유전알고리즘에 대해 설명하며, 유전알고리즘을 적용한 Match-3 게임에서의 블록조합의 생성 방법에 대해 설명한다. 3장에서는 2장에서의 방법에 따라 구현한 Match-3 게임을 설문 조사와 실험을 통해 검증하며, 4장에서는 이 검증 결과에 대해 분석하고 결론을 논한다.

2. 유전알고리즘을 이용한

Match-3 게임의 레벨 자동 생성

본 논문에서는 유전알고리즘을 적용하여 난이도에 따른 블록조합을 생성하는 Match-3 게임 레벨의 자동 생성 방법을 제안한다. 본 논문의 알고리즘의 목표는 적당한 난이도를 가지는 블록조합을 생성하는 것이다.

2.1 게임 방법 설명

[Fig. 1]은 본 게임의 실행 화면이다. 각 단계마다 가로 5, 세로 10 크기의 블록조합이 생성되고 각 블록은 특정 색깔을 가진다. 블록을 상하좌우로 움직여 가로, 세로로 같은 색을 3개 이상 한 줄로 만든다. 매칭이 성공하면 블록이 없어지고 위에 있던 블록이 내려오며, 블록이 내려와서 새로운 매칭이 생기면 블록이 없어지고 위의 블록들이 내려오는 것이 반복된다. 게임이 진행됨에 따라 위의 공간은 빈 공간이 된다.

사용자의 한 번의 조작으로 인해 없어진 블록의 개수를 k 라 하면 사용자는 1.26^k 의 점수를 얻는다. 이렇게 하면 한 번에 블록을 세 개를 더 없앨 때마다 점수가 2배가량 된다. 즉 블록을 하나 없앨 때마다 점수는 1.26배씩 증가되므로 사용자는 한꺼번에 많은 블록을 없애야 고득점을 얻을 수 있다. 그림에서 A의 Combo는 바로 전 매칭 성공에 의한 점수를 나타내는 것이고, Score는 점수의 총합을 나타낸다.

단계별로 난이도는 처음에는 쉽다가 점차적으로 어려워지도록 구성하였으며, B에 표시되는 기준 점수 이상을 받을 경우 바로 다음 단계로 이동하여 플레이할 수 있다.



[Fig. 1] Scene about Playing Match-3 Game

2.2 유전알고리즘을 이용한 블록조합 생성

2.2.1 유전알고리즘 개요

유전알고리즘(Genetic Algorithm)이란 자연계에 존재하는 생물이 진화하여 주어진 환경에 적합한 다음 세대를 만드는 과정을 모방하여 문제를 해결하는 최적화 방법이다. 주어진 문제에 대한 해를 정해진 형태의 자료구조로 표현하고 이것을 유전자로 정의하며, 교배와 돌연변이 생성을 통해 다음 세대를 만들어간다. 또한 이것이 얼마나 적합한지를 적합도 함수(Fitness Function)를 통해 계산하여 점차적으로 더 좋은 해를 만들어 내면 최적화된 해를 찾을 수 있다.

유전자란 해결할 문제에 대한 해를 정해진 자료구조를 통해 표현한 것으로서, 설정한 유전자에 대한 적합도(Fitness)를 계산하고 교배와 돌연변이 생성을 통해 다음 세대를 만든다. 생성된 자손들도 적합도 함수를 통해 계산하며 적합도가 원하는 값이라면 채택하고 아닐 경우 교배와 돌연변이 생성의 과정을 계속 반복하며 다음 세대를 만들어간다.

2.2.2 유전자 자료구조 및 유전자로부터 블록조합 생성 방법

게임 레벨에 따른 블록 조합을 구성하는 데에 유전알고리즘을 적용하기 위하여 랜덤 값을 생성하기 위한 정수 1개로 이루어진 seed를 유전자로 설정하고, 이것을 적합도 함수로 계산하여 최적화된 해가 나올 때까지 진화시키면서 다음 세대를 만들었다.

랜덤함수는 랜덤한 숫자를 만들어주는 함수이다. 랜덤함수를 위한 초기 값을 일정한 값으로 지정하면 이후에 랜덤함수에 의해 생성되는 랜덤 숫자들의 값이 같은 순서대로 출력된다. 랜덤함수의 seed를 지정한 상태에서 같은 순서대로 블록 색상을 생성하면, 항상 같은 블록 색상 조합을 가지게 되어 seed마다 고유의 블록조합을 생성할 수 있다.

유전알고리즘에서의 유전자는 진화시키려고 하는 대상을 말한다. 본 논문에서는 유전자를 정수 1개의 seed로 지정하였으며, 이 seed로 랜덤함수를 호출하면 각 seed마다 숫자의 순서가 일정하게 정해지는데 이것으로 일정한 블록조합을 생성할 수 있다.

```
set seed for RandomNumber Generation
for each element b in blockList
    b,color= RandomNumber%NumberOfColor
```

유전자(seed)로부터 블록조합을 생성하는 과정을 간단한 의사코드로 나타내면 위와 같다. 여기서 blockList란 블록조합을 저장하는 자료구조이며 2차원 배열로서 각각의 블록들의 색상을 저장한다. NumberOfColor란 각 블록 요소들이 가질 수 있는 색깔의 총 개수를 말하는 것으로 만약 이 값이 n이라면 하나의 블록 요소는 0부터 n-1까지의 인덱스 값을 가진다. 랜덤함수를 위한 초기 값을 seed로 지정하고, blockList의 각각의 블록 요소 b에 대해 RandomNumber를 생성한 후, 이것을 NumberOfColor로 나눈 나머지 값에 해당하는 인덱스를 가진 색깔을 해당 블록에 지정한다.

2.3 블록조합의 적합도

본 논문에서 다루는 게임은 사용자가 각 단계별 기준 점수를 넘으면 성공하는 게임이다. 처음에는 게임 숙련도가 낮으므로 난이도가 쉬워야 하고 점차 어려워져야 한다. 이를 위해서 난이도가 주어지면 그 난이도에 맞는 블록조합을 생성할 수 있어야 한다. 목표성공확률을 설정한 후, 블록조합을 가지고 게임을 했을 때 성공할 확률이 목표성공확률과 유사하면 유사할수록 적합도 값이 높아지도록 하였다. 주어진 블록 조합을 가지고 컴퓨터가 게임을 여러 번 플레이하여 나온 점수의 평균과 표준편차를 가지고 기준 점수를 넘길 확률, 즉 성공확률을 계산하였고 이를 바탕으로 적합도 값을 계산하였다.

다음은 주어진 블록조합을 가지고 컴퓨터가 한 번 게임을 플레이하는 것에 대한 의사코드이다.

```
for each random move {
    move random block to random direction
    k= number of removed block
    score = score + 1.5^k
}
```

하나의 블록 조합에서 랜덤한 블록 위치를 선택하고, 상하좌우 방향 중 하나를 랜덤하게 선택하여 블록을 이동하면 점수가 추가되며 이 과정을 100회 반복하여 합산한 점수를 계산하였다.

이것을 여러 번 반복하여 해당 블록조합으로 게임한 점수를 여러 개 도출한다. 도출된 점수들의 평균(m)과 표준편차(σ)를 구하고 정규분포를 가정하여 기준 점수(X)보다 높게 나올 확률을 구할 수 있는데, 이것이 바로 현재 블록조합의 성공 확률이다.

평균이 0이고 표준편차가 1인 정규분포 $N(0,1^2)$ 을 표준정규분포라 한다. 확률변수 X 가 정규분포 $N(m,\sigma^2)$ 를 따를 때, 새로운 확률변수 Z 를 정의하고 그것을 $Z = \frac{X-m}{\sigma}$ 로 변환시켜주면 새로운 확률변수 Z 의 확률분포는 표준정규분포를 따르게

된다. 기준 점수를 X 로 주면 $Z = \frac{X - m}{\sigma}$ 를 표준 정규분포 표에 적용하여 Z 의 값이 구간 $[0, z]$ 에 속할 확률 $P(0 \leq Z \leq z)$ 를 구할 수 있다. 점수를 확률변수 Z 로, 기준 점수를 z 라고 정의하면 점수가 기준 점수보다 높은 구간에 속할 확률 P 를 구할 수 있고 이렇게 구한 확률 P 가 바로 현재 성공 확률이다.

$$fitness = 1 - |목표 성공 확률 - 현재 성공 확률|$$

fitness 값은 0에서 1사이의 값을 가질 수 있으며 이 값이 1에 가까울수록 적합도가 높은 것이다. 목표 성공 확률과 현재 성공 확률의 차이가 적을수록 이 값이 1에 가까워져 원하는 성공 확률에 맞는 가장 적합한 해를 도출할 수 있다.

3. 구현 및 실험

게임 제작을 위한 구현 환경은 다음과 같다. Windows 버전은 Windows 10 Home, 프로세서는 Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz 2.20GHz, 메모리(RAM)는 8.00GB의 64비트 운영 체제, x64 기반 프로세서로 이루어진 환경에서 구현하였다. 게임 제작을 위하여 사용한 유니티 프로그램의 버전은 Unity 5.6.4f1 64bit, 비주얼 스튜디오는 Visual Studio 2017을 사용하여 구현하였다.

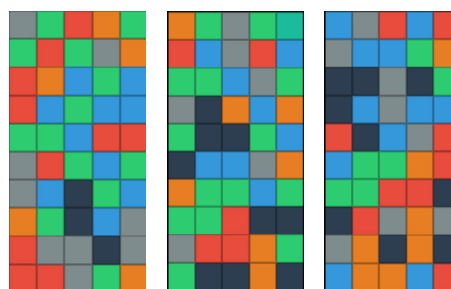
유전알고리즘을 사용하여 컴퓨터에게 기준 점수를 넘길 확률을 주고 그에 맞는 유전자 배열과 평균, 표준편차를 구하도록 하였다. 다음의 표 [Table 1]에서 Probability는 각 블록 조합으로 게임을 할 경우 획득한 점수가 기준 점수를 넘길 확률을 나타낸다. 본 논문에서는 기준 점수를 15점으로 설정하였다. Gene은 앞에서 설정한 Probability를 컴퓨터에게 주고 게임을 시켜 구한 유전자 배열을 나타낸다. 이 유전자배열에 해당하는 블록조

합을 각각 컴퓨터가 여러 번 플레이하여 구한 평균과 표준 편차가 바로 Average와 Standard deviation이다. 이렇게 구한 평균과 표준편차를 정규분포 표에 대입하여 확률을 구하면 Probability와 같은 값이 나온다. Stage는 게임의 단계를 나타내며 1이 가장 쉬운 단계이고 3으로 갈수록 난이도가 높아진다.

[Table 1] Block Combinations by Genetic Algorithm

Stage	Gene	Average	Standard deviation	Probability
1	804118649	17.2	4.31	70%
2	67428686	16.8	5.59	63%
3	-977507440	16.0	5.95	57%

이로부터 생성된 게임의 각 단계별 블록조합은 다음과 같다. 1단계는 [Fig. 2], 2단계는 [Fig. 3], 3단계는 [Fig. 4]와 같은 블록조합을 가진다.



[Fig. 2] Block Combination of Stage 1
 [Fig. 3] Block Combination of Stage 2
 [Fig. 4] Block Combination of Stage 3

유전알고리즘을 통해 생성한 각 단계별 블록조합이 적절한 난이도로 이루어졌는지 알아보기 위하여 대한민국에 거주하는 20대 남녀 20명에게 실험을 하였다. 실험은 대면 형식으로 이루어졌으며 각 단계별로 승패와 상관없이 5번씩 게임하도록 하였다. 10명은 난이도가 낮은 1단계부터 3단계의 순으로 게임하도록 하였고, 10명은 난이도가 높은 3단

계부터 1단계의 순으로 게임하도록 하여 총 20명의 실험자가 각 단계별로 게임한 점수를 기록하였다. 다음의 [Table 2]는 사람들에게 실험한 결과로 구한 점수들의 평균과 표준편차, 그리고 성공 확률을 보여준다. Gene은 각 단계에 해당하는 유전자의 배열이며, Average와 Standard deviation은 실험자들이 각 단계별로 일정한 횟수만큼 게임을 플레이한 결과로 도출된 점수들의 평균과 표준편차를 구한 것이다. 이렇게 구한 평균과 표준편차를 이용하여 정규분포를 가정하면 Probability를 도출할 수 있으며 이것은 게임의 점수가 기준점수를 넘길 확률과 같다.

[Table 2] Success Probability by Played Persons

Stage	Gene	Average	Standard deviation	Probability
1	804118649	13.8	2.65	34%
2	67428686	13.7	2.00	28%
3	-977507440	12.3	3.42	21%

[Table 1]과 [Table 2]를 비교해 보면 유전알고리즘을 이용하여 생성한 블록조합을 컴퓨터로 플레이하였을 때보다 사람이 플레이하였을 때의 결과가 더 낮은 하지만 대체적으로 비슷한 양상을 띠고 있다는 것을 확인할 수 있다. 즉 컴퓨터의 성공 확률이 높은 게임일수록 사람의 성공확률도 높았다. 두 경우 모두 난이도가 가장 낮은 1단계에서 점수의 평균과 성공 확률 모두 가장 높으며, 난이도가 가장 높은 3단계에서 점수의 평균과 성공 확률 모두 가장 낮게 나타났다. 단, 컴퓨터가 플레이하였을 경우 기준 점수를 넘길 확률, 즉 성공 확률은 반올림하여 60-70% 정도인 반면, 사람이 하였을 때의 성공 확률은 반올림하여 20-30%이다. 또한 점수의 평균은 컴퓨터의 경우 약 16에서 17 사이인 반면, 사람의 경우는 약 12에서 13 사이를 나타낸다. 컴퓨터는 랜덤한 움직임을 100회 시도하는 과정을 통해 점수를 합산하였지만 사람의 경우에는 훨씬 적은 횟수를 시도한 것이 원인으로 판단된다.

[Table 3] Comparison of the Average Score According to the Order of Game Stage

	1	2	3
1 -> 3	13.12	14.25	12.55
3 -> 1	14.67	13.33	11.96

위의 [Table 3]은 실험자들을 반으로 나누어 10명에게는 난이도가 가장 낮은 1단계부터 난이도가 높아지는 순으로, 나머지 10명은 난이도가 가장 높은 3단계부터 난이도가 낮아지는 순으로 게임을 하게 한 것에 대한 점수의 평균을 기록한 것이다. 난이도가 높아질수록 높은 점수를 받기가 어려워지기 때문에, 두 경우 모두에서 실험자들의 1단계의 점수가 3단계의 점수보다 높게 나온 것을 확인할 수 있다. 또한 1단계부터 게임을 하였을 때보다, 3단계부터 게임을 하였을 때 1단계의 점수의 평균이 더 높다는 것을 알 수 있다. 또한 3단계도 마찬가지로 3단계부터 게임을 하였을 때보다, 1단계부터 게임을 하였을 때 3단계의 점수의 평균이 더 높은 것으로 나타났다. 이를 통해 사람은 게임을 반복하여 플레이할수록 게임 방법에 익숙해지므로 같은 단계의 블록조합으로 게임을 하더라도 더 높은 점수를 받을 수 있다는 것을 알 수 있다.

위의 실험을 통하여 유전알고리즘을 사용하여 컴퓨터로 만든 블록조합의 레벨에 따라 결정된 난이도가 실제 사람이 플레이할 때의 난이도와 매우 관계가 깊으며, 게임의 점수에 영향을 준다는 사실을 확인할 수 있다.

4. 결 론

게임 레벨 개념이 존재하는 Match-3 게임을 구현하기 위하여 유전알고리즘을 사용하여 컴퓨터로 하여금 성공 확률을 주고 그 값에 해당하는 적절한 평균과 표준편차를 가진 유전자 배열을 구하도록 하였다. 이것을 통하여 난이도에 맞는 적절한 블록조합을 생성하였으며, 실험한 결과 컴퓨터가 결정한 게임의 난이도가 실제 사람이 플레이할 때

의 게임 결과에 큰 영향을 주며 난이도의 차이에 따라 게임의 점수가 달라진다는 것을 확인할 수 있었다. 이로써 유전알고리즘을 사용하여 Match-3 게임의 난이도를 쉽게 조절할 수 있다는 것을 확인하였다. 본 논문에서 우리는 Match-3 게임에서 적절한 난이도를 가진 블록조합을 생성하기 위한 유전자를 제안하였으며, 주어진 성공확률에 근접하느냐에 기반을 둔 블록조합의 적합도 계산 방법을 제안하였다. 또한 유전알고리즘을 이용하여 적절한 난이도를 가진 Match-3 게임의 블록조합 생성 방법을 제안하였다.

ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2018R1D1A1B07050099)

REFERENCES

- [1] Krivicich, Alexander Higgins, et al, "Match three gaming system and method", U.S. Patent Application No. 13/494,941, 2013.
- [2] Karl Sims, "Evolving virtual creatures", SIGGRAPH '94 Proceedings of the 21st annual conference on Computer graphics and interactive techniques, pp15-22, 1994.
- [3] T.E. Revello, R. McCartney, "Generating war game strategies using a genetic algorithm", Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), 2002.
- [4] Sang-Wook Park, Won-Hyung Lee, "Genetic Algorithm for Game Monster Generation", Proceedings of the Korea Contents Association Conference, Vol. 4, No. 2, pp82006.
- [5] Teong Joo Ong, Ryan Saunders, John Keyser, John J. Leggett, "Terrain generation using genetic algorithms", Proceeding GECCO '05 Proceedings of the 7th annual conference on Genetic and evolutionary computation, pp1463-1470, 2005.
- [6] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, Dennis Hassabis, "Mastering the game of Go with deep neural networks and tree search", Nature Vol. 529, pp484 - 489, 2016.
- [7] Shin-Jin Kang, Seung-Ho Shin, Sung-Hyun Cho, "A Game Level Design Technique Using the eGenetic Algorithms", Journal of The Korea Computer Graphics Society, Vol. 15, No. 4, pp13-21, 2009.
- [8] Sorenson N., Pasquier P., "Towards a Generic Framework for Automated Video Game Level Creation", Applications of Evolutionary Computation. EvoApplications 2010. Lecture Notes in Computer Science, Vol. 6024, Springer, Berlin, Heidelberg, 2010.
- [9] Gillian Smith, Jim Whitehead, Michael Mateas, Mike Treanor, Jameka March, Mee Cha, "Launchpad: A Rhythm-Based Level Generator for 2-D Platformers", IEEE Transactions on Computational Intelligence and AI in Games, Vol. 3, No. 1, pp1-16, 2011.
- [10] Nathan Sorenson, Philippe Pasquier, Steve DiPaola, "A Generic Approach to Challenge Modeling for the Procedural Creation of Video Game Levels", IEEE Transactions on Computational Intelligence and AI in Games, Vol. 3, No. 3, pp229-244, 2011.
- [11] Cameron Browne, Frederic Maire, "Evolutionary Game Design", IEEE Transactions on Computational Intelligence and AI in Games, Vol. 2, No. 1, pp1-16, 2010.
- [12] Edward J. Powley, Simon Colton, Swen Gaudl, Rob Saunders, Mark J. Nelson, "Semi-automated level design via auto-playtesting for handheld casual game creation", IEEE Conference on Computational Intelligence and Games, 2016.
- [13] Togelius J., Yannakakis G.N., "Emotion-Driven Level Generation", Emotion

- in Games. Socio-Affective Computing, vol 4, 2016.
- [14] Adam Summerville, Michael Mateas, “Super Mario as a String: Platformer Level Generation Via LSTMs”, Neural and Evolutionary Computing (cs.NE), vol 2, 2016.
- [15] Unnat Jain, Svetlana Lazebnik, Alexander G. Schwing, “Two Can Play This Game: Visual Dialog With Discriminative Question Generation and Answering”, The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp5754-5763, 2018.



박 인 화 (Park, In Hwa)

약 력 : 2015-현재 숭실대학교 미디어학부 수료

관심분야 : 게임 레벨 자동 생성, 컴퓨터 그래픽스



오 경 수 (Oh, Kyoung Su)

약 력 : 2001 서울대학교 전기 컴퓨터 공학부 박사

2001-2002 ㈜조이먼트 개발팀장

2003-현재 숭실대학교 글로벌미디어학부 교수

관심분야 : 실시간 컴퓨터 그래픽스, 기능성 게임
