

# 유전자 알고리즘을 사용한 게임 레벨 디자인 기법

강신진\* 신승호+ 조성현\*\*

\*홍익대학교 게임학부, +고려대학교 정보통신대학 컴퓨터학과  
directx@korea.ac.kr winstorm@korea.ac.kr scho@hongik.ac.kr

## A Game Level Design Technique Using the Genetic Algorithms

Shin Jin Kang<sup>o</sup>, Seung Ho Shin, Sung Hyun Cho

\* School of Games, Hongik University

+Department of Computer Science and Engineering, Korea University

### 요 약

레벨 디자인은 게임 개발에 있어 가장 중요한 부분 중 하나이지만, 많은 게임 내 구성 요소가 복잡적으로 영향을 미치어 최적의 조합을 찾기가 어려운 문제이다. 본 논문에서는 유전자 알고리즘을 사용하여 범용적인 게임 레벨 생성을 위한 방법을 새로이 제안한다. 제안된 시스템을 통해 게임 디자이너는 단순화된 목표 수치 입력만으로도 다양한 게임 구성 요소로 이루어진 레벨을 근사화시켜 생성할 수 있게 된다. 이는 게임 레벨 디자인 업무 프로세스 상에서 원하는 최적의 레벨을 구성하는데 걸리는 시행 착오 시간을 단축시키는데 도움이 될 수 있다.

### Abstract

Game level design is one of the important parts in the commercial game development. Because of its complexity in combining game components, game design work could be classified into a non-linear problem. In this paper, we propose a new automated game level design system by using genetic algorithms. With our system, a game designer easily generates an optimized game level by designating the key parameters in the initial stage of game design. Our system can be useful in reducing the trial-errors in the initial game level design process.

키워드: 유전자 알고리즘, 진화형 연산, 게임 디자인, 게임 레벨 디자인

Keywords: Genetic Algorithm, Evolutionary Computation, Game Design, Game Level Design

## 1. 서론

게임 레벨 (Game Level)이란 게임 내 스테이지, 미션 등을 통칭하는 것으로, 플레이어가 본인의 캐릭터를 컨트롤하여 플레이할 수 있는 공간 및 속해 있는 콘텐츠 전반을 말한다.

교신저자: 조성현, \* 이 논문은 2009학년도 홍익대학교 학술연구진흥비에 의하여 지원되었습니다. \*\* 이 논문은 2007학년도 홍익대학교 학술연구진흥비에 의하여 지원되었습니다.

게임 레벨 디자인 (Game Level Design)이란 다양한 게임의 시스템적 특징에 맞는 재미있는 게임 레벨을 설계, 구성하는 업무를 지칭한다.

여러 게임 장르 중 레벨 디자인의 중요성이 크게 부각되는 분야로는 FPS (First Person Shooting) 게임, 슈팅 (Shooting) 게임, 롤 플레이 (Role Playing) 게임, 어드벤처 (Adventure) 게임, MMORPG (Massively Multiplayer Online Role-Playing Game)이 존재한다. 이들 장르에서 대부분의 게임 플레이는 게임 내 레벨에서 이루어지게 되므로 만약 레벨 디자인 컨셉, 구성, 밸런싱, 몰입감 등에 있어 부족한 점이 있다면 플레이어로서 하여금 게임에 대한 흥미를 잃게 만들어 게임을 중단하게 되는 요인이 된다.

게임 레벨 디자인은 해당 게임이 제공하는 재미에 큰 영향을 미치지만 그 중요도만큼 가장 난이도가 높은 개발 업무 분야에 속한다. 그 이유는 바로 게임 레벨 내에는 게임 내 모든 시스템과 콘텐츠가 조합되어 적용되는데, 이때 그 복잡한 최종 결과물에 대한 재미를 사전에 정확히 예측하는 것이 게임 디자인 설계 단계에서는 사실상 불가능하기 때문이다. 사전에 게임의 최종 결과물의 예측이 불가능하므로 결국 레벨 디자인 작업 후 이를 실제로 확인하기 위한 강도 높은 플레이 밸런싱 (Balancing) 작업이 필수적으로 이어지게 된다. 그러나 이러한 레벨 디자인 밸런싱 작업은 대단히 주관적인 영역에 속할 뿐 아니라 많은 시간이 소요되어 게임 개발팀 업무에 큰 부담으로 작용하게 된다.

본 논문에서는 유전자 알고리즘을 사용하여, 현 레벨 디자인 작업 프로세스 상에서의 레벨 디자인 구성요소의 조합에 대한 수치적 불확실성을 해소하고, 시간적 소모를 줄일 수 있는 자동화된 레벨 디자인 모델을 제안한다. 이를 통해 레벨 디자인 시에 발생할 수 있는 다수의 시행 착오를 게임 레벨 디자인 단계에서 경감시키고자 한다.

## 2. 관련연구

게임 레벨 내에서는 플레이어와 인터랙션을 발생시키는 다양한 오브젝트들이 등장하며 이러한 오브젝트들은 플레이어가 컨트롤하는 PC (Player Character)와 대비되어 NPC (Non Player Character)라고 통칭된다. 이러한 NPC 들은 다양한 AI (Artificial Intelligence) 기법들에 의해 그 역할과 작동 방식이 결정되게 된다. NPC 들에게 적용되는 대표적인 AI 기법으로는 FSM (Finite State Machine)을 사용한 NPC 행동 제어와 A\* 알고리즘을 사용한 길찾기 알고리즘 부분을 뽑을 수 있다 [1][2][3]. 광범위한 AI 연구 분야에 비하여 게임 내 NPC 들에 적용되는 AI 기법은 지극히 제한되어 있는데, 이는

플레이어에게 재미를 전달하기 위해 최상의 AI 기법을 선택하기 보다는 설정된 환경에서 게임 디자이너가 원하는 결과 행동을 도출할 수 있는 시스템이 구현 안정성이 높고 재미를 전달하는데 더욱 효율적이기 때문이다 [3].

그러나 최근에 이러한 한정된 AI 기법에서 벗어나 NPC 에 좀 더 개선된 AI 를 적용하려는 시도가 지속적으로 이루어지고 있다. 관련된 기존 연구로는, 유전자 알고리즘 (Genetic Algorithm)을 사용하여 NPC 들의 외형 디자인을 진화시키거나 [4], 회피-추적 문제에서 게임 에이전트 학습을 시키거나 [5], 인공 지능 캐릭터의 성능을 지속적으로 발전시켰던 사례가 있다 [6]. 또한 학습 기법을 통해 게임 내 NPC 들을 발전시키려는 시도도 있는데, Q-Learning 기법을 사용하여 슈팅 게임에서 좀 더 지능적인 움직임을 생성시키거나 [7], 신경망 학습을 통해 대전 액션 캐릭터를 진화시켰던 연구 사례가 존재한다 [8]. 또한 다양한 진화형 알고리즘을 NPC 가 아닌 게임 제작 프로세스에 적용하여 레벨 지형을 자동화시켜 생성하거나 [9], 게임 제작 자체를 점진적으로 구성시켜 나가는 연구들도 소개되었다 [3].

본 논문에서는 기존에 NPC 학습 부분에 주로 적용되었던 유전자 알고리즘을 게임 내 레벨 디자인 부분에 새로이 적용함으로써 게임을 구성하는 모든 요소들의 최적화된 조합으로 레벨이 생성되도록 하는 것을 목표로 한다.

## 3. 유전자 알고리즘

유전자 알고리즘은 자연의 진화과정과 유전법칙에 영향을 받은 메타 휴리스틱 알고리즘이다. 유전자 알고리즘은 일반적으로 1) 개체의 모집단 정의, 2) 적합도(Fitness)에 따른 선택 과정, 3) 후손을 생성하기 위한 교배(Crossover), 그리고 새로운 후손 발생을 유도하기 위한 4) 돌연변이(Mutation) 단계를 거치며 최적의 해를 탐색한다 [10].

유전자 알고리즘은 풀고자 하는 문제에 대한 가능한 해들을 정해진 형태의 자료구조로 표현한 다음, 이들을 점차적으로 변형함으로써 점점 더 좋은 해들을 생성하게 된다. 유전자 알고리즘을 이용하여 문제에 대한 해를 찾기 위해서는 두 가지 작업이 필요하다. 하나는 풀고자 하는 문제에 대한 가능한 해를 염색체의 형태로 표현하는 것이다. 다른 하나는 각 염색체가 문제를 해결하는데 얼마나 적합한지 측정하기 위한 평가함수를 결정하는 것이다. 유전자 알고리즘은 복잡한 해공간의 탐색능력이 우수하여 변수와 제약이 많은 대형 수리문제를 푸는데 적합한 기법이고, 또한 범용성이 좋아 다양한 문제에 적용이 용이하다는 장점을 갖는다.

레벨 디자인은 게임 내 모든 시스템이 복합적으로 적용되는, 구성 요소들의 최적화된 조합을 찾아내는 대표적 업무 분야이다. 유전자 알고리즘을 게임 내 레벨 디자인에 반영하기 위해서는 레벨 디자인의 구성요소를 유전자 형태로 효율적으로 재정의하여 표현해야 하며, 구성된 레벨 디자인의 평가 함수에 대한 명확한 정의가 이루어져야 한다. 이를 위해 다음 절에서는 본 논문에서 게임 내 구성요소에 대한 모델링 작업을 소개하고자 한다.

#### 4. 게임 레벨 디자인 구성 요소 모델링

게임은 다수의 파라미터와 수식으로 구성된 시스템으로 플레이어에게 해당 게임 주제에 맞는 몰입감과 재미를 주는데 그 목적이 있다. 이러한 게임을 모델링하기 위해서는 게임을 구성하고 있는 다수의 구성 요소에 대한 모델링 작업이 선행되어야 한다. 이러한 모델링 작업은 게임의 장르에 영향을 받지 않는 범용성을 가지고 있어야 하며 동시에 게임의 특징적인 부분을 표현해 낼 수 있는 특징성을 가지고 있어야 한다. 본 논문에서는 이를 위해 적용 대상 범위를 게임의 가장 대표적인 시스템 중 하나인 몬스터와 플레이어 간의 기본적인 전투가 발생하는 전투 기반 게임으로 설정하였다. 그리고 게임 내 게임 구성요소를 스테이지, 파라미터, 몬스터 종류, 몬스터의 배치, 지형 요소, 전투 수식으로 선별한 뒤 후 이를 간략화시켜 적용하였다. 본 논문에서 선별한 구성요소들은 게임 내 장르와 특성에 영향을 받지 않는 최소한의 요소들로서, 게임의 가장 간략화된 모델인 2D 횡스크롤 액션 게임을 그 적용대상으로 하였다.

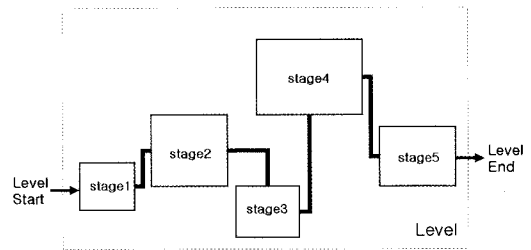
##### 4.1 스테이지

게임 내 레벨은 플레이어 캐릭터의 HP (Health Point) 자동 회복 여부에 따라 크게 연속적인 레벨과 불연속적인 레벨로 분류할 수 있다. 연속적인 레벨 환경 내에서 플레이어 캐릭터의 HP 는 초기에 최대치로 채워진 이후에 회복이 되지 않는 반면에, 불연속적인 레벨 환경에서는 일정 구간 별로 플레이어 캐릭터의 HP 가 최대치로 회복되게 된다. 본 논문에서는 게임의 난이도가 상대적으로 낮고 독립적으로 스테이지를 구성할 수 있는 불연속 레벨을 모델로 하였다. [그림 1]은 불연속 레벨과 이에 포함된 스테이지 구조를 표현한 그림이다. 관련한 플레이 규칙은 아래와 같다.

레벨 플레이 규칙:

1. 레벨 진행은 순차적으로 진행된다.

2. 레벨은 여러 개의 스테이지로 구성된다.
3. 스테이지는 강제로 일정 속도로 스크롤된다.
4. 스테이지 내 몬스터들은 순차적으로 주인공을 공격한다.
5. 스테이지 내 모든 몬스터들을 제거 시 해당 스테이지를 클리어한다.
6. 각 스테이지를 클리어 시 주인공 HP 는 회복된다.



[그림1] 스테이지와 레벨 구성도

##### 4.2 재미와 긴장도

게임의 재미를 수치화시키는 부분은 매우 불확실한 부분에 속한다. 게임 내 재미 부분은 플레이어의 주관적 경험에 크게 의존하는 관계로 이를 정확히 정량화 시키는 부분은 현실적으로 어렵다. 그러나 이를 게임 내 사용되는 다양한 수치를 근거로 예상치를 도출해 낼 수는 있다. 본 논문에서는 게임의 재미 부분을 게임 내 공간 상의 긴장도에 비례하는 것으로 가정하였다. 즉 게임 내 플레이어가 긴장도가 높은 특정 공간 클리어 시에는, 긴장도가 낮은 공간을 클리어 시 보다 좀 더 높은 성취감과 재미를 얻을 수 있을 것으로 가정하였다. 이는 일반적인 도박에서의 더 많은 배팅을 하였을 경우 승리 시 더 많은 기쁨을 얻는다는 점과, 특정 스테이지를 죽지 않고 아슬아슬하게 클리어 시 많은 긴장감과 성취감을 느낄 수 있다는 경험적 사실에 근거하였다. 본 논문에서는 이러한 긴장도를 스테이지 클리어 시 플레이어 캐릭터의 남아있는 HP 잔량을 반비례한다고 설정하였다. 본 논문에서는 가우시안 함수를 사용하여 긴장도를 계산하도록 하였다 (수식 1). 여기서  $t_c$  는 긴장도를 의미하는 텐션값 (Tension Value)을,  $h_c$  플레이어 캐릭터의 현재 HP 수치를 나타낸다. 값은 Max Hp 크기에 비례하여 설정하도록 하였다.

$$t_c = e^{-\frac{(h_c^2 / 2\sigma^2)}{2}}, \quad (0 \leq h_c \leq \text{MaxHP}) \quad (1)$$

##### 4.3 파라미터

게임 내에서 사용되는 파라미터들은 그 게임의 장르, 주제, 플레이 패턴에 따라 변동성이 심하다. 그러나 그 범위를 몬스터와의 대결이 일반적으로 발생하는 게임으로 한정시킨다면 해당 파라미터 범위를 소수의 핵심 파라미터로 축소시킬 수 있다. 본 논문에서는 레벨 디자인 밸런싱을 위한 가장 필수적으로 생각되는 아래의 두 개의 파라미터를 선별하여 이를 기반 파라미터로 사용하였다. 이는 주인공 캐릭터와 몬스터에게 동일하게 적용하였다.

#### (1) Health Point

게임 내 대상 오브젝트의 생명력을 의미한다. 외부 요인에 의해 피해를 입을 때마다 수치가 줄어들게 된다. 몬스터의 HP가 0이 되게 되면 해당 몬스터를 플레이어가 물리친 것으로 가정하여 처리한다. 플레이어 캐릭터의 HP가 0이 되면 몬스터에 의해 사망한 것으로 처리하여 게임 플레이를 중단시키고 페널티를 부여한다. 플레이어에 대한 페널티는 게임 상에서 플레이어로 하여금 본능적인 생존을 유도시키는 심리적 효과를 발생시키고 게임 플레이 규칙을 명확히 할 수 있는 가장 효과적인 방법이 된다. HP는 게임 내 특별한 아이템에 의해 회복될 수 있도록 하였다.

#### (2) Damage per Second (DPS)

게임 내 주인공 캐릭터의 초당 공격력을 의미한다. 매 초마다 이 수치만큼의 피해를 공격 대상자에게 줄 수 있다. 이 파라미터에는 주인공 캐릭터의 일반 공격력, 무기 공격력, 몬스터의 방어력, 공격 속도, 스킬 공격력, 무기구 수치 등의 기초 수치들이 종합되어 반영된 것으로 시뮬레이션 대상이 되는 파라미터 공간을 크게 감소시켜주는 2차 유도 파라미터이다. DPS 수치가 높을수록 위협 요소인 적 NPC의 HP를 빠르게 감소시켜 제거함으로써 플레이어 캐릭터의 HP를 보전하고 이를 통해 생존률을 높일 수 있게 된다.

### 4.4 오브젝트

#### (1) 몬스터

몬스터는 게임 내에서 플레이어와의 전투 인터랙션을 가지게 되는 게임 내 위기 요소이자 성장에 대한 기회적 요소이다. 실제 게임 내에서는 다양한 스타일의 몬스터가 존재하여 플레이어에게 다양한 전투 경험을 제공하게 된다. 이를 모델링 시키는데 있어 본 논문에서는 선별된 HP와 DPS 파라미터 값에 차이를 두어 대상이 되는 몬스터의 종류를 하급 몬스터, 중급 몬스터, 상급 몬스터의 3개 클래스로 구분하였다. 하급 몬스터는 게임 내 플레이어에게 쉬운 성취감을 제공하고, 중급

몬스터는 게임 플레이 도중의 긴장감을 불러 일으키며 상급 몬스터는 게임 플레이에 대한 극적 성취감을 불러일으킬 수 있도록 하는데 그 방향성을 두었고 이에 맞추어 각각의 세부 파라미터 설정을 하였다.

#### (2) 아이템

게임 내에서 몬스터가 플레이어에게 긴장을 제공하는 허들 요소라면 게임 내 아이템은 플레이어의 긴장을 낮추어 주는 행운적 요소라 할 수 있다. 게임 내 아이템 효과는 게임에 따라 매우 다르지만 본 논문에서는 위에서 선별한 두 개의 파라미터 중 HP 부분에 혜택을 주는 HP 증가 아이템을 기본으로 구성하였다.

#### (3) 트랩

아이템이 게임 내 긍정적이고 행운적 요소라면 트랩은 게임 내 불행적 요소이다. 플레이어가 트랩과 충돌하게 되면 높은 확률로 플레이어 캐릭터 파라미터를 약화시키게 된다. 본 논문에서는 HP 하락 스킬 설정으로 트랩을 구성하여 게임의 불확실성 요소를 부분적으로 강화시키고자 하였다.

### 4.5 전투규칙

모든 게임에는 그 기획 의도에 맞는 다양한 스타일의 전투 시스템들이 존재하다. 이러한 전투들은 주인공 캐릭터와 몬스터와의 근본적인 반복 인터랙션으로, 결국 간략화된 수식인 전투 공식으로 추상화시킬 수 있게 된다. 본 논문에서는 위에서 제안한 2개의 파라미터를 사용하여 다음과 같은 전투 규칙을 설정하여 전투 처리를 하였다. 공격 속도 및 공격의 기회는 모두 DPS 파라미터에 내재되어 있는 것으로 반영함으로써 수치적 단순성을 가질 수 있도록 하였다.

#### 전투 규칙:

1. 매 턴마다 주인공과 몬스터는 DPS 수치만큼 상대방 HP에게 데미지를 입힌다.
2. 주인공과 몬스터의 HP가 0이 되게 되면 사망처리한다.
3. 해당 스테이지 내에 또 다른 몬스터가 존재하게 되면 연속전투가 발생하여 주인공 HP는 데미지를 입은 상태로 다음 몬스터와 싸우게 된다.

### 4.6 오브젝트들의 배치

게임 내에서의 레벨은 특정 지형 위에 인터랙션이 가능한 다수의 오브젝트들의 배치로 생성되게 된다. 레벨에서의 재미와 긴장은 이러한 오브젝트들이 어떻게 조합되어 배치되어 있는가에 직접적인 영향을 받게 된다. 좋은 레벨 디자인이란 게임 내 플레이어에게 포기하지 않을 수준의 적절한 허들을 제공하고 이를 플레이어가 극복할 수 있도록 유도함으로써 극복 후에 성취감과 만족감을 주는데 있다. 만약 게임 내 허들이 너무 높거나 낮게 된다면 이러한 성취감에 대한 강도가 떨어지게 되어 결과적으로 플레이어가 게임에 대한 흥미를 잃게 만드는 요인이 되게 된다.

본 논문에서는 사전에 게임 디자이너로부터 원하는 수준의 레벨 디자인 기본 수치를 입력받고 유전자 알고리즘 기반의 시뮬레이션을 통해 자동으로 생성된 레벨이 입력된 기본 수치와 근사화될 수 있도록 반복적으로 구성 요소 조합을 시도하게 된다.

## 5. 유전자 알고리즘 적용

유전자 알고리즘을 레벨 디자인 최적화에 적용하기 위하여 다음의 5 단계 프로세스를 통해 하나의 세대를 생성하였고 최적화된 값이 나올 때까지 진화시켰다. 각각의 단계별 구성은 다음과 같다.

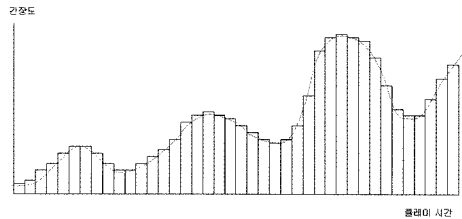
### 5.1 목표값 설정

본 논문에서 목표값은 레벨 디자이너가 각 레벨에서의 원하는 텐션값이다. 게임 레벨은 게임 디자이너가 의도하는 수준의 연속된 텐션을 제공하는 것으로 가정하였고 유전자 알고리즘을 통해 주어진 텐션값을 만족시킬 수 있는 최적화된 조합을 찾아내는 것을 목표로 하였다.

게임 디자이너로부터의 텐션값의 입력은 배지어 커브로 생성되는 곡선 함수를 이산화시켜 이루어지게 된다. 이산화 시 구간 정밀도는 레벨을 구성하고 있는 각 스테이지 개수에 맞추어지게 된다. 이는 텐션 함수를 좀 더 세분화시켜 이산화시킬수록 해당 레벨을 구성하는 세부 스테이지 개수는 늘어 나게 되고 이는 레벨을 구성하는 유전자의 길이를 증가시킴으로써 가능하다.

본 시스템에서는 이러한 형태의 수치화된 재미 입력 모델을 선택함으로써 게임 디자이너가 원하고자 하는 텐션값을 손쉽게 레벨 플레이 시간 공간상에서 연속적으로 배분할 수 있도록 하였고, 시스템은 이를 받아들여 해당 텐션값 수준에 맞는 레벨 셋을 자동으로 생성시키게 된다. 가장 일반적으로 생각해볼 수 있는 레벨 내 텐션 구조는 레벨 내 시나리오

스크립트와 연계된 기-승-전-결 형태의 구조를 예를 들 수 있다 [그림 2].



[그림 2] 텐션 그래프와 이산화된 텐션값

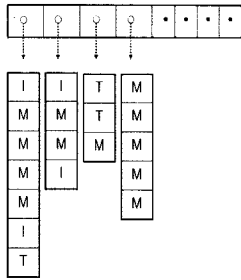
### 5.2 유전자 생성

본 논문에서 풀고자 하는 문제는 텐션 목표값에 수렴되는 최적화된 게임 내 오브젝트 조합을 찾는 것이다. 이를 위해 본 시스템에서는 하나의 염색체를 스테이지를 상징하는 다수의 유전자(Gene)으로 구성하였다. 각 염색체는 1 차원 구조이다. 유전자 생성 시 하나의 유전자 내에는 해당 스테이지를 구성하는 다수의 인터랙션 오브젝트들이 연결된다. 시스템은 각 유전자별로, 연결되어 있는 오브젝트들과 가상의 플레이 시뮬레이션을 돌려 최종적으로 플레이어 캐릭터의 남아있는 HP 값을 가지고 수식 1 에 따라 결과 텐션값을 생성시키게 된다. 플레이 시뮬레이션 시에는 4 절에서 제안한 모델링 규칙들을 따랐다. 생성된 결과 텐션값은 최종적으로 해당 유전자에 실수값으로 인코딩(Encoding)되게 된다. 결과적으로 유전자에 인코딩 된 실수값은 해당 스테이지를 플레이하였을 때의 게임 내 구성요소 인 아이템, 트랩, 몬스터와의 가상 인터랙션 결과가 반영된 누적 텐션값이라 할 수 있다. [그림 3]은 본 논문의 염색체 인코딩 구조도이다. 하나의 유전자들은 다양한 오브젝트들을 - M: 몬스터, I: 아이템, T: 트랩 - 리스트형태로 가지게 된다. 본 시스템에 의해 이러한 오브젝트들을 기준으로 각 유전자별로 플레이 시뮬레이션되어 나온 결과 텐션값을 [그림 4]와 같이 인코딩하게 된다.

### 5.3 선택

유전자의 적합도를 측정하는 함수로 본 논문에서는 목표값과 염색체 간의 최소 자승 오차법 (Least Mean Square Error Method)을 사용하였다. 각 유전자 단위로 최소 자승 오차법에 의해 구해진 결과값으로 염색체 단위의 분산값을 구했고 이 값을 염색체 선택의 적합도 기준으로 삼았다. 이를 가지고 선택

시에는 적합도를 기준으로 한 순위 기반 선택 방법 (Ranking Selection)을 사용하여 최상위 2 개의 염색체가 선택되도록 하였다.



[그림 3] 염색체 구조도

10.04	2.456	5.34	0.0	9.34	30	90.01	6.85	25.03	0.0
0.04	0.48	15.3	3.0	21.34	25	19.01	18.3	0.03	7.0
3.6	4.46	6.3	4.5	15.34	17.0	8.01	5.3	0.03	1.0
95.6	21.46	25.3	13.5	5.34	12.0	32.01	8.31	0.631	10.0

[그림 4] 염색체 인코딩 형태

### 5.4 교배

교배는 2 개의 개체간에 염색체를 부분적으로 서로 바꿈으로써 새로운 개체를 생성하는 단계이다. 이 때 부모의 형질이 자손에게 적절히 계승되는 형질 유전성 (Character Preservingness)을 가질 수 있도록 균일 교배 (Uniform Crossover) 방식을 선택하였다. 이것은 각 유전자가 독립적으로 교환될 수 있도록 하는 것으로서 마스크를 씌워 비트가 0 일 경우에만 유전자를 교환하였다. 마스크 설정의 기준은 각 유전자별 적합도 수준으로 일정 수치 이상의 적합도를 가진 유전자들만 상호간에 교배가 될 수 있도록 하였다.

### 5.5 변이

변이는 개체에 근접한 새로운 개체를 생성하는 국소적인 랜덤 탐색의 일종이다. 또한 집단에서 잃어버린 유전형질을 복구하여 다양성을 유지하기 위한 수단으로도 사용된다. 본 논문에서는 돌연변이 확률을 1.0% 수준으로 설정하여 변이시켰다. 변이로 선택된 유전자는 해당 내부 오브젝트들이 초기화되어 모두 다시 재조합되어 인코딩되도록 하였다.

### 5.6 레벨 생성

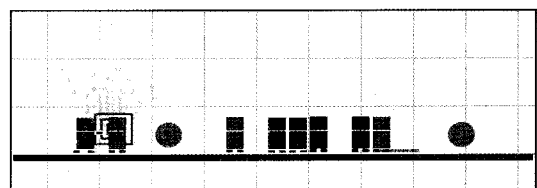
유저가 입력한 텐션 목표값과 가장 유사한 텐션값을 생성해 내는 염색체가 선별되게 되면 시스템은 이 염색체를 기준으로 레벨을 생성하게 된다. 이 때 해당 염색체에 연결되어 있는 오브젝트 리스트들은 각 염색체 생성 시 시스템이 수행한 플레이 시뮬레이션과 동일한 1 차원 배치 순서와 간격에 따라 레벨 내 배치 되게 된다.

## 6. 구현 결과

본 시스템은 Windows XP 운영 체제, Intel Core2 Duo 1.83GHz, 1GB 메모리의 환경에서 C++와 DirectX 9.0, Game Maker 7.0 을 사용하여 구현되었다. 유전자 알고리즘을 제어하기 위한 주요 진역 변수 값들은 최대 세대수는 1,000 회, 집단의 크기는 10, 돌연변이 확률은 1.0%로 하였다. 종료 조건으로는 빠른 최적해를 찾는 부분에 주안점을 두어 수렴된 염색체의 수가 일정 비율 이상이면 종료하도록 하였다. 각 7 개의 구성요소로 조합되는 10 개의 스테이지에 대해 목표치에 대한 95% 수준의 최적해를 찾는 데 소요된 평균 세대수는 약 520 세대였다.

본 논문의 적용 대상으로 2D 횡스크롤 액션 게임을 그 대상으로 실험해 보았다. 대상이 된 모델은 고전 게임 중 하나인 2D KungFu Master [11] 게임 시스템이다. 횡스크롤 액션 게임에는 유저가 컨트롤 할 수 있는 요소가 제한되어 있다는 점에 있어 본 논문의 실험 결과를 직접적으로 체감하기 용이하다. [그림 5]는 본 시스템에 의해 생성된 데이터를 기반으로 한 액션 게임의 실행화면이다.

본 시스템에 의해 생성된 결과물들을 [표 1][3] [5]와 같다. [표 2][4][6]은 각각의 레벨 구성에 사용된 오브젝트 리스트이다. 각각의 테스트는 전체 스테이지 길이를 고정시킨 상태에서 해당 레벨을 구성하는 오브젝트 구성요소를 차별화시켜 그 효과를 테스트 해보았다. 테스트 1 은 두 종류의 NPC 들만 레벨을 구성하였고, 테스트 2 는 세 종류의 NPC 들과 회복 아이템으로 레벨을 구성해 보았다. 테스트 3 은 여기에 추가적으로 트랩형 아이템을 배치하여 레벨을 구성해 보았다.



[그림 5] 횡스크롤 액션 게임 결과 화면

[표 1] 테스트 1 최적해 결과표

스테이지	목표값	최적값	조합
1	10	11	JJJJ
2	20	18	JJJJJJ
3	30	28	MJJJJ
4	40	40	JJJMJMJ
5	30	26	MJJJJJ
6	50	45	MMJJJMJJJ
7	60	58	MJJJMJJMM
8	70	67	MMMJJJMMJJ
9	80	79	MMJJMMJMMMJ
10	70	68	MJJJMJJMMMJJ

[표 2] 테스트 1 레벨 디자인 구성요소

번호	오브젝트명	파라미터	수치/기능
1	주인공	HP	100
		DPS	20
2	일반 적 (J)	HP	80
		DPS	10
3	중간 보스 (M)	HP	40
		DPS	15

[표 3] 테스트 2 레벨 최적해 결과표

스테이지	목표값	최적값	조합
1	10	11	M
2	20	21	JJMJJ
3	30	28	JMJJJJ
4	40	35	JMMJJMJ
5	30	28	JJMJJJJJ
6	50	48	JID
7	60	62	MMMJJMJJM
8	70	74	JDMJM
9	80	85	DDIM
10	70	68	JJMJJJMJJ

[표 4] 테스트 2 레벨 디자인 구성요소

번호	오브젝트명	파라미터	수치/기능
1	주인공	HP	100

		DPS	20
2	회복 아이템 (I)	효과	HP 25% 회복
3	일반 적 (J)	HP	20
		DPS	10
4	중간 보스 (M)	HP	40
		DPS	15
5	대형 보스 (D)	HP	80
		DPS	20

[표 5] 테스트 3 레벨 최적해 결과표

스테이지	목표값	최적값	조합
1	10	9	JJJ
2	20	17	JJRJ
3	30	27	JJMR
4	40	45	TJJT
5	30	34	MJJMM
6	50	44	RJJTMJJ
7	60	62	DIJ
8	70	72	DMJJ
9	80	85	DMT
10	70	73	TTMTRM

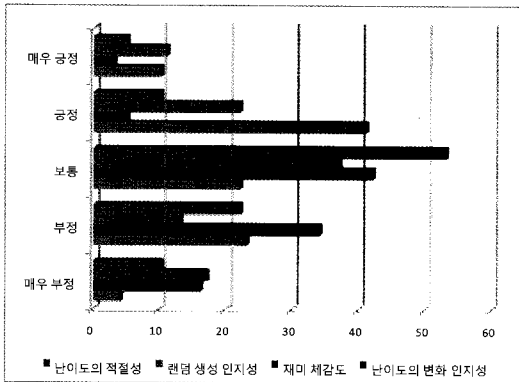
[표 6] 테스트 3 레벨 디자인 구성요소

번호	오브젝트명	파라미터	수치/기능
1	주인공	HP	100
		DPS	20
2	회복 아이템 (I)	효과	HP 25% 회복
3	일반 적 (J)	HP	20
		DPS	10
4	중간 보스 (M)	HP	40
		DPS	15
5	대형 보스 (D)	HP	80
		DPS	20
6	드림통 (R)	효과	50% 확률로 HP 20% 데미지
7	화염 트랩 (T)	효과	50% 확률로 HP 30% 데미지

[표 7] 설문 평가 내용

항목	설문 내용
----	-------

난이도 변화 인지성	기-승-진-결 구조의 난이도 변화를 체감할 수 있었는가?
난이도의 적절성	난이도가 적절하게 구성되었는가?
재미 인지성	플레이도중 일반적인 액션 게임의 재미를 느낄 수 있었는가?
자동 생성 레벨 인지성	게임이 자동으로 생성된 레벨임을 인지할 수 있었는가?



[그림 6] 설문조사 결과도표

본 논문에 의해 생성된 레벨들은 실제 유용성을 확인해 보기 위해 HCI 에서 주로 사용하는 평가 기법 중 하나인 휴리스틱 평가 (Heuristic Evaluation) [12] 방법을 사용하였다. 평가 과정은 다음과 같다. 첫째로 동일한 전문 지식 수준을 지닌 평가단을 구성하기 위해 게임 개발 1 년 차 수준의 개발자 30 명을 모집하였다. 그리고 이들을 대상으로 본 시스템에 의해 랜덤으로 생성된 레벨 4 개 중 임의의 2 개 레벨을 15 분 동안 플레이 한 후 5 개의 평가항목에 대해 리커트 (Likert) 5 단계 척도를 사용하여 평가받았다. 5 개의 평가항목은 [표 7]에 정리하였고 설문 평가 결과는 [그림 6]과 같다.

난이도의 적절성 테스트에서는 평가자의 52% 이상이 보통 이상의 평가를 주었고, 랜덤 생성 인지성 부분에서는 평가자의 69% 이상이 보통 이상의 평가를 주었다. 재미 체감도 부분에서는 평가자의 52% 수준이 부정 이하의 평가를 주었고, 난이도의 인지성 테스트에서는 평가자의 71%이상이 보통 이상의 평가를 주었다. 설문 내용을 전체적으로 요약해 보면, 시스템적으로 생성된 게임 레벨에 대해 대다수가 만족하는

재미는 제공하지 못하였지만, 전반적으로 난이도의 변화를 느낄 수 있었고, 그 난이도는 비교적 긍정적으로 생각되었으며, 랜덤으로 생성된 레벨임을 인지하지는 못했던 것을 알 수 있다.

## 7. 결론

본 논문에서는 유전자 알고리즘을 사용하여 범용적인 게임 레벨 생성을 위한 방법을 새로이 제안하였다. 제안된 시스템은 게임 디자이너가 입력한 텐션값을 기준으로 게임 내 구성 요소를 조합하여 최적화 조합을 생성해 낼 수 있도록 하였다. 본 논문의 방법은 게임 실무에 적용 시 게임 디자인 단계에서 비교적 단순한 입력값만으로도 복잡한 게임 구성 요소로 이루어진 레벨을 근사화시켜 쉽게 도출시킬 수 있음을 실제적으로 보여준다. 또한 제한된 시간 내에 다수의 레벨을 자동적으로 생성시킴으로써 게임 디자이너가 다양한 컨텐츠 구성안을 사전에 직관적으로 확인해 볼 수 있도록 하였다.

본 논문의 결과를 실제 2D 액션 게임에 적용해 보았을 때 기본적인 텐션 변화를 느낄 수 있는 수준의 게임 초기 레벨을 빠르게 생산해 낼 수 있었다. 본 레벨이 세밀하게 튜닝된 최종적인 재미를 자동적으로 제공하기에는 한계가 있지만 실제 게임 디자인 레벨 생성 시에 게임 디자이너가 레벨을 구상하는데 있어 초기 근사화된 레벨 자료로 충분히 활용될 수 있을 것으로 기대한다.

## 참고 문헌

- [1] D. Fu, Y. Houlette, S. Henke, "Putting AI in Entertainment: An AI Authoring Tool for Simulation and Games," *IEEE Intelligent and Systems*, vol. 17, no. 4, pp. 81-84, 2002.
- [2] D. Johnson and J. Wiles, "Computer Games with Intelligence," *IEEE International Fuzzy Systems Conference*, pp. 1355-1358, 2001.
- [3] J. Togelius and J. Schmidhuber, "An Experiment in Automatic Game Design," *IEEE Symposium on Computational Intelligence and Games*, 2008.
- [4] 김미숙, 강태원, "게임 캐릭터의 진화하는 디자인," *한국정보과학회 학술발표대회 논문집*, 제30권, 제1호, pp. 410-413, 2003.
- [5] 권오광, 박종구, "유전 프로그래밍을 이용한 추격-회피



문제에서의 게임 에이전트 학습,” *정보처리학회논문지B*, 제15B권, 제3호, pp. 253-258, 2008.

[6] 이면섭, 조병현, 정성훈, 성영락, 오하령, “대전 액션 게임에서 유전자 알고리즘을 이용한 지능 캐릭터의 성능평가,” *전자공학회논문지*, 제41권, 제4호, pp. 119-127, 2004.

[7] 신용우, “강화학습을 이용한 지능형 게임캐릭터의 제어,” *한국인터넷정보학회*, 제8권, 제5호, pp. 91-94, 2007.

[8] 조병현, 정성훈, 성영락, 오하령, “대전 액션 게임을 위한 신경망 지능 캐릭터의 구현,” *퍼지 및 지능 시스템 학회 논문지*, 제14권, 제4호, pp. 383-389, 2004.

[9] 연제혁, 김성수, 임형준, 이원형, “게임 난이도를 고려한 게임지형 자동 생성 기법에 관한 연구,” *한국인터넷정보학회 학술발표대회 논문집*, 제5권, 제2호, pp. 477-481, 2004.

[10] 문병로, “쉽게 배우는 유전 알고리즘,” *한빛미디어*, 2008.

[11] IREM cooperation, KungFu Master, 1984.

[12] J. Nielsen, and R. Molich, “Heuristic Evaluation of User Interfaces,” *In Proc. of ACM CHI*, pp. 249-256, 1990.

### 〈저자 소개〉



강신진

- 2003년 고려대학교 정보통신대학 컴퓨터학과 (이학석사)
- 2009년 고려대학교 정보통신대학 컴퓨터학과 (박사수료)
- <관심분야> 게임 기획, 컴퓨터 그래픽스



신승호

- 2007년 고려대학교 정보통신대학 컴퓨터학과 (이학석사)
- 2009년 고려대학교 정보통신대학 컴퓨터전파통신학과 (박사수료)
- <관심분야> 게임 프로그래밍, 컴퓨터 그래픽스, 가상현실



조성현

- 1978년 서울대학교 계산통계학과 (이학사)
- 1980년 서울대학교 계산통계학과 (이학석사)
- 1995년 UCLA 컴퓨터과학과 (이학박사)
- 1996년~현재 홍익대학교 게임학부 교수
- <관심분야> 게임 프로그래밍, 게임 그래픽스, 게임 물리, 분산 시스템