

유전 알고리즘을 이용한 플레이어 적응형 몬스터 생성 기법

Players Adaptive Monster Generation Technique Using Genetic Algorithm

김 지 민¹ 김 선 정¹ 홍 석 민^{2*}
Ji-Min Kim Sun-Jeong Kim Seokmin-Hong

요 약

게임 산업이 발전하면서 콘텐츠의 생성 속도보다 훨씬 빠른 속도로 콘텐츠가 소비되고 있고, 플레이어의 게임 숙련도에 적합한 레벨의 게임 콘텐츠들이 지속적으로 제공될 것을 필요로 하고 있다. 이러한 문제를 효과적으로 해결하기 위해 활용되는 방법이 인공지능(Artificial Intelligence, AI)을 이용한 절차적 콘텐츠 생성(Procedural Content Generation, PCG)이다. 본 논문에서는 유전 알고리즘을 이용하여 플레이어에게 적합한 난이도를 가지고 있는 다양한 종류의 몬스터를 자동 생성하는 절차적 방법을 제안한다. 몬스터들의 주요 속성을 유전자로 구성하고 다양한 종류의 몬스터 유전자들로 염색체를 만들어 이용한다. 생성된 몬스터와 플레이어의 전투 시뮬레이션으로 유전자를 평가하여 선택 후 교배한다. 본 논문의 제안 방법을 이용해 플레이어 적응형 몬스터들을 유전 알고리즘에 기반을 두어 절차적으로 생성하고, 염색체 개수에 따라 생성된 몬스터의 다양성을 비교해본다.

☞ 주제어 : 인공지능, 유전 알고리즘, 절차적 콘텐츠 생성, 플레이어 적응형 몬스터 생성

ABSTRACT

As the game industry is blooming, the generation of contents is far behind the consumption of contents. With this reason, it is necessary to afford the game contents considering level of game player's skill. In order to effectively solve this problem, Procedural Content Generation(PCG) using Artificial Intelligence(AI) is one of the plausible options. This paper proposes the procedural method to generate various monsters considering level of player's skill using genetic algorithm. One gene consists of the properties of a monster and one genome consists of genes for various monsters. A generated monster is evaluated by battle simulation with a player and then goes through selection and crossover steps. Using our proposed scheme, players adaptive monsters are generated procedurally based on genetic algorithm and the variety of monsters which are generated with different number of genome is compared.

☞ keyword : Artificial Intelligence, Genetic Algorithm, Procedural Content Generation, Players Adaptive Monster Generation

1. 서 론

게임 산업이 발전하면서 게임을 개발하는데 많은 시간과 자본이 소모되게 되었다. 개발자들이 제작한 콘텐츠들은 플레이어들에 의해 매우 빠르게 소비되고 있으며, 만약 추가 콘텐츠가 없이 같은 콘텐츠들이 반복적으로 소비된다면 플레이어들은 목표를 상실하고 게임을 떠나게 된다[1]. 이러한 플레이어들의 이탈을 막기 위해서 개발자들은 짧은 시간 내에 다양한 콘텐츠를 생성할 필요가 있다. 또한 게임의 개발량이 증가되면서 모든 것을 수작

업으로 처리 할 수 없기 때문에, 최소한의 수작업으로 고품질의 콘텐츠들을 만들어야 할 필요성도 함께 증가되고 있다[2]. 이러한 필요성들로 인해 제안된 방식들 중 하나가 절차적 콘텐츠 생성 방법이다. 절차적 콘텐츠 생성이란 미리 디자인된 콘텐츠를 생성하는 대신, 프로그램 실행 중 알고리즘에 의해 자동으로 콘텐츠를 생성하는 방법을 뜻한다. 절차적으로 생성 가능한 게임 콘텐츠의 예는 지형 맵, 텍스처 이미지, 퀘스트, 아이템 등으로 그 종류가 매우 다양하다. 몬스터와 같이 플레이어와 경쟁할 콘텐츠를 자동으로 생성해야 하는 경우, 몬스터의 속성과 플레이어의 능력 대한 밸런스가 맞지 않는다면 “압도적인 전략”이 나타나거나 다른 전략을 쓸모없게 만들 수 있다[3]. 그리고 밸런스가 맞는다고와 맞지 않는다고에 대한 명확하거나 엄격한 판단 기준이 없기 때문에, 밸런스를 정형적인 방법으로 자동 결정하는 절차적 생성 방법을 구상한다는 것은 매우 어려운 문제에 속한다[3, 4]. 예를 들

1 Graduate School, Dept. of Interaction Design, Hallym University, Chuncheon-si, Gangwon-do, 24252, Korea.

2 Dept. of Advertising and Public Relations, Hallym University, Chuncheon-si, Gangwon-do, 24252, Korea

* Corresponding author (seokminhong@hallym.ac.kr)

[Received 29 September 2016, Reviewed 2 October 2016(R2 10 December 2016), Accepted 15 February 2017]

어 롤플레잉 게임에서는 각 캐릭터의 종족이나 직업 간에 밸런스를 맞춰주는 것도 매우 중요하다. 즉, 게임의 재미를 위해서 이러한 밸런스를 맞추는 것이 게임의 성패를 좌우할 수도 있다[5, 6]. 그러나 일부 게임은 일정한 규칙 없이 필요에 따라 임의로 몬스터와 캐릭터의 밸런스를 조정하는 것으로 알려져 있다. 체계적인 알고리즘이나 수학적 모델 없이 직관적으로 밸런스를 결정하는 것보다는 정형적인 기준에 의해 자동으로 밸런스가 조정된 콘텐츠를 자동으로 생성할 수 있다면 더 효과적인 게임 개발이 가능할 것이다[7]. 또한 게임이 재미있기 위해서는 플레이어에게 적절한 경쟁 상대를 제공해주는 것이 중요한데 이를 위해서는 플레이어에게 적절한 난이도의 도전을 제공해야한다[8]. 적절한 난이도를 제공하지 못하여 다양한 재미요소를 가지고 있음에도 불구하고 시장에서 외면되는 경우도 있었다[9].

본 논문에서는 유전 알고리즘에 기반을 둔 절차적 생성 방법을 이용하여, 플레이어의 능력에 맞는 적절한 난이도를 제공할 수 있도록 몬스터를 자동 생성한다. 기존의 유전 알고리즘을 이용한 절차적 생성 방법들은 몬스터의 탐색 속도를 향상시키는 것과 같은 최적화에 중점을 둔 반면, 본 논문의 제안 알고리즘은 특정 난이도를 가진 다양한 몬스터들을 생성하기 위해, 탐색체와 유전자의 개수, 유전자의 종류를 변화시킨다.

2. 관련 연구

2.1 절차적 콘텐츠 생성

초창기의 게임은 메모리의 제약이 심하여 알고리즘으로 콘텐츠를 생성하는 방식이 제안되었다. 절차적 콘텐츠 생성이란, 던전과 몬스터, 아이템 등을 알고리즘적으로 생성하는 방법을 뜻한다. 1970년대 후반에 ASCII 기반의 게임인 로그(Rogue)를 중심으로 시작되었으며 현대에는 토치라이트와 문명, 스킵 등 다양한 상용게임에서 맵과 아이템, 배경 등을 생성하는데 사용되고 있다.

컴퓨터공학에서 절차적 생성은 알고리즘으로 데이터를 생성하는 방법인데, 컴퓨터 그래픽스에서는 텍스처를 생성하는데 사용되고 비디오 게임에서는 스토리, 퀘스트, 캐릭터, 아이템, 무기 등을 생성하거나 레이싱 게임의 맵 생성, 룰과 보상구조 등을 생성하는데 사용한다.

개발자들은 주로 다음의 세 가지 이유에서 절차적 콘텐츠 생성을 사용하는데, 첫 번째로 개발자로 하여금 더 빠르게 콘텐츠를 생성할 수 있도록 해준다. 두 번째는 게임이

플레이어의 동작에 실시간으로 반응할 수 있도록 해준다. 마지막으로 콘텐츠가 디스크에서 차지하는 용량을 줄여준다[10]. 또한 절차적 콘텐츠 생성 방법을 이용하여 게임에서 인공지능을 탑재한 NPC나 몬스터를 생성하는 연구는 최근 학계와 더불어 많은 기업들도 관심을 가지고 있는 분야이다.

절차적 콘텐츠 생성에 대한 연구로 콘텐츠의 자동 생성 방법뿐만 아니라, 실제로 생성된 콘텐츠가 플레이어에게 얼마나 흥미를 유발시킬 수 있는가도 중요한 연구 주제이다. 생성된 콘텐츠가 플레이어의 흥미를 이끌어내지 못하면 의미가 없기 때문이다. 그래서 완성도 있는 게임의 콘텐츠 제작을 위하여, 플레이어가 얼마나 게임 콘텐츠에 흥미를 느낄 수 있는지에 대한 조사가 필요하다. 게임 콘텐츠에서 몰입이란 재미, 호기심, 통제, 주의집중 등의 요소로 구분된다[11]. 이와 관련된 연구들은 Flow 몰입 이론을 게임 난이도 조절에 적용시키는 기법[12], 속성 데이터의 분석을 통해 개인에 대한 맞춤 학습 피드백을 제공하는 방법[13], 플레이어의 특성에 맞추어 퀘스트의 주요 사건들을 조합하여 새로운 퀘스트를 생성[14] 혹은 유전 알고리즘을 이용하여 몬스터를 생성하는 연구[15] 등이 진행되었다. 이와 같은 연구들은 플레이어의 실력에 맞는 콘텐츠를 생성하고 적은 데이터를 이용하여 콘텐츠를 생성하는 방안을 제시했다.

2.2 유전 알고리즘

유전 알고리즘은 진화 연산의 한 가지 방법으로 진화 생물학 분야의 영향을 받은 알고리즘이다. 다윈의 적자생존의 진화과정을 모방한 메타 휴리스틱 알고리즘이다. 최적의 해를 구하는 방법으로 Holland가 유전 알고리즘의 기본적인 형태에 대해 제안한 후, 유전 알고리즘은 여러 분야에서 사용되었다[16].

유전 알고리즘은 일반적으로 개체의 모집단을 정의하는 ‘생성’, 적합도에 따른 ‘선택’, 후손을 생성하기 위한 ‘교배’, 새로운 후손 발생을 위한 ‘돌연변이’ 단계를 반복하며 변형함으로써 최적의 해를 생성하는데, 유전 알고리즘을 이용하여 문제의 해를 찾기 위해서는 풀고자하는 문제에 대한 가능한 해를 탐색체의 형태로 표현한 후에 탐색체가 문제를 해결하는데 얼마나 적합한지 측정하여 해를 탐색한다[17].

게임 AI에서 절차적 콘텐츠 생성이 많은 연구가 이루어지는 이유는 다음과 같다. 첫 번째로 동적인 절차적 콘텐츠 생성을 이용하면 기존의 게임과는 다르게 메모리의 소비를 줄일 수 있다. 일반적인 게임은 콘텐츠의 사용을 위해 미리 메모리에 해당 리소스를 올려두게 되는데, 절

차적 콘텐츠 생성에서는 필요에 따라 콘텐츠가 생성되기 전까지는 이러한 리소스에 할당할 메모리가 필요하지 않기 때문이다. 절차적 콘텐츠 생성을 사용하는 두 번째 이유는 수동으로 게임의 콘텐츠를 생성하는데 드는 많은 비용 때문이다. 세 번째 이유는 절차적 콘텐츠 생성을 사용하면 완벽하게 새로운 형태의 게임을 만들어 낼 수 있기 때문이다. 만약 실시간으로 충분히 다양한 형태의 새로운 콘텐츠를 생성해 낼 수 있다면, 진정한 의미의 끝이 없는 게임이 가능해 질 것이다[10,18].

3. 구 현

게임은 다수의 파라미터와 규칙으로 이루어진 시스템으로 이러한 게임을 제작하기 위해서는 게임을 구성하고 있는 요소에 대한 모델링 작업이 선행되어야 한다. 유전 알고리즘을 이용하여 다양한 종류의 몬스터들을 생성하기 위하여 본 논문에서는 롤플레이 게임의 대표적인 시스템인 던전 탐험을 기반으로 하는 게임으로 설정하였다.

3.1 게임 규칙

(표 1) 몬스터의 파라미터
(Table 1) Monsters' Parameters

Unit	HP	Armor	Damage	Speed	Unit	HP	Armor	Damage	Speed
Peasant	220	0	5.5	2.00	T. Berserker	450	6	34.0	2.31
Peon	250	0	7.5	3.00	Huntress	600	8	23.0	1.80
Acolyte	220	0	9.5	2.50	Mortar Team	360	6	79.0	3.50
Flying Machine	200	8	12.0	2.00	Gargoyle	410	9	27.0	1.40
Priest	370	0	8.5	2.00	DoC Druid Form	580	1	25.5	1.50
Shaman	415	0	8.5	2.10	Meat Wagon	380	2	108.0	4.00
Necromancer	385	0	8.5	1.80	Shadow Wolf	500	0	21.5	1.00
Skeleton Warrior	180	7	19.0	2.00	Rifleman	535	6	28.5	1.50
Obsidian Statue	550	4	7.5	2.10	Dragonhawk Rider	725	7	25.0	1.75
Spirit Wolf	200	0	11.5	1.00	Demolisher	425	8	109.0	4.50
Sorceress	405	0	11.0	1.75	Crypt Fiend	550	6	39.0	2.00
Skeletal Mage	230	6	16.0	1.50	Raider	610	7	34.0	1.85
Banshee	365	0	11.0	1.50	Kodo Beast	1000	1	18.0	1.44
Witch Doctor	395	0	12.0	1.75	Wind Rider	570	6	49.0	2.00
DoT Druid Form	380	0	12.0	1.60	Grunt	800	7	30.0	1.60
Treant	300	5	16.0	1.75	Destroyer	900	9	26.0	1.35
Militia	220	10	17.0	1.20	Hippogryph Rider	765	7	26.0	1.10
Troll Batrider	325	6	20.0	1.80	Water Elemental	900	2	45.0	1.50
Serpent Ward	135	0	43.0	1.50	Avatar of Vengeance	1200	2	30.5	1.35
Archer	245	6	26.0	1.50	Siege Engine	700	8	88.0	2.10
Dire Wolf	300	0	16.5	1.00	Abomination	1175	8	48.0	1.90
T. Headhunter	350	6	34.0	2.31	DoC Bear Form	960	9	47.0	1.50
Dryad	435	6	24.0	2.00	Tauren	1300	9	45.0	1.90
Footman	420	8	17.0	1.35	Knight	985	11	43.0	1.40
Faerie Dragon	450	6	21.0	1.75	Mountain Giant	1600	10	54.5	2.50
Ghoul	340	6	17.5	1.04	Gryphon Rider	975	6	88.0	2.20
Glaive Thrower	300	2	73.0	3.50	Chimaera	1000	8	102.0	2.50
Carrion Beetle	410	2	24.5	1.50	Frost Wyrn	1350	7	123.5	3.00
Spell Breaker	600	9	20.0	1.90	Phoenix	1250	1	68.0	1.40
Spirit Walker	620	0	19.5	1.75	Inferral	1500	6	54.5	1.35

본 논문에서 실험에 쓰일 던전을 구성하기 위해 일반적인 상용게임에서 많이 사용하는 던전을 토대로 단순화하여 구성하였다. 단순화된 던전 생성 및 플레이 규칙은 다음과 같다.

[규칙 1] 던전에는 10마리 임의의 몬스터를 생성한다.

[규칙 2] 몬스터는 순차적으로 등장한다.

[규칙 3] 플레이어는 자동으로 몬스터와 전투를 치른다.

[규칙 4] 플레이어와 몬스터는 서로 한 번씩 공격을 주고 받는다.

[규칙 5] 플레이어가 몬스터를 처치하면 다음 몬스터가 등장한다.

[규칙 6] 마지막 몬스터를 처치하면 던전을 클리어한 것으로 간주한다.

3.2 유전자 구성

게임에서 사용되는 파라미터들은 게임의 장르에 따라 다르지만 플레이어와 몬스터의 전투가 일어나는 게임에서는 핵심 파라미터를 축소시킬 수 있다. 본 논문에서는 (표 1)과 같이 워크래프트3의 유닛데이터를 기반으로 HP, Armor, Damage, Speed와 같은 4개 파라미터를 선별하여 유전자로 사용하였다.

3.2.1 HP

게임 내 대상 객체의 생명력을 의미하며 외부요인에 의해 피해를 입으면 수치가 줄어들고 HP가 0이 되면 죽게 된다.

3.2.2 Armor

외부 요인의 피해를 감소시킬 수 있는 수치로 HP와 함께 몬스터의 방어 변인을 담당한다. 상용게임에서는 방어구에 종류와 무기의 종류에 따라서 Damage의 감소여부를 적용하지만 본 논문에서는 단순화하여 Damage를 퍼센트로 감소시키는 역할을 한다.

3.2.3 Damage

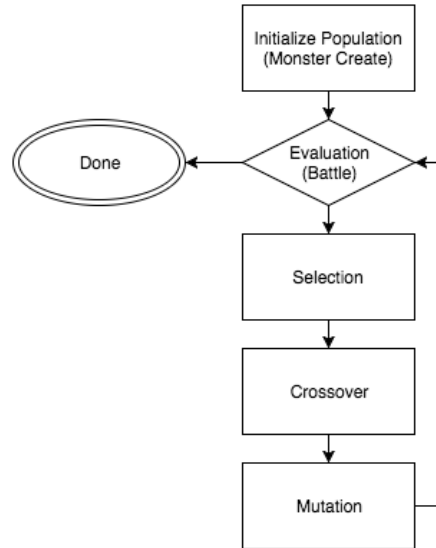
게임에서 객체의 공격력을 의미하며 상대방에게 어느 정도의 피해를 줄 수 있는지에 대한 주요 공격 변인이다. 이 파라미터는 상용게임에서 일반 공격력, 무기 공격력, 스킬 공격력 등으로 나눌 수 있지만 본 논문에서는 단순화하여 일반 공격력만을 사용한다.

3.2.4 Speed

1초에 몇 번 공격을 할 수 있는가를 의미하는 수치이다. 이 수치와 Damage 파라미터를 이용하여 DPS(Damage per Second)를 의미하게 되는데 DPS 수치가 높을수록 상대방의 HP를 빠르게 감소시킴으로서 상대방을 제거하고 플레이어의 생존율을 높일 수 있다.

3.3 유전 알고리즘 적용

유전 알고리즘을 게임 내 몬스터의 절차적 생성에 반영하기 위해서는 몬스터의 속성들(HP, Armor, Damage, Speed)을 이용하여 유전자 형태로 재정의해야 하며, 생성된 몬스터들에 대한 평가함수가 정의되어야 한다. 본 논문에서는 플레이어의 능력에 해당하는 목표값(Rating)에 맞는 몬스터들을 생성하기 위하여, (그림 1)과 같이 목표값 설정, 유전자 생성, 평가 및 선택, 교배, 변이와 같은 5단계의 프로세스를 통해 하나의 세대를 생성하며, 특정한 난이도의 몬스터들이 생성될 때까지 진화시킨다. 본 연구에서는 (표 1)에 나와 있는 몬스터들을 유전자로 정의하고, 1개 염색체는 10개 유전자들로 구성하였다.



(그림 1) 유전 알고리즘의 흐름도
(Figure 1) The flowchart of Genetic Algorithm

3.3.1 목표값 설정

본 논문에서 각 몬스터의 속성은 유전자로 생성되며, 몬스터의 종류는 유전자의 종류로 결정된다. 던전 내의 여러 몬스터들의 유전자가 모여 염색체를 이루고, 몬스터의 개수와 몬스터의 종류를 변화시키며 플레이어의 능력에 맞는 몬스터들이 생성되도록 목표값을 설정한다. 목표값은 HP, Armor, Damage, Speed를 이용하여 다음과 같이 계산된다[19].

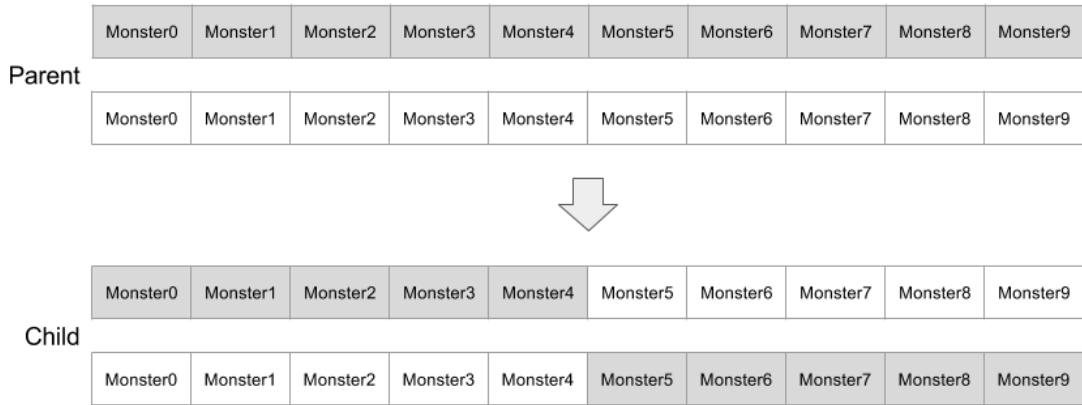
$$Rating = (HP + HP \times Armor / 100) / Damage \times Speed$$

3.3.2 유전자 생성

유전자 생성 작업은 사전에 설정한 몬스터 데이터를 랜덤으로 초기화시켜 염색체를 생성한다.

3.3.3 평가 및 선택

유전 알고리즘을 통해 생성되는 몬스터들은 던전의 적합도를 측정하는 함수에 의해 평가된다. 본 논문에서는 몬스터의 적합도를 평가하기 위하여 각각의 몬스터와 플레이어의 전투를 시뮬레이션 하여 플레이어에 남은 체력을 염색체 선택의 적합도 기준으로 삼았다. 이를 기준으로 적합도 순위 상위 50%의 개체가 선택되도록 한다.



(그림 2) 균일 교배
(Figure 2) Uniform Cross

3.3.4 교배

교배는 2개의 염색체를 부분적으로 서로 바꿈으로서 다음 세대의 개체를 생성하는 단계이다. 이 때 부모의 형질이 자손에게 적절히 유전되도록, (그림 2)와 같이 균일 교배방식을 선택한다. 선택을 통한 평가 함수 결과값에 따라서 더 높은 확률로 교배가 일어나도록 하며, 이는 염색체의 적합도가 높은 염색체의 유전자를 다음 세대에 더 많이 선택되도록 하기 위함이다.

3.3.5 변이

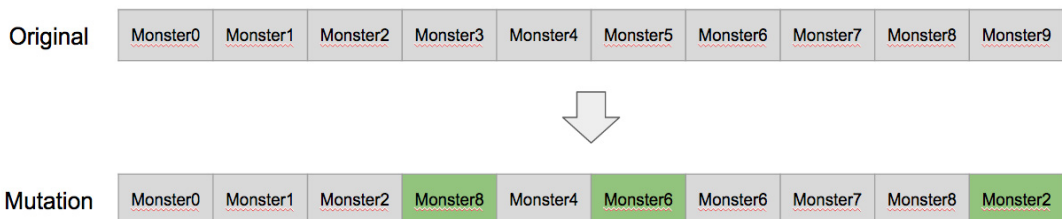
변이는 주어진 해의 유전자 안에 있는 염색체가 임의로 변경되어 다른 해로 변형되는 랜덤 탐색의 일종이다. 교배가 부모 해들을 이용해서 자식 해를 생성하는 반면,

(그림 3)과 같은 변이는 임의의 새로운 해가 탄생하는 것과 같다. 이러한 변이 연산을 수행하면 전체 세대가 모두 지역 최적해만 선택하는 것을 방지한다.

4. 실험 결과

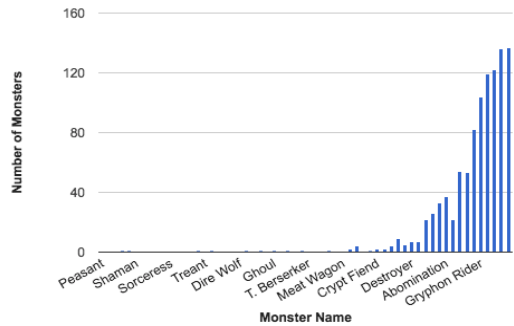
본 논문은 유전 알고리즘을 이용하여 특정한 난이도의 몬스터들을 생성하는 실험을 하였다. 염색체의 개수를 각각 8, 16, 24개로하는 총 3개의 실험군으로 목표값에 적절한 몬스터들 100개를 생성한 결과, (그림 4, 5, 6)과 같은 결과를 보였다. x축은 생성된 몬스터의 이름을 나타내며, y축은 최종 선택된 몬스터의 개수를 나타낸다.

염색체의 개수별 사용된 몬스터의 수는 각각 8염색체에서 47종, 16염색체에서 33종, 24염색체에서 58종이 선

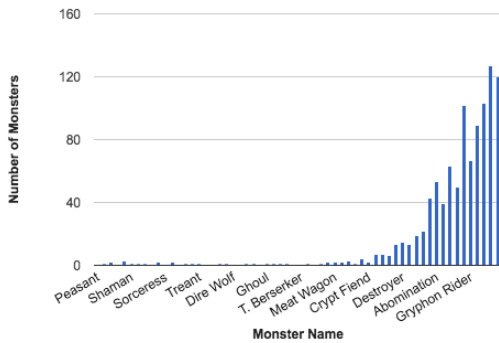


(그림 3) 변이
(Figure 3) Mutation

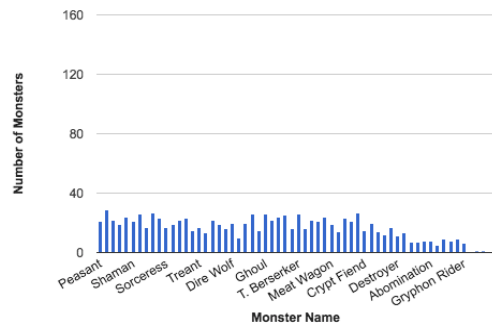
택되었다. 그래프에서 보듯이 24염색체에서만 균일하게 몬스터가 선택되었고 8염색체와 16염색체 실험군에서는 한쪽으로 치우친 결과를 보였다. 그 이유는 염색체의 수가 충분하지 않을 경우, 염색체의 다양성을 확보하지 못하기 때문에 몇 종류의 몬스터만 계속 선택되는 결과가 발생하였다. 24염색체의 경우에는 58종의 몬스터들이 비교적 균일하게 생성됨을 알 수 있다.



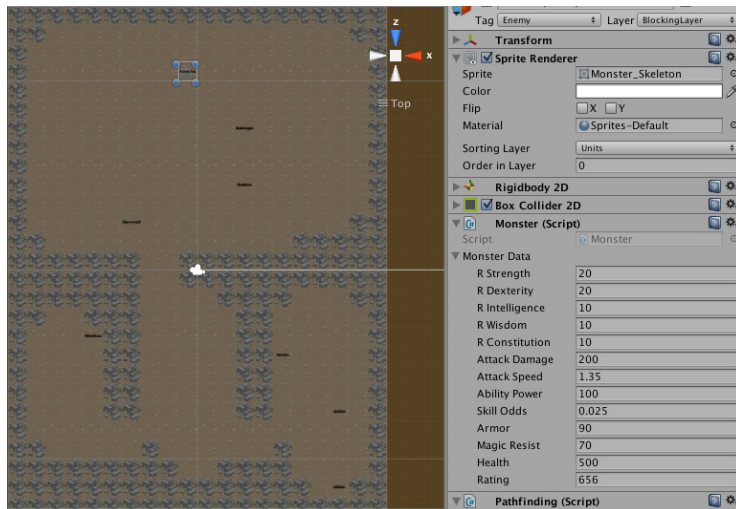
(그림 5) 16개 염색체를 이용한 결과
(Figure 5) The Result with 16-Genome



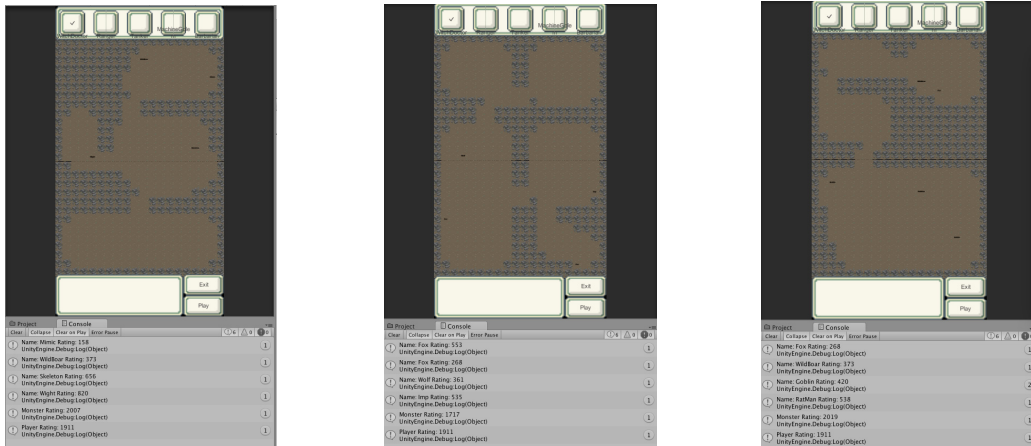
(그림 4) 8개 염색체를 이용한 결과
(Figure 4) The Result with 8-Genome



(그림 6) 24개 염색체를 이용한 결과
(Figure 6) The Result with 24-Genome



(그림 7) Unity3D에서 구현된 결과 화면
(Figure 7) The Screen Shot of Implemented Result using Unity3D



(a) (b) (c)

(그림 8) 다양한 종류의 몬스터 생성 결과

(Figure 8) The Result of Generation of Various Monsters

(그림 7)은 Unity3D에서 본 논문의 제안 알고리즘을 구현한 결과 화면이다. 왼쪽 화면은 플레이어의 능력에 맞도록 설정된 목표값(Rating)에 맞추어 절차적으로 생성된 몬스터들의 위치가 던전 안에 표시되어 있다. 각 몬스터의 위치를 선택하면, 오른쪽 화면에서 몬스터들의 속성과 Rating 값을 확인할 수 있다.

(그림 8)은 설정된 목표값에 따라 유전 알고리즘에 의해 절차적으로 생성된 다양한 종류의 몬스터들의 Rating 값들을 확인하는 결과 화면이다. 아래 콘솔창에서 보면, 같은 목표값에 근사하도록 유전자 알고리즘에 의해 진화하여 각각의 다른 4종류의 몬스터들이 생성되었고, 이들의 Rating 함은 플레이어의 목표값에 근사함을 알 수 있다.

5. 결론 및 향후 연구

본 논문에서는 유전 알고리즘을 통해서 플레이어의 능력치에 맞는 몬스터들을 생성했으며, 각 생성 방법의 염색체수를 다르게 하여 총 3개의 실험군으로 구성하여 시뮬레이션 하였다. 작은 수의 염색체를 갖는 던전에서는 몇 종류의 몬스터만 계속 선택되는 결과를 얻었고, 24개의 염색체를 갖는 던전에서는 다양한 종류의 몬스터들이 세대마다 균일게 몬스터가 선택됨을 알 수 있었다. 즉 염색체의 수가 충분하지 않으면 몬스터의 다양성이 부족해지는 현상이 발생하였다. 그러므로 충분한 염색체를 기반으로 몬스터들을 생성해야 같은 난이도의 게임이라도 다

양한 몬스터가 출현하는 던전이 생성되어 사용자의 흥미를 유발하리라 판단된다.

향후 연구로는 본 논문의 제안 알고리즘이 게임의 흥미를 어느 정도 증진시켰는지에 대해 조사 분석을 할 예정이다. 또한 본 논문의 연구 결과가 상용 게임에 적용되기 위해 다양한 파라미터를 이용하는 유전 알고리즘으로 확장시킬 것이며, 몬스터의 인공지능과 같은 요소를 포함하여 시뮬레이션 할 계획이다.

참고문헌(Reference)

- [1] Yungha Kim., "Making plausible random generated content", NDC, 2010.
- [2] Jubok Kim. "Procedural Terrain and the Raiders Of The Trend", NDC, 2011.
- [3] Keith Burgun, "Understanding Balance in Video Games", Gamasutra, 2011.06.08
- [4] Chan-II ParkO, Hae-Sool Yang. "Balance for Fighting System between Characters", Journal of Korea Game Society, Vol. 7, No. 3, 2007.
- [5] Dong-Seong Choi, Sung Hyun Cho "A Methodology for MMORPG Design Based on the Flow Theory", Korea Game Society, Vol. 8. No. 2. pp.14-16, 2008.
- [6] HyeJung Hyun, Taesik Kim, "An Efficient Algorithm for Character Adjustments in Game Balancing", The

- Korea Contents Association, Vol. 8 No. 1, pp. 340-341, 2008.
<http://dx.doi.org/10.5392/JKCA.2008.8.1.339>
- [7] D. Nicolas, E. N. Yee, and J. M. Robert, "Alone Together? Exploring the Social Dynamics of Massively Multiplayer Online Games," CHI 2006 Proceedings Games and Performances, pp.407-416, 2006(4).
<http://dx.doi.org/10.1145/1124772.1124834>
- [8] Joon Hyun Jeon. Dong Eun Kim. Eui Jun Jeong. "Preliminary Study of a MMORPG Combat Balance Assessment Model", Korean Society For Computer Game, p49~60, vol. 26 No 3, 2013.
- [9] Daesoon Yong., "A study on Game Design for applying fun factor", Game Industry Journal. 2003
- [10] Procedural Content Generation: Thinking With Modules, Ichiro Lambe, 2012,07,05, Gamasutra
- [11] Jennett, Charlene Ianthe. Is game immersion just another form of selective attention? An empirical investigation of real world dissociation in computer game immersion. Diss. UCL (University College London), 2010.
- [12] Cowley, B., Charles, D., Black, M. and Hickey, R., Toward an Understanding of Flow in Video Games, Computer Entertainment, 6, 1-27, 2008.
<http://dx.doi.org/10.1145/1371216.1371223>
- [13] Jinsung Park, Gyuho Lee, Byoungwoo Song, Youn-guk Ha. "Personalized Learning Evaluation System Using Attribute-Based Elo Rating Algorithm". Korea Information Science Vol.2014 No.6, 2014.
- [14] Jun-Woo Kim, Kwang-Hyuk Im. "Developing a Subset Sum Problem based Puzzle Game for Learning Mathematical" Journal of the Korea Contents Association, 2013.
<http://dx.doi.org/10.5392/JKCA.2013.13.12.680>
- [15] Sang-Won Um, Tae-Yong Kim, Jong-Soo choi, "Player dependent difficulty control algorithm Using Genetic Algorithm", Journal of the Korea Computer Games Association, vol. 1, 2002.
- [16] https://en.wikipedia.org/wiki/Genetic_algorithm
- [17] Byung-Ro Moon, Easy Learning Algorithm: An Evolutionary Approach. HanbitMedea 2008
- [18] Julian Togelius, Georgios N. Yannakakis, Kenneth O. Stanley, and Cameron Browne, "Search-Based Procedural Content Generation", EvoApplication 2010, Part 1, LNCS, 6024, pp.141-150, 2010.
http://dx.doi.org/10.1007/978-3-642-12239-2_15
- [19] Hyung-Ryul Son, Game balance story that makes interesting games, HanbitMedea, 2014.

● 저 자 소개 ●

김 지 민



2014년 한림대학교 유비쿼터스 게임공학학과 졸업(학사)
2017년 한림대학교 대학원 인터랙션 디자인학과 졸업예정(석사)
2014년~현재 한림대학 인터랙션 디자인학과 학생
관심분야 : 인공지능, 유전 알고리즘, 절차적 콘텐츠 생성
E-mail : rudengkim@gmail.com

김 선 정



1996년 고려대학교 컴퓨터학과 졸업(학사)
1998년 고려대학교 대학원 컴퓨터학과 졸업(석사)
2003년 고려대학교 대학원 컴퓨터학과 졸업(박사)
2005년~현재 한림대학교 융합소프트웨어학과 교수
관심분야 : 컴퓨터그래픽스, 3D 게임엔진, 가상/증강현실
E-mail : sunkim@hallym.ac.kr

홍 석 민



1989년 연세대학교 사회학과 졸업(학사)
1998년 University of Texas at Austin 광고학 (석사)
2003년 University of Texas at Austin 광고학 (박사)
2003년~현재 한림대학교 광고홍보학과 교수
관심분야 : 사회연결망분석, 뉴미디어
E-mail : seokminhong@hallym.ac.kr