

# 난이도 예측 기반 적응형 퀴즈

## 서버의 설계

김연정<sup>0</sup>곽미라 조동섭

이화여자대학교 과학기술대학원 컴퓨터학과

{chuwoo, mirakwak, dscho}@ewha.ac.kr

## Design of the Adaptive Quiz Server based on Difficulty Prediction

Yeun-Jung Kim<sup>0</sup> Mi-Ra Kwak Dong-Sub Cho

Dept. of Computer Science and Engineering, Ewha Womans University

### 요 약

인터넷 사용자의 증가와 더불어 온라인 게임 사용자가 늘어나고 있으며 이러한 온라인 게임 사용자의 관심을 끌기 위한 여러 종류 게임들이 생겨나고 있다. 그 중에서도 온라인 퀴즈 게임은 네티즌을 끌어들이기 용이하게 구성된 솔루션으로 평가되어진다. 또한 퀴즈게임은 온라인 게임 장점을 가장 잘 반영하고 있는 게임이라고 해도 과언은 아닐 것이다. 현재 대부분의 퀴즈 출제 방법처럼 난이도의 변화를 고려하지 않는 단순히 반복적이고 랜덤하게 문제를 제출하는 것은 퀴즈게임의 흥미를 잃어버리게 하는 요인이 된다. 이러한 단점을 보완하는 사용자 수준과 난이도에 맞는 문제를 온라인에서 제공하는 시스템을 필요로 한다. 이 문제를 해결하기 위해 현재상태 따라 그 문제에 대한 난이도를 예측할 수 있는 난이도 예측 기반 적응형 퀴즈 서버의 알고리즘을 제안한다.

### 1. 서론

초고속 인터넷과 고성능 컴퓨터의 보급 속에서 다양한 게임 개발이 이루어지고 있다. 그와 함께 게임의 대한 관심도 계속해서 증가되고 있으며 게임은 이제 단순한 오락이 아닌 하나의 산업으로 인정받고 있다. 그 뿐 아니라 21세기 미래의 핵심 산업의 중심이 될 것이라고 예견되어진다. 실제로도 고부가가치 산업으로 발전하는 양상을 띤다. 특히 우리나라에서는 온라인 게임은 다른 소프트웨어들에 비해 세계 시장에서 높은 수익을 얻는 분야로 많은 관심과 투자가 이루어지고 있다.

그 중 퀴즈 게임은 인터넷을 하는 중에 네티즌을 자연스럽게 게이머가 되게 하는 등 네티즌을 끌어들이기 용이한 솔루션으로 평가받고 있다. 하지만 지금 이루어지는 형식의 퀴즈 제출 방식은 게이머를 급방 식상하게 만드는 요인이 된다. 현재 대부분의

퀴즈 서버가 하는 문제 제출방식은 문제의 난이도를 결정해 놓고 게이머의 레벨에 따라 그 레벨에 해당하는 난이도의 문제를 임의적으로 추출하는 제출하는 방식이다. 이렇게 문제의 난이도가 고정되어있는 방식은 시간이 지난 후에는 그 문제의 난이도가 떨어졌음에도 불구하고 높은 난이도의 문제로 출제되는 우를 범하게 된다. 이러한 잘못된 게이머의 흥미를 잃게 하는 결정적인 요인이 될 것이다. 다양한 방식 화려한 인터페이스도 중요하나 무엇보다 퀴즈 게임에서 중요한 것은 퀴즈 문제를 어떻게 효과적으로 출제할 것인가에 달려있다. 그러한 문제를 해결하기 위해 게이머들의 레벨에 맞는 퀴즈를 내기 위한 난이도 예측 기반 적응형 퀴즈생성기를 제안하고자 한다.

2장에서는 난이도 예측 기반 적응형 퀴즈생성기를 적용할 수 있는 퀴즈 게임 서버에 대해 디자인하여 보고 3장에서는 퀴즈 게임서버에 필요한 서버의 기본적인 구성 요소들을 정의한다. 그리고 4장에서는

본 연구는 2002년도 두뇌한국 21산업에 의하여 지원되었음.

게이머의 실력에 맞게 퀴즈를 제출하기 위해 퀴즈 문제 알맞은 난이도를 적용시키는 알고리즘과 현재 출제된 문제들에 대한 난이도를 평가하는 방법에 대해 제안해 보겠다. 5장에서는 결론과 향후 연구 과제를 알아보겠다.

## 2. 퀴즈 게임 서버

그림1은 난이도 예측 기반 적응형 퀴즈생성기를 적용하기 위해 디자인한 게임 서버의 전체적인 모습으로 한 개의 서버에 n개의 독립적인 게임 그룹이 존재한다.

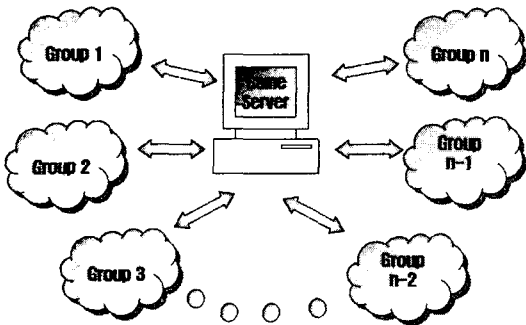


그림 1 게임의 전체적인 구조

각 그룹에는 두 명의 게이머(Creator와 Joiner)와 게임을 하면 다른 게이머(Watcher)들이 게임 진행 과정을 지켜보며 자신도 답과 어느 게이머가 맞출지 배팅하여 점수를 얻는 방식의 퀴즈 게임이다. Creator는 게임 그룹을 생성하는 게이머이고 Joiner는 생성된 그룹에 참가하는 게이머이다. 두 명의 주 게이머의 성적과 답은 다른 게이머에게 전달되지만 나머지 게이머의 데이터는 자신에게만 전달된다. 각각의 게임 그룹이 만들어지며 각 그룹은 독립적으로 돌아간다.

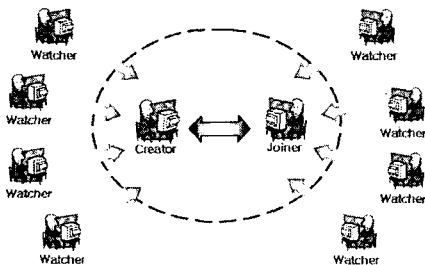


그림 2 한 그룹 내의 구조

## 3. 퀴즈 게임 서버의 기본 요소

게임 서버는 User Session Manager, Evaluation Manager, Adaptive Quiz generator based on Difficulty Prediction, System Manager 의 네 부분으로 구성된다.

System Manager는 데이터 전송을 관리한다. User Session Manager는 게이머가 login 하여 게임이 시작하기 전까지의 게이머의 관리를 하고 게임 끝난 다음 종료한 게임의 사용자와 그룹을 각 리스트로부터 제거한다. Evaluation Manager는 게이머의 성적을 관리한다. Adaptive Quiz generator based on Difficulty Prediction는 Quiz 데이터베이스를 가지고 퀴즈문제의 출제를 관리한다.

### 3.1 System manager

System manager는 데이터 전송을 관리한다. 게임 그룹 내의 게이머간에 메시지 전송과 게이머들에게 퀴즈문제를 전송하고 각 게이머의 답과 배팅 결과를 전송 받는다. 게임이 시작되면 퀴즈 데이터베이스로부터 Adaptive Quiz generator based on Difficulty Prediction를 이용하여 퀴즈를 가져와 게이머들에게 전송한다. 그리고 각 게이머가 답을 입력하면 답을 전송받아 그 게이머가 주 게이머이면 다른 게이머들에게 전송하고 아니면 전송하지 않는다. 전송받은 답들은 Evaluation manager에게 전달한다.

### 3.2 User Session manager

User Session manager는 각각의 게이머가 자신이 원하는 게임에 참가하고 게임을 시작할 수 있는 준비를 한다. 그리고 게임이 종료되어 게이머가 나가면 그 게임을 게임 리스트에서 삭제하고 전체 사용자 리스트에 각 게이머를 삭제한다.

게이머가 login을 하면 같은 id 게이머가 있는지 검색하여, 없으면 그 게이머를 받아들여 전체 사용자에 등록하고 아니면 다시 입력하도록 한다. login을 마친 게이머는 직접 Quiz게임에 참가 모드나, 직접 참가하지는 않으나 Creator와 Joiner의 진행 보며 Quiz를 푸는 모드를 선택한다. 직접 Quiz 게임에 참가 모드에서는 새 게임그룹을 만들 수 있고 이미 만들어져 다른 참가자를 기다리는 게임에 직접 참가할 수도 있다.

### 3.3 Evaluation Manager

System Manager로부터 각 문제에 대한 게이머의

답과 배팅결과(Watcher일 경우)를 전송받아 manager로부터 받은 정답과 비교하여 점수를 매기고 그 결과를 데이터베이스에 저장하고 System Manager에게 전송한다.

일반적인 다중연결리스트의 처리처럼 여기서도 멀티 스레드를 사용해서 클라이언트의 요청을 처리한다. User Session Manager와 System manager는 멀티 스레드를 이용하여 클라이언트를 받아들이고 계속해서 새로운 그룹을 생성하고 게임을 진행 시켜 나간다.

**3.4 Adaptive Quiz generator based Difficulty Prediction**

각 그룹별로 퀴즈 데이터베이스와 연동하여 퀴즈를 출제한다. 이렇게 출제된 문제를 System manager를 이용하여 그 그룹의 게이머에게 넘겨준다. 퀴즈를 출제할 때 현재의 단순한 출제 방식이 아닌 사용에 따라 문제를 다르게 출제하는 알고리즘을 사용한다. System Manager를 이용하여 Evaluation Manager에 그 문제에 답을 전송한다.

**4. Adaptive Quiz generator based Difficulty Prediction (난이도 예측 기반 적응형 퀴즈생성기)**

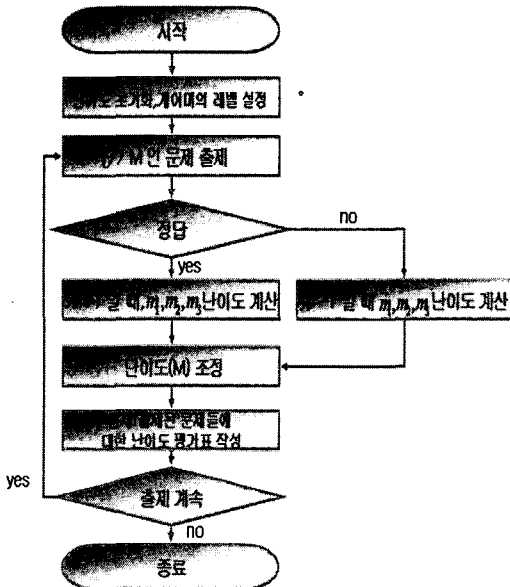


그림3. 난이도 예측 기반 적응형 퀴즈생성기 알고리즘

게이머 수준에 맞는 퀴즈를 출제하기 위해서는 퀴즈문제의 난이도를 정확히 결정하고 그것이 얼마나 정확인지 검토할 수 있는 알고리즘을 필요하다. 난이도 예측 기반 적응형 퀴즈 생성기 알고리즘은 퀴즈문제의 현재 난이도를 과거의 난이도에 의해 예측하여 결정한다. 그리고 한 회에 출제된 문제 set의 평균 난이도를 구하여 얼마나 정확하게 난이도를 결정했는지 시스템 성능 평가를 하는 알고리즘이다.

난이도 예측 기반 적응형 퀴즈생성기의 알고리즘은 그림4와 같다. 퀴즈 게임이 시작되면 각 문제의 난이도를 사용자의 레벨에 맞게 초기화한다. 사용자 레벨보다 낮은 난이도의 문제를 임의로 추출하여 출제하고 문제를 맞추면 λ은 1의 값을 맞추지 못하면 -1의 값을 주어 문제를 맞추는 비율(P)를  $P_n = P_0 + \frac{\sum \lambda_n}{2n}$  구한다. 그러면 이 문제의 난이도(d)는  $1 - P_n$  가 된다. 이식을 이용하여 난이도  $m_1, m_2, m_3$ 을 구하여 난이도를 다시 조정한다. 다음 현재 난이도(M)를 구한다. 표1은 각 문제의 난이도를 구한 난이도 표이다.

No.	P	K	$m_0$	$m_1$	$m_2$	M
1	37	23	0.62	0.68	0.62	0.64
2	54	37	0.5	0.56	0.5	0.52
3	42	36	0.86	0.89	0.86	0.87
⋮						
n	72	32	0.4	0.6	0.43	0.56

표1 난이도 표

P = 출제 문제수, K = 정답수

$n < 10$ 이면,  $m_1 = m_2 = m_3 = P_n$

$n < 50$ 이면,  $m_1 = m_3 = P_n$

$$m_2 = P_n = P_0 + \frac{\sum_{n=9} \lambda_n}{2 \times 10}$$

$n \geq 50$ 이면,  $m_1 = P_n$

$$m_2 = P_n = P_0 + \frac{\sum_{n=9} \lambda_n}{2 \times 10}$$

$$m_3 = P_n = P_0 + \frac{\sum_{n=50} \lambda_n}{2 \times 50}$$

문제의 현재 난이도(M)는 여러 개의 난이도( $m_1$ ,

$m_2, m_3$ )의 가중합으로 구하여진다.  $m_1$ 는 지금까지 출제되는 동안의 난이도,  $m_2$ 는 최근 10번 출제되는 동안의 난이도,  $m_3$ 는 최근 50번 출제되는 동안의 난이도이다. 각각의 가중치는 다르게 주어 현재의 퀴즈문제의 난이도의 특징을 결정할 수 있다.  $M = \alpha m_1 + \beta m_2 + \gamma m_3$  ( $\alpha + \beta + \gamma = 1$ ,  $\alpha, \beta, \gamma$ 는 가중치)인데  $\alpha$ 값을 높이 주면 최근의 생긴 이상점의 영향을 줄일 수 있고  $\beta$ 를 높이 주면 최신 데이터라 더 이용할 수 있는 이점이 있다. 즉 가중치를 어떻게 주느냐에 따라 자신이 원하는 대로 문제 난이도를 어느 정도 조정할 수 있다.

Quiz Set No.	문제 수	정답 수	정답률	평균 난이도
1	24	16	0.66	0.43
2	54	47	0.87	0.19
3	34	21	0.61	0.45
⋮				
n	67	34	0.51	0.53

표2 시스템 성능 분석표

이렇게 구해진 난이도(M)는 현재 제출된 문제들에 대한 난이도 평가에 사용된다. 출제된 각 문제들은 하나의 set으로 구성된다. 표2는 시스템평가용 난이도 표이다. 이 표에는 문제수, 정답수, 정답률, 평균 난이도 등 있다. 평균 난이도는 하나의 게임이 종료할 때까지 구한 난이도(M)의 평균값이다. 이 표에서 처럼 지금까지의 누적된 평균 난이도를 보며 게이머의 레벨에 맞게 문제를 출제했는지 확인할 수 있다.

**5. 결론 향후 연구 과제**

먼저 난이도 예측 기반 적응형 퀴즈생성기를 적용시킬 수 있는 게임서버의 전체적인 구조와 그 서버의 기본적인 구성요소들을 디자인하여 보았다. 현재 퀴즈 출제의 문제점을 해결하기 위해 난이도 예측 기반 적응형 퀴즈생성기의 알고리즘을 제안하였다. 그리고 이 알고리즘의 성능 평가 방법에 대해서 알아보았다. 이 난이도 예측 기반 적응형 퀴즈생성기는 현재 게이머의 상태를 고려하여 퀴즈의 난이도를 난이도들의 가중합으로 결정한 후 퀴즈의 중복을 줄이기 위해 난수를 이용하여 임의로 퀴즈를 추출하는

방법을 택하고 있다.

난이도 예측 기반 적응형 퀴즈생성기를 더욱 효과적으로 사용하기 위해서는 문제를 제공하는 방식과 더불어 효과적인 점수 관리 시스템이 필요하다. 사용자의 수준과 난이도에 맞는 문제를 온라인에서 제공하고 자신이 틀린 문제에 대한 평가와 점수를 실시간으로 서비스하며 게이머의 점수가 서로 유기적으로 영향을 미치는 점수 관리하는 알고리즘의 개발이 필요하다.

**[참 고 문 헌]**

[1] 이만재, 온라인 게임 엔진 기술 동향, 정보과학회지 제 20권 제 1호, 2002년 1월  
 [2] 곽광현, 온라인 게임 개발 사례 : 헬브레스, 정보과학회지 제 20권 제 1호, 2002년 1월  
 [3] 신동일, 신동규, “ 다수 사용자 기반의 온라인 게임 서버의 설계 및 제작,” 전자 공학 회지, 제 27권 제 9호, pp.954-960, 2001년  
 [4] 김정식, “온라인 3D 슈팅게임 만들기”  
 [6] 교육, 첨단 게임 기술 동향, 정보과학회지, 제 15권 8호, 1997년 8월  
 [7] Deloura, Mark, Game Programming Gems, Charles River Media, 2000년 8월 1일  
 [8] 김연정, 퀴즈게임서버의 디자인과 구현, 한국멀티미디어 학술대회 논문집, 제 5권 1호 pp. 469-473, 2002년 5월  
 [9] Sheldon Ross, A, first Course in Probability Forth Edition, MACMILLAN  
 [10] 김연정 능동형 퀴즈 생성을 위한 서버의 설계 및 구현, 한국 게임 학회 학술대회 논문집 pp.174-176, 2002년 7월