

Java를 이용한 최적 미로 게임 개발

정갑중* · 이영준*

*경주대학교

Development of Optimal Maze Path Game Using Java

Gab Joong Jeong* · Yeong Joon Lee*

*Gyeongju University

E-mail : gjeong@gyeongju.ac.kr

요 약

본 논문은 웹기반 게임 콘텐츠로서의 최적 미로 게임 개발에 대한 논문이다. 웹을 이용한 클라이언트 접속자는 자바 애플릿을 이용하여 웹상에서 접근 가능하고 Java Bytecode의 다운로드에 의해 각 클라이언트 접속자의 하드웨어시스템에 무관하게 작동가능하다. 본 논문에서 개발된 최적 미로 게임은 랜덤 미로 생성기, 미로 내 경로 입력기, 가중 최적 경로 탐색기, 및 비교기 등으로 구성되어 있다. 최적 미로 탐색알고리즘으로 탐색된 경로와 사용자가 선택한 경로의 cost 비교를 통하여 사용자의 지리적 인지력을 정량적으로 평가 및 도시함으로써 사용자의 지리적 인지력 향상을 유도할 수 있다.

ABSTRACT

This paper describes the development of an optimal maze path game as web-based game contents. Client user using web can access and run java applet program with download of java byte code with the independence of hardware system. The optimal maze path game developed in this paper consists of random maze path generation module, selected path input module, weighted optimal path search module, and path comparison module. It enhances the cognition faculty of game users with the comparison of the maze path searched by optimal path search algorithm and the selected maze path by game users.

키워드

미로, 웹 기반 게임, 자바, 지리적 인지력, 최적 경로 탐색

1. 서 론

근래의 컴퓨팅 환경은 인터넷의 빠른 확산과 컴퓨터의 성능 향상으로 인해 컴퓨터의 응용 분야 확산을 매우 가속화 시키는 환경을 제공하고 있다. 컴퓨터 전문가들의 각 전문 분야에는 물론이고 컴퓨터의 일반 사용자들도 일상 생활에서 많은 활용성을 제공하고 있으며 특히 고속 인터넷의 보급 확산으로 인해 기술적인 응용뿐만 아니라 오락과 여가 생활에도 인터넷 및 컴퓨터의 활용이 매우 쉽게 이루어진다. 특히 최근에는 인터넷 게임의 도입과 웹브라우저의 발달에 의한

인터넷 게임 사용자의 폭증을 경험하고 있는 실정이다. 이러한 사용자의 폭증은 인터넷을 이용하여 쇼핑이나 여가 시간을 즐기고자하는 이용자가 매우 많아지고 있다. 그러한 시장성의 증가에 비해 실제 이용 가능한 인터넷 콘텐츠들은 아직도 부족한 면이 있고 사용자들의 취향에 맞는 더욱 다양한 인터넷 콘텐츠의 필요성이 증대되고 있다.

본 연구에서는 그러한 많은 인터넷 이용자들의 다양한 취향에 맞는 필요 콘텐츠의 개발 요구에 대한 하나의 새로운 게임 콘텐츠를 개발하였다. 본 연구에서 개발된 웹 기반 게임 콘텐츠는 기존의 정적인 미로 찾기 게임의 개념 및 방식을 탈

피하여 최적 미로를 찾아 입력하고 사용자가 선택한 미로가 최적 경로에 대해 어느 정도 가까운가를 평가하여 지형적 인지도의 수준을 나타내주는 인지력 학습에 도움이 되는 최적 미로 게임이다.

II. 소프트웨어 구성

본 논문의 최적 미로 게임은 Java 언어로 작성되었으며 JDK1.3 컴파일러로 컴파일 되었다[1], [2]. 그림 1에서와 같이 소프트웨어 구성은 GUI 사용자 인터페이스기, 랜덤 경로 발생기, 최적 경로 탐색기, 사용자 선택 경로 입력기, 및 인지력 측정기로 크게 5개의 부분으로 되어 있다.

사용자 인터페이스기(GUI module)는 전체 화면의 메뉴 구성 및 게임 정보를 사용자에게 나타내 주며 마우스를 이용한 사용자의 경로 선택을 가능하게 하고 최종 경로 입력을 받아들인다[3].

랜덤 경로 발생기(random path generation module)는 게임 패널(game panel) 내에 고유 패턴을 가진 임의의 블록 장애물을 발생시키고 좌우 경계선 쪽에 출발지(source location)와 목적지(destination location)를 임의의 위치에 발생시킨다. 이때 출발지에서 목적지까지의 도달 경로가 존재하지 않는 경우를 방지하기 위해 경계 및 대각선 상의 장애물 블록을 제거한다.

최적 경로 탐색기(optimal path search module)는 랜덤 경로 발생기에서 발생된 현재의 미로 내에서 maze routing 알고리즘을 이용하여 출발지에서 목적지까지의 최적 경로를 탐색하고 배열 자료구조 형태로 경로의 시작 선분부터 끝 선분까지 저장한다.

선택 경로 입력기(selected path input module)는 사용자가 게임 화면에 제시된 미로 내에서 마우스를 이용하여 각각의 선분을 입력할 수 있게 해준다. 출발지를 기점으로 가로 및 세로 선분을 차례대로 입력 하도록 하며 대각선 입력은 받아들이지 않는다. 이때 직전 입력 선분을 재입력할 수 있도록 하여 사용자의 입력 선분 수정을 가능하도록 한다.

인지력 측정 환산기(cognition faculty test module)는 사용자에게 의해 선택 및 입력된 출발지에서 목적지까지의 전체 경로와 최적 경로 탐색기(optimal path search module)에 의해 선택된 전체 경로를 비교 및 분석하여 사용자의 인지력 레벨을 수치로 계산하고 GUI를 통하여 게임의 결과 정보를 화면에 나타내어 준다. 이때 최적 경로의 값을 1.0으로 볼 때 사용자의 입력 경로가 얼마나 최적 경로에 가까운지를 전체 인지력 수치로 나타내어준다. 또한 전체 인지력을 지형적 인지력, 논리적 인지력, 및 수리적 인지력으로 나누어 상세한 정보를 나타내어 준다.

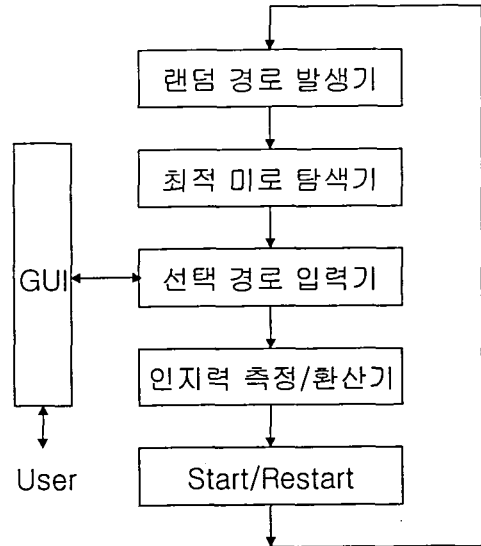


그림 1. 소프트웨어 구성

III. 사용자 인터페이스

사용자 인터페이스(GUI)는 클라이언트 컴퓨터(client computer) 시스템의 형태에 관계없이 소프트웨어를 수행할 수 있는 Java의 장점을 그대로 이용하고 해당 클라이언트 시스템의 웹브라우저(Web Browser)와 웹브라우저에 플러그인(plug-in)되는 자바 가상 머신(java virtual machine)을 탑재한 소프트웨어에서 언제라도 수행할 수 있도록 개발되었다. 그림 2에서와 같이 전체 화면 및 메뉴 구성은 게임을 할 수 있는 게임 패널 영역, 인지력 테스트 결과 및 정보 디스플레이 영역, 시작 버튼 영역, 그리고 출발지 및 도착지에 대한 간단한 설명과 소개 영역으로 구성되어 있다. 그림 3에 실제 게임 화면의 메뉴 구성을 나타내었다. 게임 방법은 게임 시작 버튼을 이용하여 시작 화면으로 들어가면 새로운 미로가 발생되고 좌측 출발지에서부터 사용자가 마우스를 이용하여 경로를 선택할 수 있다. 이때 내부 장애물 블록은 경로로 선택할 수 없기 때문에 장애물 블록을 피해가면서 목적지까지 가장 짧은 거리이면서 가장 간단한 경로를 선택하는 사용자가 최적 경로에 가까운 결과를 얻게 되고 사용자의 입력 경로가 최적 경로와 동일하다면 인지력 1.0의 결과를 얻게 된다.

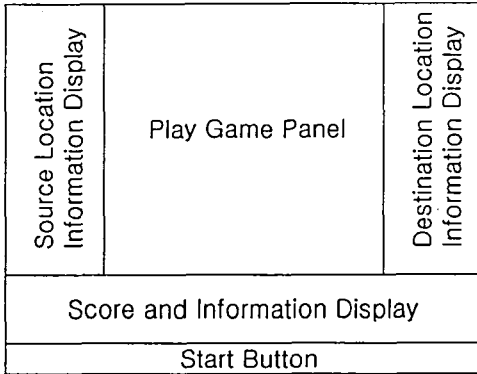


그림 2. 메뉴 구성

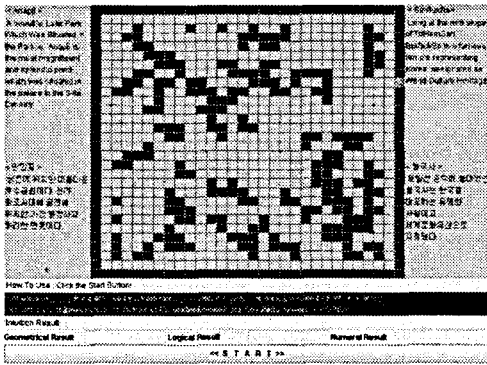


그림 3. 게임 화면의 메뉴 구성

IV. 자료구조 및 알고리즘

본 논문에서 사용한 자료 구조는 그리드베이스 라우팅(grid base routing)을 위한 2차원 정수 배열을 사용하였다. 2차원 정수 배열 상에서 최적 경로 탐색에 사용한 알고리즘은 lee's algorithm으로 알려진 maze-route algorithm을 사용하였다[4]. 게임을 위한 그리드 배열의 크기가 크지 않고 최단 거리의 경로를 간단하게 찾아 주는 lee's algorithm의 가상코드(pseudo-code)를 그림 4에 나타내었다. 여기서 B는 입력 배열, s는 출발지, t는 목적지, L[v]는 출발지 s로부터 노드 v까지의 거리, 그리고 P는 탐색후의 최단 거리 경로를 나타낸다. 출발지에서부터 목적지까지의 거리를 그림 4의 가상코드와 같이 모든 이웃 노드에 전파 기록한 후 역추적(retrace)하였다. 이때 최단 거리 경로를 찾음과 동시에 가장 간단한 경로를 최종 선택하기 위해 역 추적 시 현재 추적방향을 높은 우선 순위 방향 (high priority direction)으로 선택하였다. 그림 5에 역추적 가능 경로의 예를 나

타내었다.

```

Algorithm Lee-Router(B, s, t, P)
input: B, s, t
output: P
begin
  plist = s;
  nlist = 0;
  temp = 1;
  path_exists = False;
  while plist != 0 do
    for each vertex vi in plist do
      for each vertex vj neighboring vi do
        if B[vj] = Unblocked then
          L[vj] = temp;
          insert(vj,nlist);
          if vj = t then
            path_exist = True;
            exit while;
          temp = temp + 1;
          plist = nlist;
          nlist = 0;
        if path_exists = True then retrace(L, P);
        else path does not exist;
      end
    end
  end
  
```

그림 4. lee-router 가상코드

7	8	9	10	11	12		
6	7	X	9	10	11	X	
5	6	X	8	X	12	X	12
4	X	X	7	X	11	10	11
3	4	5	6	X	X	9	X
2	X	6	5	X	7	8	9
1	X	X	4	5	6	7	8
S	1	2	3	4	5	6	X
1	2	3	X	X	X	7	8
2	X	4	5	6	7	8	X
3	X	X	6	7	8	9	10

그림 5. 역추적 가능 경로

사용자가 선택한 경로가 최적경로에 얼마나 가까운지 그 경로의 수준을 계산하여 지수로 나타내는 방법은 최적 경로와 선택경로에서 총 경로의 면적비를 지형적 인지력, 전체 길이의 합의 비를 수리적 인지력, 선택의 수의 비를 논리적 인지력으로 계산하여 합산한 결과를 총 인지력 지수로 계산한다. 따라서 총인지력을 지수는 [총인지력지수(total cognition index) = 지형적인지력(geometrical cognition) + 수리적인지력(numeral cognition) + 논리적인지력(logical cognition)]으로 나타낸다. 그림 6에 경로 입력 후의 인지력 정보

와 경로를 나타내는 게임의 종료화면을 나타내었다.

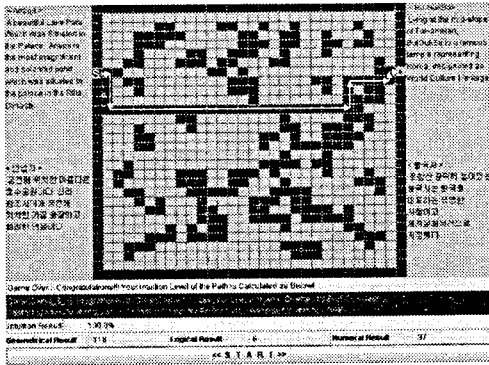


그림 6. 경로 입력 및 게임 종료 화면

V. 결 론

본 논문에서는 웹 기반 게임 콘텐츠로서의 최적 미로 게임을 개발 및 구현하였다. 최적 경로 탐색 알고리즘을 기반으로 하여 사용자의 선택 경로가 최적 경로에 근접하는 정도를 이용하여 사용자의 미로 인지력을 평가할 수 있도록 하였으며 사용자 인지력을 지형적 인지력, 수리적 인지력 및 논리적 인지력으로 세분화하여 평가할 수 있도록 하였다. 그로인해 반복적인 사용자의 학습을 통한 인지력 향상을 기대할 수 있는 소프트웨어로써 게임의 재미와 인지력 훈련 및 학습 효과를 함께 얻을 수 있는 웹 기반 학습 및 게임 콘텐츠 개발에 응용 가능하다.

참고문헌

- [1] John Lewis and William Loftus, Java Software Solutions, Addison Wesley Longman, Inc., 2001.
- [2] Y. Daniel Liang, Introduction to JAVA Programming, Prentice Hall, 2002.
- [3] J.D. Foley and A. Van Dam, Fundamentals of Interactive Computer Graphics, Addison Wesley, 1984.
- [4] Naveed A. Sherwani, Algorithms for VLSI physical design automation, Kluwer Academic Publishers, 1993.