

2019 졸업과제 중간보고서

Graph상 Cops and Robber 게임

A분과

팀명: 오뚜기

201624563 임학천

201624501 송태영

201324456 배선우

지도교수: 조환규

목차

1. Cops and Robber 게임 이론
2. 목표
3. 요구조건 분석
4. Cops and Robber 게임 변형
 - 4-1. 여러 명의 Cops and Robber
 - 4-2. 시야 제한
 - 4-3. 이동 제어
5. 웹 기반 Cops and Robber 게임 개발
 - 5-1. 전체
 - 5-2. 클라이언트
 - 5-3. 서버
6. Human-AI COM 개발
7. 중간 결과
 - 7-1. 클라이언트
 - 7-2. 서버
 - 7-3. 통신
8. 추진체계 및 일정
9. 구성원 역할 분담

1. Cops and Robber Game Theory

Cops and Robber 게임 이론에는 수 많은 변형 버전이 있지만 가장 기초가 되는 방식은 다음과 같다. 각 Cop들과 Robber들은 그래프 중 하나의 노드에 위치하고 있다. Cop들과 Robber들은 턴을 번갈아 가며 각 턴마다 자신의 위치를 고수하거나, 자신에 인접한 노드로 움직일 수 있다. 만약 Cop의 위치가 Robber의 위치와 일치한다면 해당 Robber는 게임에서 제거되게 된다.

통상적으로 해당 게임에서 해결하고자 하는 문제는 그래프 상의 모든 Robber를 제거하기 위해 총 몇 명의 Cop가 필요한지 구하는 것이다. 만약 1명의 Cop만으로 충분하다면, 해당 그래프는 *cop-win-graph*로 정의된다. 이 경우 n 개의 노드를 가진 그래프 상에서 1명의 Robber가 1명의 Cop에 의해 n 의 선형 시간 내에 항상 제거됨을 의미한다. r 명의 Robber가 k 명의 Cop에 의해 m 에 비례하는 선형 시간 내에 항상 제거되는 상황 역시 *cop-win-graph*로 정의되지만, 이에 소요되는 정확한 시간에 대한 한계점은 아직 알려지지 않았다.

2. 목표

해당 프로젝트는 Cops and Robber Game Theory에 소개된 게임을 실제 컴퓨터 게임을 통해 구현하고자 한다. 온라인을 통해 AI 또는 여러 명의 사용자와 게임 진행이 가능하며, 1개의 Cop이 1개의 Robber를 잡으러 다니는 아무런 제약 조건이 없는 가장 클래식한 게임 방식부터, 여러 개의 Cop이 여러 개의 Robber를 잡으러 다니거나, 시야 제한 / 움직임 제한 등의 각종 방해 요소가 존재하는 변형 방식까지 구현하는 것에 목표를 두고 있다.

3. 요구조건 분석

- 1) 각 플레이어는 각 턴을 돌아가며 Cops & Robbers 게임을 플레이 한다.
- 2) 턴은 플레이어가 아닌 Agent (Cop, Robber)를 중심으로 번갈아 가며 진행된다.
- 3) 1명의 Robber 플레이어로 Cop AI와 게임이 가능해야 한다.
- 4) 1명 이상의 Cop 플레이어가 1명 이상의 Robber 플레이어와 온라인상에서 게임이 가능해야 한다.
- 5) 1개 이상의 Cop과 Robber가 존재할 수 있다.
- 6) 각 플레이어는 게임 플레이 도중 각종 이익이 되는 요소 또는 불이익이 되는 요소에 의해 영향을 받을 수 있다.
- 7) Cop 플레이어와 Robber 플레이어는 각자의 승리 조건을 가지고 있다.

8) Cop AI는 주어진 상황에서 최대한 승리를 쟁취하기 위해 노력해야 한다.

4. Cops and Robbers 게임 변형

4-1. 여러 명의 Cops and Robber

Cops and Robber의 가장 기초가 되는 방식은 1명의 Cop과 1명의 Robber에 대해 게임을 진행하는 것이다. 이 경우 주어진 그래프가 cop-win-graph인지 robber-win-graph인지 판단이 가능한 경우가 대부분이지만, Cop과 Robber들이 각각 2명 이상 존재할 경우에 대해서는 아직 알려지지 않은 경우가 대부분이다. 해당 프로젝트에서는 온라인을 통한 사람 대 사람 대결에서 여러 명의 Cop과 Robber을 가지고 게임을 진행해볼 수 있도록 한다.

4-2. 시야 제한

Cops and Robber의 변형 버전에서 가장 자주 언급되는 변형법인 시야 제한을 구현하고자 한다. 각 Cop과 Robber는 자신과 인접한 k 개의 노드에 대해서만 시야를 가지고 있으며, 시야가 없는 노드의 경우 해당 노드에 누가 위치하고 있는지에 대한 정보를 알 수 없다. 이는 현실 세계에서 실제 인접한 장소에 대한 정보를 획득할 수 있는 상황과 부합한다.

4-3. 이동 제어

또 다른 변형 중의 하나는 Cop과 Robber가 한 턴에 움직일 수 있는 노드의 거리에 변형을 두는 것이다. 그래프 상의 특정 노드에 아이템이 생성되고, 해당 아이템을 획득하면 해당 플레이어는 한 번에 2개의 노드를 건너 뛸 수 있게 된다. 이는 현실 세계에서 교통 수단 등을 통해 빠르게 이동하는 상황과 부합한다.

5. 웹 기반 Cops and Robber 게임 개발

5-1. 전체

기본적으로 Cops & Robbers 게임은 두 가지 게임 모드를 가지고 있다.

- 1) 사용자가 Robber가 되어 Cop AI 플레이어와 대결
- 2) 온라인을 통해 여러 명의 사용자가 Cop 또는 Robber가 되어 대결

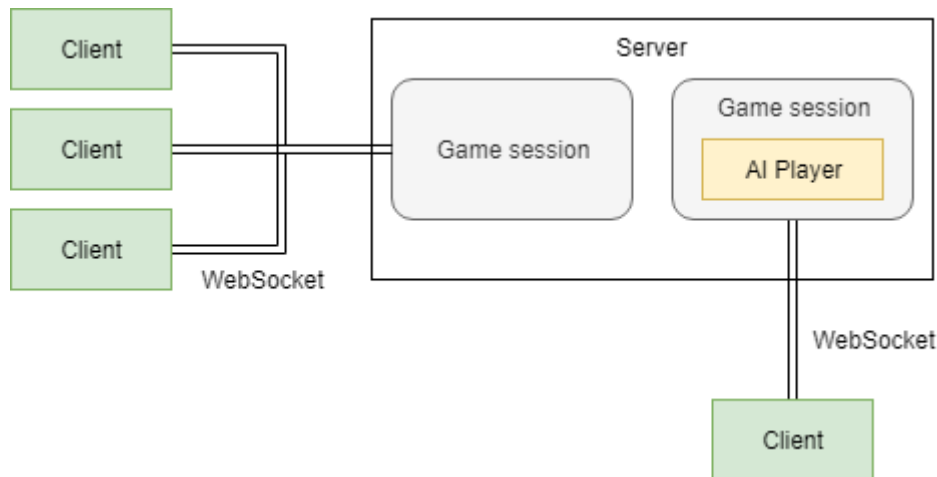
게임이 온라인 상에서 여러 명의 플레이어와 진행될 수 있어야 하므로, 소켓 통신을 지원하는 서버를 구현할 필요가 있다. 따라서 서버와 클라이언트 단을 분리하여 개발하도록 한다. 또한 사용자가 AI 플레이어와 대결하는 경우에도 AI에 대한 연산은 서버에서 이루어지도록 한다.

통신을 위해 사용할 프로토콜의 경우 해당 게임이 Turn-based 게임임을 고려하였을 때 메시지의 송수신 여부가 보장되며 (Acknowledgement control) 메시지의 순서가 보장되는 (In-order delivery) TCP 프로토콜을 사용하도록 하였다.

클라이언트 개발 환경으로는 Web platform을 선택하였다. 해당 게임이 그렇게 화려한 그래픽을 요구하지 않기 때문에, HTML&CSS 만으로도 충분히 GUI에 대한 구현이 가능하다고 판단하였고, 타 언어 / 플랫폼과의 생산성을 고려하였을 때 웹이 월등한 우위를 가지고 있다. 또한 사용자들이 사용하는 플랫폼에 관계없이 웹 브라우저를 통해 간단히 게임에 접근할 수 있다는 장점이 존재한다.

서버 개발 환경으로는 C/C++을 선택하였다. 사용자가 AI 플레이어와 대결하는 경우 AI 플레이어에 대한 연산이 웹 서버에서 이루어지므로, AI에 대한 연산이 복잡해질수록 연산 시간에 대한 성능문제가 발생할 수 있다. 이에 동시성 또는 생산성보다 실행 성능을 중점에 두었다.

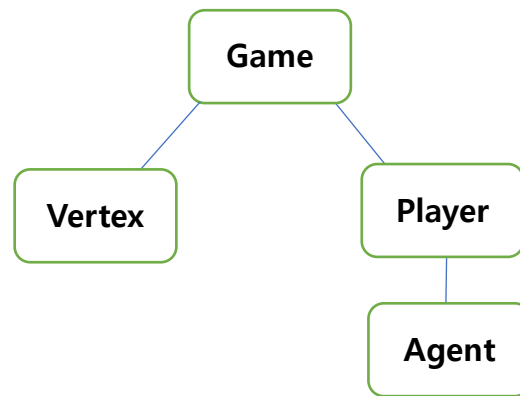
현재 웹 플랫폼 상에서는 TCP/IP 표준을 바탕으로 한 WebSocket 프로토콜을 지원하고 있다. 따라서 클라이언트와 서버는 WebSocket 프로토콜을 통해 통신하도록 한다.



<그림> Client – Server 구성도

5-2. 클라이언트

다양한 OS에서 실행이 가능하고 업데이트가 간편하다는 장점 때문에 전체적인 시스템은 웹을 이용해 만들었다.



- 1) **Game** – agent를 관리하는 player 와 그래프를 만드는 vertex 객체를 포함하고 관리한다.
- 2) **Vertex** – 그래프를 만드는 역할을 한다. 각 vertex마다 id가 있으며 x, y 좌표를 받아 vertex를 생성하고 그 vertex들을 잇는 Edge를 생성한다.
- 3) **Player** – player는 agent들을 조종하는 역할을 한다.
- 4) **Agent** – 경찰과 도둑이 되는 객체이다. 원하는 vertex에 초기 agent의 위치를 설정하고 원하는 vertex를 클릭하면 움직일 수 있다.

5-3. 서버

맵 생성 - 게임 시작 요청을 받았을 때 맵을 생성한다. 맵은 랜덤한 숫자의 가로x세로 그리드(grid)형식이다. 그 중 몇 개의 Vertex가 뚫려 있고, 인접한 Vertex사이에 Edge가 있다. Vertex는 경찰 또는 도둑이 위치하는 곳이고 Edge는 인접한 Vertex로 이동할 수 있는 통로이다.

게임 진행 - 게임이 시작되면 게임을 진행한다. 각 플레이어별로 경찰/도둑, 말 개수, 말 순서를 정하고 순서에 따라 이동요청을 받거나(유저) 임의로(COM) 이동시킨다. 경찰이 도둑과 같은 위치로 이동하게 되면 도둑이 잡히는 것으로 간주하고 경찰이 도둑을 못 잡거나 모든 도둑이 잡힌 경우 승패를 가리고 게임을 종료시킨다.

COM - 유저가 COM와 플레이를 선택하면 COM의 차례일 때 말을 이동시킨다. 주변 상황을 고려하여 COM이 승리할 확률이 높은 쪽으로 이동시킨다.

6. Human-AI COM 개발

COM 구현 시 논문에서 고안된 Cop-win Strategy(경찰이 도둑을 잡기위한 최선의 전략), Robber-win Strategy를 분석하고 자체 Cop-win Strategy, Robber-win Strategy를 고안, 개발하여 인공지능 식의 스스로 승률이 높은 쪽을 찾아 나가는 Human-AI COM을 개발한다.

7. 중간 결과

7-1. 클라이언트

서버에서 맵 정보를 받으면 정보에 따라 그리드를 그리고 이동요청을 받으면 요청된 위치로 말을 이동시킨다.

7-2. 서버

클라이언트에서 게임시작 요청을 받으면 랜덤한 숫자의 가로x세로 그리드를 만들고 무작위로 몇 개의 Vertex를 뚫어 맵을 만든다. 만들어진 맵 정보를 클라이언트로 보낸다

7-3. 통신

WebSocket을 이용하여 메시지 형식으로 클라이언트와 서버 간에 정보를 주고받는다.

8. 추진체계 및 일정

| 월 | 6 | | 7 | | | | | 8 | | | | 9 | | | |
|------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 주 | 3 | 4 | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 개발방향 및 일정확인 | | | | | | | | | | | | | | | |
| 전체 설계 | | | | | | | | | | | | | | | |
| 클라이언트 구현 | | | | | | | | | | | | | | | |
| 서버구현 | | | | | | | | | | | | | | | |
| 전략 알고리 즘 분석개발 | | | | | | | | | | | | | | | |
| 수정 및 보완 | | | | | | | | | | | | | | | |
| 최종발표 준비 | | | | | | | | | | | | | | | |

9. 구성원 진척도

| | | |
|-----|----------------|----|
| 임학천 | 게임서버 구현 | 진행 |
| | COM 구현 | 예정 |
| | 전략 알고리즘 분석, 개발 | 예정 |
| 송태영 | 웹 구현 | 진행 |
| | 전략 알고리즘 분석, 개발 | 예정 |
| 배선우 | 클라이언트-서버 통신 구현 | 완료 |
| | GUI 애니메이션 구현 | 진행 |
| | 전략 알고리즘 분석, 개발 | 예정 |