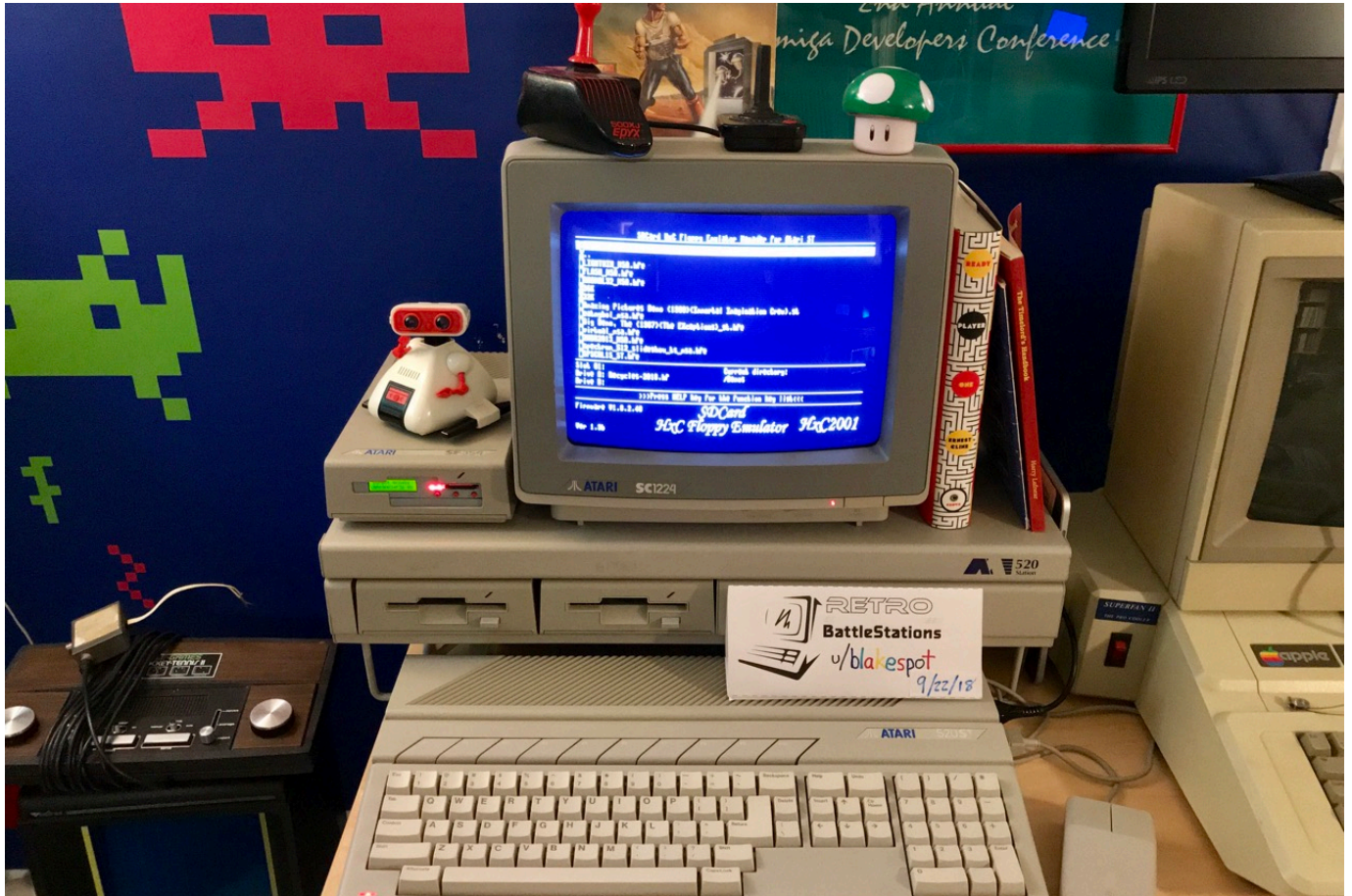


조별과제 계획서 : 숫자야구

Proposal of Team Task : Bulls and Cows



19-1 객체지향프로그래밍

Team 2 최한규 팀

49003157 단일계열 최우석

20180476 창의IT융합공학과 박한빛

20180038 무은재학부 박형규

Introduction

숫자야구는 몇 가지 옵션을 더해 다양한 방식으로 즐길 수 있다. 플레이어가 숫자열을 제시하는 횟수에 제한을 두어 출제자와 플레이어 사이의 대결구도를 만들 수도 있고, 한 명의 출제자와 두 명의 플레이어를 뒤 두 유저 간의 대결구도를 만들 수도 있다. 여기에서 더 나아가 한 플레이어가 숫자열을 제시했을 때, 해당 숫자열과 그로부터 나온 스트라이크와 볼의 수를 다른 플레이어도 알 수 있게끔 하는 한 편, 두 플레이어의 숫자열 제시를 번갈아가면서 하게끔 turn-based game 형식의 제한을 둬으로써 심리전을 유발하는 게임을 진행할 수도 있다. 이렇듯 다양한 바리에이션이 존재하는 숫자야구 게임을 프로그래밍 언어, 특히 코드의 re-usable을 쉽게 이룰 수 있는 C++로 구현하면 흥미로운 프로젝트가 될 것이라고 생각했다.

또한 이러한 두 플레이어간의 대결구도에서, 상대 플레이어를 대신하는 AI를 구현해보는 것도 흥미로운 도전이 될 것 같았다. 본격적으로 전공에 뛰어들기에 앞서, 인공지능의 정의와 개념에 대해 알아보고 미숙하게나마 ‘AI’라고 부를 수 있을 만한 상대 플레이어를 이번 학기에 배운 C++을 통해 구현해보고 싶었다.

Rule of Game

✓ 게임 시작

- 문제를 제시하는 '출제자'와 문제를 푸는 '플레이어'가 존재한다.
- 출제자는 0에서 9까지의 서로 다른 숫자로 이루어진 숫자열을 하나 생각한다. 플레이어는 이 숫자열을 정확하게 맞춰야 한다.
- 초기에 플레이어는 숫자열에 들어간 숫자의 수만 제공받는다.

✓ 게임 진행

- 플레이어가 예상하는 숫자열의 각 숫자를 차례로 Z_i , 정답 숫자열의 각 숫자를 차례로 S_i 라고 하자. (i 의 최댓값은 정답 숫자열 내 숫자의 갯수)
- 플레이어는 출제자에게 예상하는 숫자열을 제시한다.
- $Z_i = S_i$ 를 만족하는 하나의 쌍은 '스트라이크', $i < j$ 인 $Z_i = S_j$ 를 만족하는 하나의 쌍은 '볼'이라 칭한다.
- 출제자는 플레이어에게 '스트라이크'와 '볼'의 수를 제공한다. 만약 둘 다 0이라면 '아웃'을 제공한다.
- 모든 i 에 대해 $Z_i = S_i$ 를 만족할 때까지 위 과정을 반복한다.

✓ 게임 종료

- 모든 i 에 대해 $Z_i = S_i$ 를 만족하면 게임을 종료한다

Conceptual Design

✓ Program Flow

- 초기 메뉴에서 싱글플레이모드, AI 대전모드, PVP 모드 중 선택
- 각 모드별로 세 가지의 난이도 존재, 난이도는 정답숫자열의 수 (3자리, 4자리, 5자리)
- 게임은 9회로 제한, 9회까지 승부가 나지 않는다면 무승부처리
- 한 게임이 종료되면 다시 초기메뉴 출력
- 프로그램 종료시 진행했던 게임의 과정과 결과 로그파일로 저장

✓ Class Design

- Manager ; 프로그램의 전체적인 구동 관리
 - data : 게임 진행 횟수, 현재까지의 전적, 난이도 및 메뉴선택 flag
 - method : 메뉴 출력, 난이도 선택, 이하 모든 클래스 관리
- Referee ; 출제자
 - data : 한 게임 내 예상숫자열 제시횟수, 정답 숫자열
 - method : 스트라이크 및 볼 계산
- AI ;
- Log ; 진행된 게임의 로그파일 출력
 - data : fstream객체와 file path
 - method : 한 게임이 종료되면 해당 게임의 진행을 .txt 파일에 기록하여 저장

✓ AI Algorithm

<가장 구현이 확실한 방법>

정답이 세 자리 숫자열인 경우, '123' ~ '987' 까지의 제시 가능한 숫자열을 모두 배열 S에 저장.

정답일 가능성이 있는 S 내의 숫자열 하나를 랜덤으로 제출(초기에는 모든 숫자열이 가능성을 갖고 있음). 이를 d라고 하자. 이후 제공받은 '스트라이크', '볼', 혹은 '아웃' 정보를 k라고 한다면, S의 각 숫자열과 d를 비교하여 k를 받을 수 없는 숫자열을 모두 삭제.

예를 들어, 만약 '원 스트라이크'를 제공받았다면, S의 각 숫자열과 d를 비교하여 '원 스트라이크'가 나올 수 있는 숫자열을 제외하고 모두 삭제.

만약 '아웃'을 제공받았다면, S에서 d 숫자열에 포함된 세 숫자 중 하나라도 포함된 숫자열을 모두 삭제.

<불확실한 방법>

정답이 세 자리 숫자열인 경우, 스트라이크와 볼의 수를 (s,b) 의 쌍이라고 하였을 때, 받을 수 있는 스트라이크와 볼의 쌍은 (0,0), (0,1), (0,2), (0,3), (1,0), (1,1) ... (2,0) 이다. 위의 알고리즘에서 가능성이 있는 숫자열을 랜덤으로 제출하는 대신, 위 스트라이크와 볼의 쌍의 경우를 세어서 정답일 수 있는 숫자 갯수의 최댓값을 최솟값이 되를 방향으로 숫자열을 제출.

Project Management

✓ Distributing Roles

- 최우석 : full-stack of game flow
- 박한빛 : AI design and implement
- 박형규 : AI design and Log structure design

✓ Using Github for Version control of Programming

✓ 5.12(일), 5.19(일)에 서로 대면하여 진행결과 나누기 및 회의