

2016 정보과학II 프로젝트 최종 보고서

학년	1	학반	6	번호	10	이름	오경준
----	---	----	---	----	----	----	-----

작품명	절대 죽지 않는 알고리즘
-----	---------------

1. 프로그램 개요

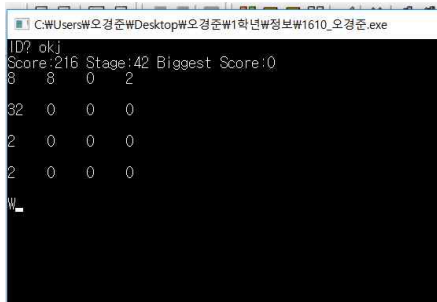
1) 구현된 기능 소개

아이디를 입력하고 실행하는 2048 게임, 찬스 추가, 인공지능 추가, 순위 정렬 추가

2) 프로그램 실행 흐름도

거의 렉 걸리는 일 없이 잘 진행된다.

2. 프로그램 사용법 및 주요 실행 결과(이미지 첨부)

사용법	시연(혹은 실행) 이미지
<p>게임이 실행되면 아이디 입력 W(위쪽), A(왼쪽), S(아래쪽), D(오른쪽)으로 밀기 C를 통해 달팽이 배열로 찬스 U, R을 통해 실행취소, 원상복귀 P를 통해 인공지능 실행 N 새로운 게임, X 게임 종료</p>	

3. 주요 알고리즘 소개

밀어서 더하는 함수, 찬스 함수, 실행 취소 함수, 아이디 정렬 함수, 인공지능 함수 등

4. 배운 점

게임을 만드는 일은 매우 어렵고 그 게임의 인공지능을 만드는 일은 더더욱 어렵다.

5. 아쉬운 점, 느낀 점

인공지능을 구현하기는 했는데 더하는 함수에서 오류가 있었는지 계속해서 숫자가 사라져서 아쉽다.

6. 참고 프로그램 및 문헌(혹은 사이트 주소)

없습니다.

7. 개발일지

여기에 지금까지 작성했던 개발일지 모두 복사하여 붙여 넣기

8. 첨부파일 : 소스파일(꼼꼼한 주석 처리 요함)

※ 주의

- 해당사항 기입시 위 형식(글꼴 변경 금지)으로 작성하며 그 외 별도의 제한은 없습니다.

- 필요 · 충분 조건을 만족하도록 요약 정리하여 작성합니다.

2016 정보과학II 프로젝트 개발 일지

학년	1	학반	6	번호	10	이름	오경준
----	---	----	---	----	----	----	-----

작품명	2048 게임 1단계
-----	-------------

1단계 9 월 18 일~ 9 월 18 일	
주요 활동	<ul style="list-style-type: none"> · N을 입력했을 때 게임이 새로 시작되게 만들었다. · X를 입력했을 때 게임이 종료되게 만들었다. · 2가 나올 확률을 80%, 4가 나올 확률을 20%가 되게 만들었다.
구체적 활동 기록	<p>· 2가 나올 확률을 80%, 4가 나올 확률을 20%가 되게 하기 위해서 랜덤함수를 통해 랜덤으로 나온 수를 5로 나눈 나머지를 보았을 때 4가 나오면 4가 나오고 4가 아닌 수가 나오면 2가 나오게 해서 확률을 조정했다.</p> <p>· 지난번 프로젝트 때는 '새 게임' 이라는 개념이 없어서 firstrandom 함수에서 굳이 모든 배열을 0으로 초기화 시킬 필요가 없었다. 그런데 이번 프로젝트에 새 게임 N이 생기면서 기존의 프로젝트에 spell이 N일 때 firstrandom 함수를 실행한다고 하니까 arr가 0으로 초기화가 되지 않아서 계속 기존의 상태에서 임의의 자리에 숫자가 추가되었다. 그래서 firstrandom 함수에 모든 배열을 0으로 초기화 시키는 내용을 추가하니까 그 문제점이 해결되었다.</p> <p>· 원래는 움직일 수 있는 방향이 있으면 while 문을 돌다가 더 이상 움직일 수 없는 상태가 되면 while 문을 빠져나와 GAME OVER!!를 출력했다. 이번 프로젝트에서 X가 추가되면서 spell이 X이면 바로 while 문을 빠져나오게 해서 바로 GAME OVER!!가 출력되게 했다.</p>
연구 반성 및 다음 주차 활동 계획	<ul style="list-style-type: none"> · 현재 나의 코드는 뒤로 가면 갈수록 심하게 느려진다는 문제점이 있다. · 다음 주차에서는 다른 친구의 방법을 배워서 새롭게 코드를 다시 짜서 속도를 빠르게 해야 할 것 같다.
배운 점	<ul style="list-style-type: none"> · 랜덤함수로 확률도 조절할 수 있다는 사실 · 나의 코드는 느려서 발전이 필요하다는 사실

※ 주의 사항 : 기재할 항목 변경은 가능하며 주당 최대 2쪽이내 작성

2016 정보과학II 프로젝트 개발 일지

학년	1	학반	6	번호	10	이름	오경준
----	---	----	---	----	----	----	-----

작품명	2048 2단계
-----	----------

2 단계	10월 9일~ 10월 12일
주요 활동	<ul style="list-style-type: none"> · 아이디 입력 기능이 추가되었고 해당 아이디의 점수, 단계를 저장하고 출력 · 최종적으로 5위까지 내림차순으로 출력 · 방향을 입력하면 바로 바뀌게 하였고 속도가 매우 빨라짐
구체적 활동 기록	<p>· random이라는 함수가 랜덤으로 i, j를 고르고 arr[i][j]가 0이 아니면 그곳에 2 또는 4를 넣는 방식을 택했다. 하지만 이런 방식으로 2 또는 4를 넣는 것이 2048 게임을 지나치게 느린다는 것을 알게 되었고 랜덤으로 i, j를 고르는 것이 아니라 먼저 0이 아닌 곳의 위치를 파악하고 그 안에서 랜덤으로 2 또는 4를 넣는 방법을 택했다. 이렇게 코드를 바꾸니 기존의 코드에 비해 훨씬 빨라졌다.</p> <p>· 처음에는 받는 모든 이름을 2차원 배열에 저장하고 모든 점수와 단계를 일차원 배열에 저장하고 마지막에 버블 소트를 통해 정리하려고 했다. 하지만 어차피 5위까지만 출력하면 되는데 이렇게 되면 메모리 낭비가 너무 심할 것 같아서 다른 방법을 선택했다. N을 누를 때마다 sortmine이라는 함수에 가서 지금까지 있는 데이터 중에서 자기 점수가 몇등인지 순위를 찾아가서 끼우는 것이다. 이렇게 되면 재귀적으로 5위까지만 저장이 된다.</p> <p>· 원래는 scanf를 써서 입력할 때마다 엔터를 쳐야했다. 그런데 그렇게 하니까 단계가 한번에 두 개씩 올라가는 문제점이 생겼다. 입력을 받은 뒤에 1을 빼줘도 되는데 더욱 더 2048에 비슷한 게임을 만들기 위해서 getch로 입력을 받았다.</p>
연구 반성 및 다음 주차 활동 계획	<ul style="list-style-type: none"> · 다른 친구의 random 함수를 참고했다. · 3단계 코드는 쉽다고 들어서 이제 슬슬 인공지능에 대해 생각해봐야 할 것 같다.
배운 점	<ul style="list-style-type: none"> · 지나치게 랜덤으로 하면 코드가 느려진다. · 간단한 방법으로 할 수 있다면 조금 더 편하더라도 그렇게 하는게 더 도움이 된다.

※ 주의 사항 : 기재할 항목 변경은 가능하며 주당 최대 2쪽이내 작성

2016 정보과학Ⅱ 프로젝트 개발 일지

학년	1	학반	6	번호	10	이름	오경준
----	---	----	---	----	----	----	-----

작품명	2048 프로젝트 3단계
-----	---------------

3 단계 10 월 23 일 ~ 10 월 23 일	
주요 활동	<ul style="list-style-type: none"> · C 또는 c를 입력했을 때 달팽이 배열의 형태로 재배치했다. · 찬스 기능은 3번만 사용가능하도록 했다.
구체적 활동 기록	<ul style="list-style-type: none"> · 기존의 arr[4][4] 배열 말고 arr2[4][4]라는 배열을 새로 만들어 C 또는 c가 눌릴 때마다 snailchance라는 함수에 들어가게 했다. snailchance라는 함수는 달팽이 배열의 형태로 arr[4][4]의 값을 arr2[4][4]에 저장하고 그것을 다시 arr[4][4]에 저장하는 함수이다. · 찬스 기능이 세 번만 사용 가능하므로 c_num이라는 전역 변수를 새롭게 만들었다. c_num은 snailchance 함수 안에 들어갈 때마다 1 씩 증가한다. 3일 때부터는 snailchance 함수에 들어가지면 안 되므로 if(c_num!=3)이라는 코드를 넣어 3번까지만 실행되도록 했다. · 이 코드에서는 한 사람 말고도 다른 사람이 계속 아이디를 입력하면서 사용한다. 하지만 c_num은 전역변수이기 때문에 다른 사람들이 c_num 이 3이 된 상태로 사용할 수가 있게 된다. 따라서 N 또는 n을 누를 때 c_num을 다시 0으로 초기화시켜 준다.
연구 반성 및 다음 주차 활동 계획	<ul style="list-style-type: none"> · 어떻게 하면 달팽이 배열을 간단하게 할 수 있을까 고민을 해봤지만 시간이 없는 관계로 노가다로 끝낸 점을 반성해야 할 것 같다. · 다음 주차는 원래 쓰던 checkarr 배열을 쓰면 쉽게 해결할 수 있을 것 같다.
배운 점	이번 주차는 쉽게 해결해서 딱히 없었던 것 같다.

※ 주의 사항 : 기재할 항목 변경은 가능하며 주당 최대 2쪽이내 작성

2016 정보과학II 프로젝트 개발 일지

학년	1	학반	6	번호	10	이름	오경준
----	---	----	---	----	----	----	-----

작품명	2048 게임 4단계
-----	-------------

4단계	11 월 4 일~ 11 월 5 일
주요 활동	<ul style="list-style-type: none"> · U를 입력했을 때, 게임의 전체적인 상황(점수, 스테이지, 배열)을 모두 이동하기 전으로 되돌렸다. · R을 입력했을 때 U를 입력하기 전의 상황으로 되돌렸다.
구체적 활동 기록	<p>· U를 통해서 한 번에 최대 3칸 전까지 갈 수 있도록 만들기 위해서는 1, 2, 3 단계 전의 배열을 저장하는 새로운 배열이 필요했다. 따라서 undoarr1, 2, 3을 만들었고 readyundo라는 함수에서 재귀적으로 1~3단계 전의 배열을 저장했다. switch문에서 case를 나눠서 case가 1일 때는 undoarr1을 가져오는 방식으로 했다. undo_num이라는 변수를 통해서 한 사람 당 3번 밖에 사용하지 못하게 했다. stage의 경우 한 단계 전으로 가면 1 감소하기만 하면 돼서 stage—로 해결했지만 score의 경우 그렇지 않아서 score1, score2, score3이라는 변수를 만들어서 재귀적으로 저장했다.</p> <p>· R을 통해서 다시 원래대로 돌아가게 하기 위해서 undoarr를 쓰면 될 것 같았지만 쉽게 되지 않아서 redoarr라는 변수를 새롭게 만들었다. undo를 한 번 썼을 때는 redoarr1을 불러오고 두 번 썼을 때는 redoarr2가 나오고 그 다음에 redoarr1이 나오는 방식으로 코딩을 했다. redo의 경우도 마찬가지로 stage는 원래대로 돌아갈 때마다 1씩 증가하게 하기만 돼서 stage++로 해결했다. score의 경우 undo를 할 때 저장해놨던 score1, score2를 사용했고 score0도 새로 도입했다.</p>
연구 반성 및 다음 주차 활동 계획	<ul style="list-style-type: none"> · U를 누르지 않은 상태에서 R을 누르면 “다른 키를 눌러주세요” 가 나오게 코딩을 해보려고 했는데 잘 안돼서 아쉽다. · 인공 지능 코딩을 어떻게 해야 할지 핸드폰으로 직접 2048 게임을 해보면서 생각을 미리 해봐야 할 것 같다.
배운 점	<ul style="list-style-type: none"> · 사실 undoarr1,2,3, redoarr1,2,3을 만들면서 코드가 매우 느려질 것이라고 예상했는데 생각보다 속도에는 달라진게 없어서 신기하다. · 코드의 어떤 문장의 위치에 따라 결과가 매우 달라질 수도 있다는 사실을 당연히 알고 있기는 했지만 이번에 시간을 엄청 낭비하면서 더 배웠다.

※ 주의 사항 : 기재할 항목 변경은 가능하며 주당 최대 2쪽이내 작성