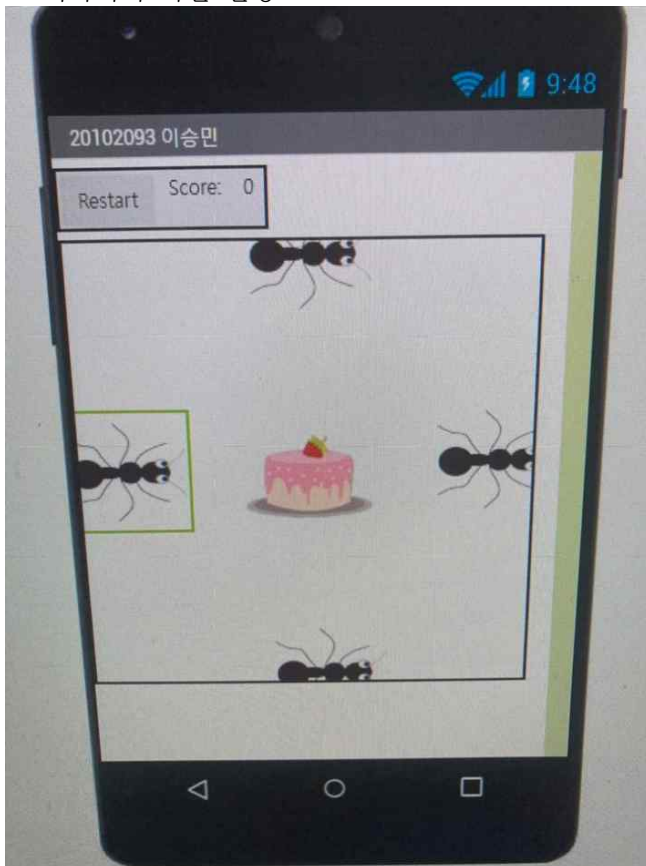


차례

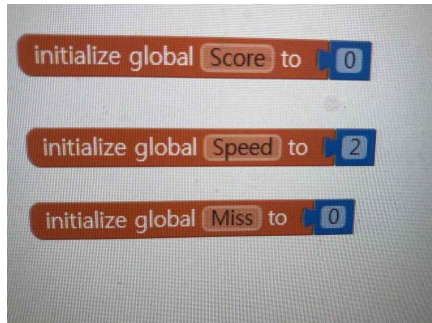
1. 디자이너 화면 설명
2. 코드 블록 화면 설명
3. 다양한 실행 결과 이미지 첨부
4. 전체 코드 블록 캡처사진 첨부

1. 디자이너 화면 설명

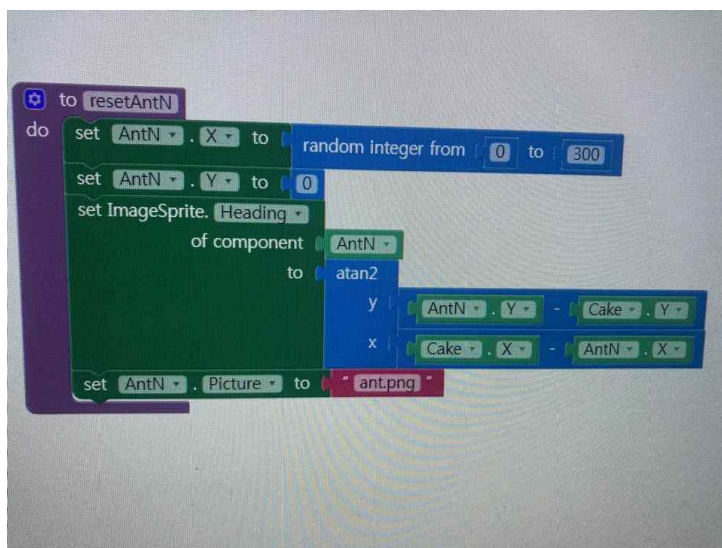


설명: 우선 screen1 Title에 학번,이름 기입을 하였다. horizontal arrangement에 restart button, score label, score value label, gameover label을 삽입하였다. 단, 여기서 game over는 화면에 보이지 않는데 visible에 체크하지 않았기 때문이다. 게임이 종료되면 화면에 나타나야 하는 gameover는 코드 블록을 통해 구현을 하였다. game canvas는 Height: 300 pixels, Width: 300 pixels로 설정하였다. 리소스로 주어진 cake.png ant.png ant2.png를 Media에서 파일 업로드를 하였다. 케이크의 Height: 60pixels, Width: 100 pixels로 설정하였다. 그리고 케이크 이미지가 중앙에 오도록 케이크의 좌표를 x=100,y=120으로 설정하였다. 개미는 동서남북 각 엣지에서 출발한다는 의미로 이미지 이름을 각각 AntE, AntW, AntS, AntN로 하였다. 개미가 출발하는 위치는 코드 블록을 통해 구현을 하였다.

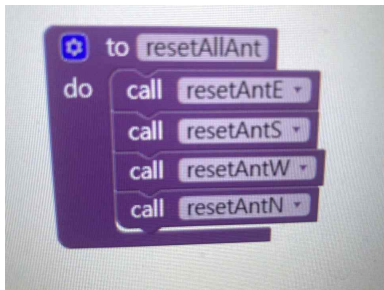
2. 코드 블록 화면 설명



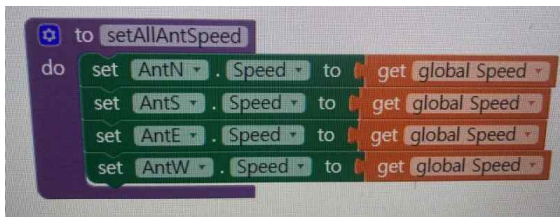
설명: 처음 글로벌 score=0, speed=2, miss=0으로 설정



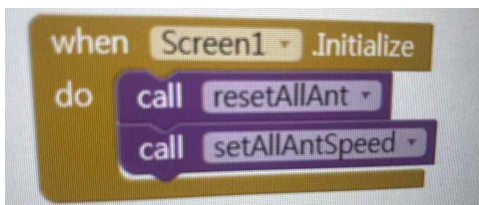
설명: to resetAntN do블록을 이용해 resetAntN이라는 동작을 정의한다. 우선 AntN은 북쪽 엣지에 있어야 하므로 y=0이고 엣지의 랜덤한 위치에서 출발하므로 x는 0~300중에서 임의로 결정되도록 set블록을 이용하여 설정한다. set imagesprite heading of component AntN 블록을 이용하여 AntN의 이미지스프라이트 방향을 설정한다. 여기서 아크탄젠트 블록을 이용해서 AntN의 방향을 항상 케이크를 향하도록 설정한다. set AntN picture를 이용해서 개미 사진을 ant.png로 설정한다. AntS의 경우 남쪽 엣지에 있어야 하므로 y=300이고 x는 0~300중에서 임의로 결정되도록 set블록을 이용하여 설정한다. 나머지 동작들의 경우 AntN과 동일하다. AntE의 경우 동쪽 엣지에 있어야 하므로 x=300이고 엣지의 랜덤한 위치에서 출발하므로 y는 0~300중에서 임의로 결정되도록 set블록을 이용하여 설정한다. 나머지 동작들의 경우 AntN과 동일하다. AntW의 경우 서쪽 엣지에 있어야 하므로 x=0이고 엣지의 랜덤한 위치에서 출발하므로 y는 0~300중에서 임의로 결정되도록 set블록을 이용하여 설정한다. 나머지 동작들의 경우 AntN의 경우와 동일하다. 이러한 과정들을 통해 resetAntN, S, E, W 동작들을 모두 정의한다.



설명: call블럭을 이용하여 resetAntN, resetAntS, resetAntE, resetAntW 동작들을 불러와서 모두 실행하는 것으로 resetAllAnt를 정의한다.



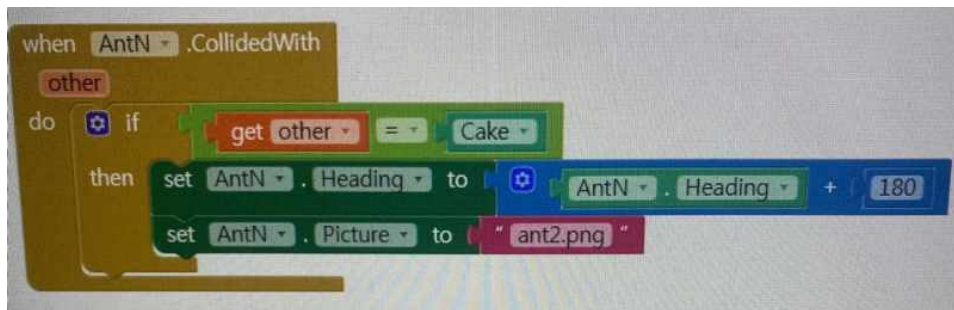
설명: AntN,AntS,AntE,AntW의 속도를 global speed로 설정하는 것을 모두 실행시키는 블록이 to setAllAntSpeed do이다. 여기서 당연히 모든 개미들의 속도는 서로 동일하다.



설명: 스크린1이 시작할 때 resetAllAnt와 setAllAntSpeed를 불러와서 실행시키는 블록이다.

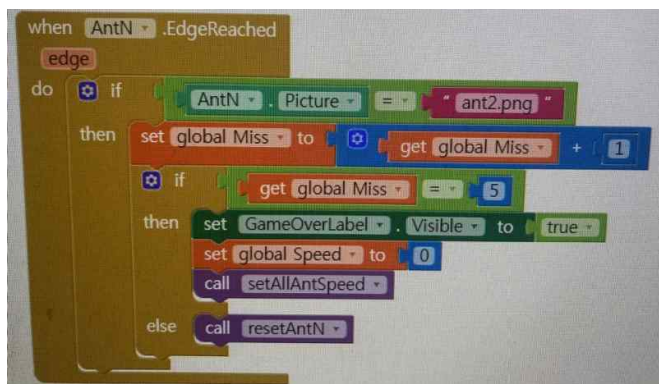


설명: restart 버튼을 클릭했을 때 global miss, score가 0으로 초기화되고 global speed는 2로 초기화된다. 동시에 set scorevalueLabel text to get globalscore 블록을 이용해 global score가 화면에 갱신되도록 설정하고 call블록을 이용해 setAllAntSpeed, resetAllAnt를 실행한다. set gameoverlabel visible to false 블록을 이용해 화면에서 game over가 보이지 않도록 한다.

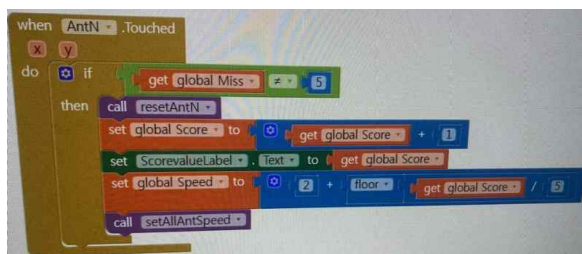


설명: 블록을 해석해보면 AntN과 other이 충돌했을 때, 근데 그 충돌한 other가 cake라면 (이를 쉽게 정리해보면 AntN이 cake와 충돌했을 때!)

이때 set블럭을 이용하여 AntN의 방향에 180도를 더해서 방향 설정을 다시 한다. 이렇게 하는 이유는 개미가 케이크와 충돌했을 때 180도를 더함으로써 개미 방향을 반대로 돌려서 원래 출발했던 옛지 지점으로 개미를 되돌아가게 하기 위해서이다. 그리고 set블럭을 이용하여 AntN의 이미지를 ant2.png로 바꾼다. AntE, S, W가 케이크와 충돌했을 때에도 동일한 과정으로 진행된다.



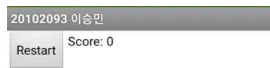
설명: 블록을 해석해보면 AntN이 엣지에 도달했을 때 AntN의 이미지가 ant2.png라면(첫번째 if문) 일단 set global miss to get global miss+1 블록을 이용하여 기존 글로벌 미스에 1을 더하여 글로벌 미스를 설정한다. 그리고 나서 만약 글로벌 미스가 5와 같아진다면(두번째 if문) set gameoverlabel visible to true블록을 이용해 gameover가 화면에 나타나도록 하고, 동시에 global speed를 0으로 설정하고 call setAllAntSpeed 블록을 이용하여 setAllAntSpeed를 실행하여 4마리 개미 speed를 모두 0으로 하여 화면이 멈추도록 한다. 만약 글로벌 미스가 5가 아니라면(else문) call resetAntN 블록을 통해 resetAntN을 실행한다. AntE, S, W가 엣지에 도달했을 때에도 동일한 과정으로 진행된다.



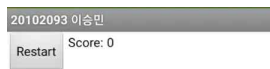
설명: 블록을 순서대로 해석해보면 AntN을 만졌을 때 만약 global miss가 5가 아니라면(if문) resetAntN을 실행시키고 기존 global score에 1을 더하여 global score를 설정한다. set scorevalueLabel text to get global score블록을 통해 global score가 화면에 나타나도록

한다. 모든 개미의 초기 속도 2이고, 점수가 5만큼 증가할 때마다 속도가 1씩 증가해야하므로 floor블럭을 이용해서 global speed를 설정한다. call setAllAntSpeed 블록을 이용하여 모든 개미의 속도를 global speed로 설정한다. AntE, S, W를 만졌을 때에도 동일한 과정으로 진행된다.

3. 다양한 실행 결과 이미지 첨부



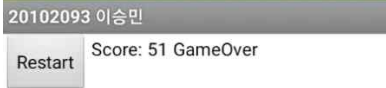
(맨 처음 실행모습, 개미들이 4곳 각각의 엣지 임의의 위치에서 케이크를 향해 출발하고 있다.)



(케이크와 충돌한 개미는 이미지가 ant.png에서 ant2.png로 바뀌었고, 출발 방향과 정반대로 방향이 바뀌어 다시 출발했던 엣지 위치로 돌아가는 것을 확인할 수 있다.)



(개미를 터치해서 스코어가 올라간 것이 화면에 표시되었다. 개미의 속도가 점수가 5증가함에 따라 1만큼 점점 빨라지고 이 과정은 global miss가 5가 되기 전까지 유효하다.)

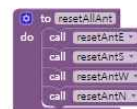
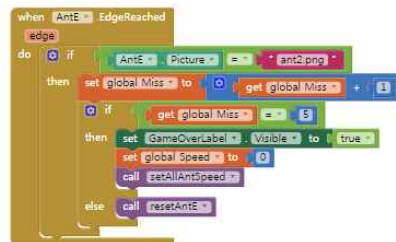
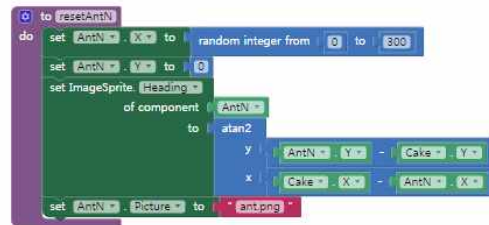
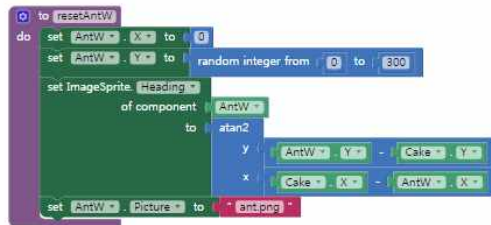
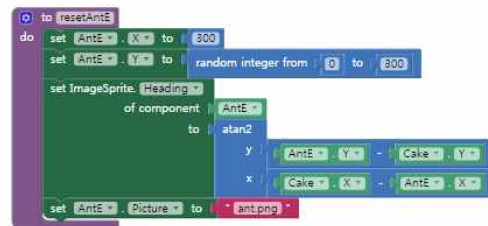
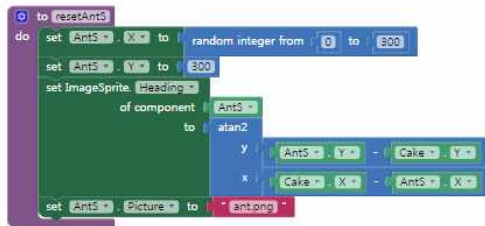


(global miss가 5가 됐을 때 상황이다. 모든 개미들의 속도가 0이 되어 화면이 정지한다. 화면에는 GameOver라는 문구가 표시된다.)



(Restart 버튼을 눌렀을 때 global miss, score가 0으로 초기화되고 global speed는 2로 초기화된다. 그리고 화면에서 game over가 사라진다.)

4. 전체 코드 블록 캡처사진 첨부





(포트폴리오 설명: 기말과제로 진행한 미니게임 앱개발 프로젝트를 보고서로 요약한 것이고 이외에도 골프게임, 계산기, 두더지잡기 게임, 바운스볼 게임 등 다양한 프로젝트를 앱인벤터와 자바를 이용하여 진행하였습니다.)