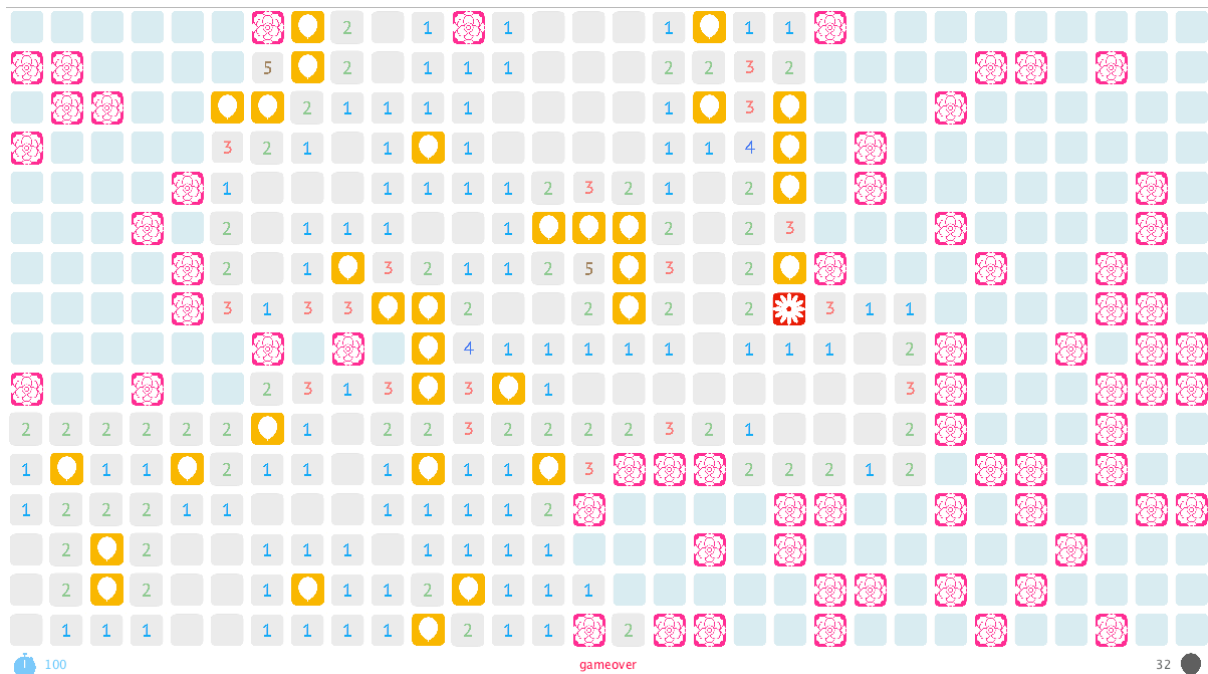


# Processing을 이용한 게임 소프트웨어 제작



대 회 명: 한동대학교 SW Festival  
부 문: 1 학년 자작소프트웨어 공모전  
학 부: 글로벌리더십학부  
학 번: 21800140  
이 름: 김예진  
제 출 일: 2018년 11월 11일

## 목차

- I.    소프트웨어 소개
- II.   제작과정
- III.  소프트웨어내용
- IV.  활용방안 및 기대효과

## I. 소프트웨어 소개

‘Look out balloons!’는 혼자서 하는 컴퓨터 게임이다.

가로 30칸, 세로 16칸의 상자들 중에 99개의 풍선이 숨어있다. 풍선이 없는 상자를 열어가며 게임을 진행하고 모든 상자를 열면 게임이 완료된다. 주어진 단서를 따라 풍선의 위치를 추리하고, 어느 상자가 안전한지 생각하며 진행하는 ‘Look out balloons!’는 추론능력을 향상시킬 수 있고, 게임 진행에 따른 성취감을 제공한다.

## II. 제작과정

### 가. 제작 기간

2018년 5월 25일부터 동년 11월 11일에 걸쳐 제작했다.

### 나. 제작 동기

‘지뢰 찾기’는 컴퓨터 고전게임중의 하나로 마이크로소프트사에서 윈도우즈 번들로 제공했던 게임이 가장 유명하다.

마이크로소프트사에서 나온 ‘지뢰 찾기’에서 실행시킨 후 시작 화면을 불러 오기 전에 종료되는 버그가 있었다. 이 버그에 대해 더 찾아보니 한국어 버전에서 흔히 있는 버그였다. 게임을 삭제하고 다시 깔아도 똑같은 일이 일어나자 버그가 고쳐지는 것보다 차라리 내가 만들어서 지뢰찾기를 하는 일이 더 빠르겠다는 생각이 들었다. 마침 SW Festival에 1학년 자작 소프트웨어 부문이 있어 이 부문에 제작한 지뢰찾기를 내면 좋겠다는 생각이 들어 본격적인 제작에 들어갔다.

### 다. 기능 정리

본격적인 개발에 들어가기 앞서 프로그램에 필요한 기능을 다음과 정리했다.

1. 무작위로 지뢰 99개를 뿌릴 것.
2. 지뢰가 있지 않은 칸 주변 8칸에 지뢰가 몇개 있는 지 표시할 것
3. 위 결과를 다른 이미지로 덮을 것
4. 왼쪽 클릭할 때

-만약

- i) 지뢰를 누르면 게임오버 이미지가 뜰 것
  - ii) 숫자를 누르면 덮은 이미지가 사라지고 숫자가 나타날 것
  - iii) 숫자가 없는 곳이라면 숫자가 있는 곳까지 덮개가 사라질 것
5. 오른쪽 클릭할 때 깃발이 꽂힐 것
  6. 지뢰를 모두 찾으면 클리어 이미지가 뜰 것
  7. 실행시작부터 시간이 얼마나 흘렀는지 표시하도록 할 것.

## 라. 실제 제작

앞서 다에서 한 기능 정리를 바탕으로 실제 제작에 들어갔다.

1. 지뢰찾기 고급의 경우 480칸 안에 지뢰가 99개 들어간다. 지뢰를 할당하는 배열을 480 사이즈로 선언해 안에 모두 0을 집어넣었다. `random()`; 함수를 이용해 0부터 479까지 무작위로 숫자를 받고, 이 위치의 배열에 0이 아닌 다른 수가 들어있는지 확인하고 없으면 1을 넣는 식으로 지뢰를 설정했다.
2. 1에서 나온 배열과 반복문과 조건문을 활용하여 주변의 지뢰가 얼마나 있는지 저장하는 배열에 주변 지뢰 개수를 저장했다.
3. 당초 생각은 1,2의 결과를 숫자와 지뢰 이미지로 실행창에 출력하고 이를 다른 이미지로 덮어 게임 시작 화면을 만드는 것이었다. 이 후 화면을 30\*16의 구역별로 쪼개어 구역별로 숫자를 부여했다. 시작 화면은 모든 구역에 동일한 덮개 이미지가 떠있고, 덮개 이미지의 구역을 클릭했을 때 그 위치가 지뢰의 위치라면 게임이 끝났음을, 지뢰의 위치가 아니라면 그 위치에 해당하는 숫자를 출력하도록 했다. 1부터 시작 화면까지는 `void setup()` 구역에서 담당하고, 누르면 나오는 이미지들은 `void draw()` 구역에서 담당한다.
4. 위에서 사용했던 배열의 차원은 1차원이었다. 2차원 배열을 사용할 생각이었으나 Processing에서는 2차원 배열을 지원하지 않아 1차원 배열을 사용하고 이후 '%' 연산자와 '/' 연산자를 사용해 인덱스를 30진법으로 분리하는 방법을 사용했다. 30의 자리 수와 1의 자리 수를 각각 다른 변수에 저장하고, 30자리의 수는 세로 방향, 1의자리 수는 가로 방향을 나타내는 좌표로 활용했다.

만약 마우스가 특정 좌표에 위치해 왼쪽 클릭을 했을 때를 가정해 보자.

- i) 그 위치가 지뢰의 위치였으면 깃발이 표시되지 않은 모든 지뢰에 폭발 이미지를 띄우고, 누른 위치에는 다른 모양의 폭발이미지를 띄운다. 중앙 하단에는 'game over' 메시지를 띄운다. 이 부분의 경우 반복문과 조건문이 많이 사용되어 가독성이 떨어지기에 `void explosion(int groundZero);`로 분리하였다.
- ii) 만약 그 위치가 지뢰가 없고, 주변 8칸에 지뢰가 있었다면 지뢰가 주변에 얼마나 있는지 숫자를 표시하는 칸이었다면, 그 숫자를 표시하도록 하였다. 숫자에 따라 나타나는 이미지가 달라지고, 아래에서도 계속 쓰이는 기능이기 때문에 특정 칸의 이미지를 실행창에 띄우는 함수인 `void around(int hand, int air, int con);`을 작성하여 계속 호출할 수 있도록 하였다.
- iii) 만약 그 위치 주변 8칸에 아무 지뢰도 없다면 그 위치부터 지뢰가 몇개 있는지 표시하는 숫자가 있는 칸까지 확장되도록 하였다. 이 또한 배열을 이용한 flag를 활용하였고, 숫자 이미지의 표시에는 ii에서 작성한 함수인 `around`를 활용하였다..

5. 특정 좌표에서 오른쪽 클릭을 하면 깃발이미지가 출력되지만, 게임 진행의 원활을 위하여 실행창 우하단에 깃발의 개수를 출력하도록 하기 위하여 `void mousePressed()`를 이용해 우클릭 시 깃발 개수를 나타내는 배열에 좌표값에 1이 들어있지 않다면 1이 되도록 하고 좌클릭시 0이 아니라면 0이 들어가도록 했다. 깃발의 총 개수는 반복문을 사용하여 배열 전체의 총합계를 변수에 저장하고, 이 변수를 실행창 우하단에 띄우도록 하였다.
6. 그 좌표의 개봉 유무를 표시하는 배열을 선언하였다. ii에서 작성된 `around`함수에 개봉되었으면 배열에 1을 넣도록 하여 5와 같은 방식으로 총 개봉 수를 확인하고, 이를 지뢰를 제외한 전체 칸수와 비교하여 만약에 지뢰가 없는 칸이 모두 개봉되었으면 게임을 끝내도록 하였다.
7. `millis()`함수를 이용하여 시간을 재었다. 시간이 얼마나 지났는지 저장하는 변수를 선언하고, 이 변수는 `millis()`보다 1000만큼 크도록 초기화 했다. 이

후 `millis()`가 시간변수와 같아지면 1초가 흐른 것으로 판단하고, 이를 출력한 뒤, 시간변수를 다시 그 시점의 `millis()`보다 1000 큰 값으로 재설정했다.

## 마. 프로그램 내 이미지 추출

프로그램 내에서 사용한 이미지는 모두 애플사의 `pages`에서 제공하는 클립아트이다.

## III. 소프트웨어내용

'Look out balloons!'의 목표는 단순하다. 풍선이 들어있지 않은 상자를 모두 열면 된다.

실행시키면 화면에는 480 개의 상자가 나타난다. 이 상자 중에는 무작위로 99 개의 풍선이 들어있고, 상자를 마우스 왼쪽으로 클릭하면 안의 내용물을 보여준다. 운이 나쁘면 첫번째 클릭에도 풍선을 선택할 수 있고, 풍선을 선택하면 상자에 있던 모든 풍선이 터지고 게임이 종료된다.

상자 안에는 풍선이나 주변 8 칸에 풍선이 몇개 있는지 알려주는 숫자도 들어있다. 이 숫자를 이용하여 풍선이 있는 칸을 유추해 낸다. 만약 상자 안에 아무것도 들어있지 않으면 상자 주변 8 칸에 지뢰가 없다는 말이다. 아무것도 들어있지 않은 상자는 주변의 상자를 열어 숫자가 있는 칸까지 열어준다.

풍선이 들어있는게 확실한 칸은 마우스 오른쪽 클릭을 통해 풍선이 들어있다고 표시할 수 있다. 화면 오른쪽 아래에 몇 개를 표시했고, 찾아야 할 지뢰가 얼마나 있는지 알려주는 숫자가 풍선 옆에 있다.

다만 표시해 놓았다고 해서 마음을 놓으면 안된다. 실수로 표시해 놓은 칸을 마우스 왼쪽으로 클릭하면 풍선이 터져 게임이 종료될 수 도 있다. 또, 표시했지만 아니라고 확신이 들면 그 칸을 마우스 왼쪽으로 클릭하여 열면 된다.

화면 왼쪽 아래에는 시계와 함께 숫자가 있다. 실행 순간부터 시간이 얼마나 지났는지 초단위로 표시해 준다. 만약 풍선을 터뜨리거나, 풍선 없는 상자를 모두 열면 시계는 멈춘다.

#### IV. 활용방안 및 기대효과

    풍선이 있는 위치를 추리해 가며 스트레스를 해소하고, 추론능력을 향상시킬 수 있다.