
Minesweeper++
by 졸업시켜조

Table of Contents

Team Introduction

Team Name

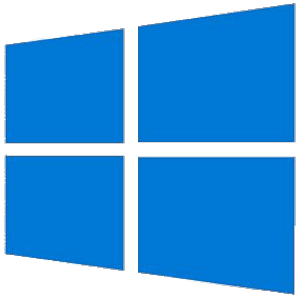
"졸업시켜조"

Team Roles

- 이승섭: 기계항공공학부 13학번 - 팀 리더, 그래픽 설계, 코더
- 김동영: 생물교육과 16학번 - 로직 설계, 코더, 발표
- 김재현: 경제학부 17학번 - 기본 기능 설계, 코더, 발표 준비

Project Ideation

Remote collaboration...



Familiar, yet challenging?

Project Design: Basic Functionality (1)

게임 관련 설명

기존 지뢰 찾기 게임에서 **일정 수의 시도 이후 지뢰들의 위치를 변화시키는 방식**으로 신선함을 더한 게임

게임 진행 방법

- **맵 설정:** 9x9, 16x16, 30x16 크기의 지도에 지뢰가 각각 10개, 40개, 99개가 숨겨져 있음
- **게임 방법:**
  - 플레이어는 좌클릭으로 지뢰가 아닌 곳을 누르거나, 우클릭으로 지뢰인 곳에 깃발을 세우는 것이 가능
  - 게임 시작 후 경과한 시간 및 남은 지뢰의 수를 확인하는 것이 가능
  - 클릭한 타일 주위 8개 타일 중 지뢰 타일의 개수만큼 클릭한 타일에 숫자가 표시됨
  - 지뢰 타일이 주위에 없을 경우, 클릭한 타일은 빈 타일로 변하고 주위 모든 빈 타일이 한 번에 밝혀짐
- **승리/패배 조건:** 지뢰가 아닌 모든 곳을 밝히면 승리, 지뢰를 한 번이라도 누를 시 패배 .

게임 이미지 (예시)

Project Design: Basic Functionality (2)

게임 주요 기능 및 사용 함수

1. "__init__": 보드와 게임 상태를 초기화

```
def __init__(self, width, height, mines):
    self.width = width
    self.height = height
    self.mines = mines
    self.board = [[0 for _ in range(width)] for _ in range(height)]
    self.revealed = [[False for _ in range(width)] for _ in range(height)]
    self.flags = [[False for _ in range(width)] for _ in range(height)]
    self.place_mines()
```

2. "place_mines": 맵 내에 무작위로 지뢰 배치

```
def place_mines(self):
    for _ in range(self.mines):
        while True:
            x = random.randint(0, self.width - 1)
            y = random.randint(0, self.height - 1)
            if self.board[y][x] == 0:
                self.board[y][x] = -1
                self.update_numbers(x, y)
                break
```

3. "update_numbers": 지뢰 숫자를 설정

```
def update_numbers(self, x, y):
    for dy in range(-1, 2):
        for dx in range(-1, 2):
            nx, ny = x + dx, y + dy
            if 0 <= nx < self.width and 0 <= ny < self.height and self.board[ny][nx] != -1:
                self.board[ny][nx] += 1
```

4. "reset": 게임 보드 초기화

```
def reset(self):
    self.board = [[0 for _ in range(self.width)] for _ in range(self.height)]
    self.revealed = [[False for _ in range(self.width)] for _ in range(self.height)]
    self.flags = [[False for _ in range(self.width)] for _ in range(self.height)]
    self.place_mines()
```

Project Design: Basic Functionality (3)

게임 주요 기능 및 사용 함수

5. "reveal": *클릭한 타일을 밝히기 + 빈 타일의 인접 타일 밝히기*

```
def reveal(self, x, y):
    if self.revealed[y][x] or self.flags[y][x]:
        return
    self.revealed[y][x] = True
    if self.board[y][x] > 0:
        return
    if self.board[y][x] == 0:
        for dy in range(-1, 2):
            for dx in range(-1, 2):
                nx, ny = x + dx, y + dy
                if 0 <= nx < self.width and 0 <= ny < self.height:
                    self.reveal(nx, ny)
```

6. "toggle_flag": 깃발을 표시하거나 제거

```
def toggle_flag(self, x, y):
    if not self.revealed[y][x]:
        self.flags[y][x] = not self.flags[y][x]
```

7. "get_state": 현재 게임 상태 확인

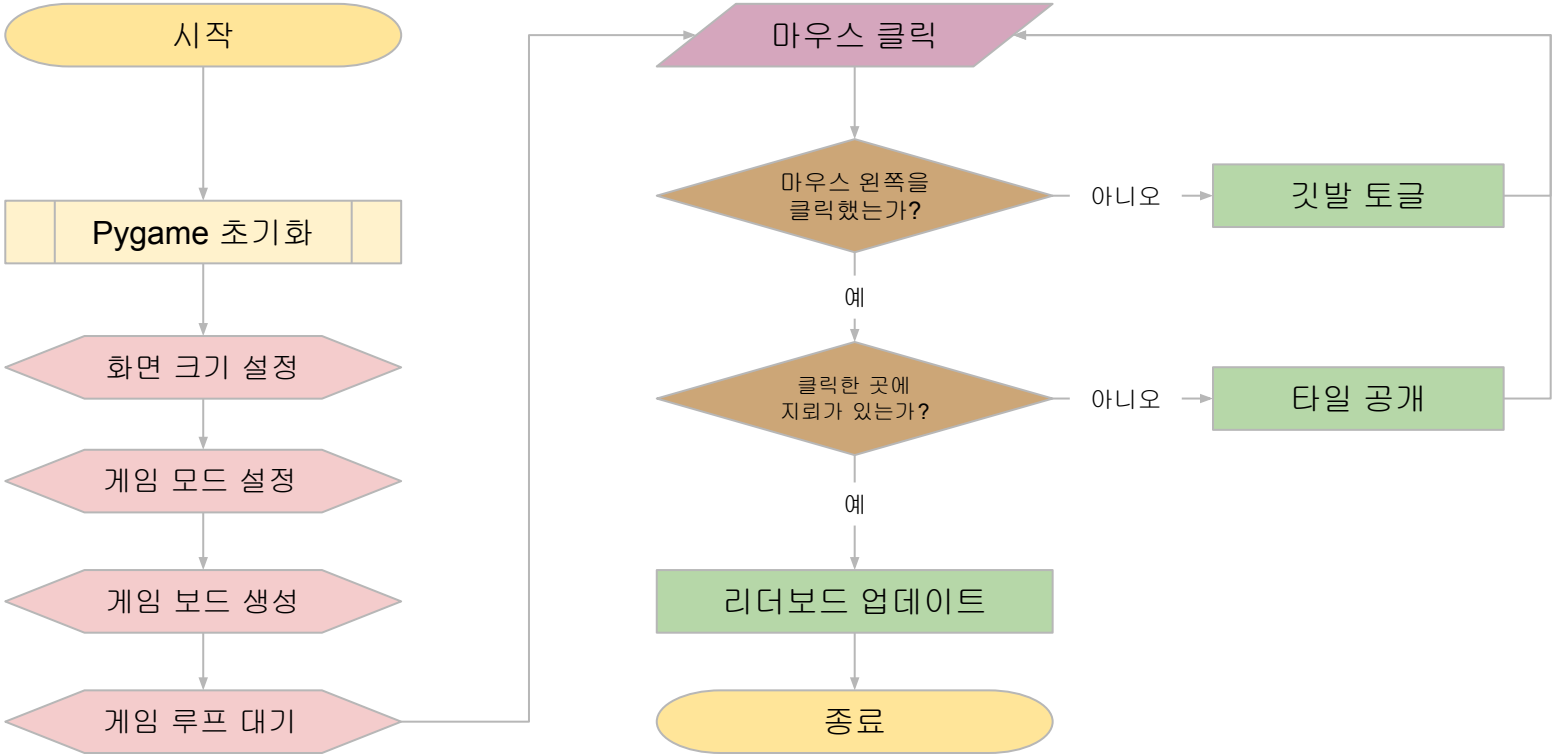
```
def get_state(self):
    revealed_count = sum(sum(row) for row in self.revealed)
    return 'win' if revealed_count == self.width * self.height - self.mines else 'playing'
```

8. "main": 게임을 초기화, 이벤트를 처리 및 화면 업데이트

```
def main():
    clock = pygame.time.Clock()
    game_over = False
    clicks = 0
    difficulty = 'easy'
    width, height, mines = GRID_SIZES[difficulty]
    board = MinesweeperBoard(width, height, mines)
    while not game_over:
        screen.fill(WHITE)
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                game_over = True
            elif event.type == pygame.MOUSEBUTTONDOWN:
```

이하 생략

Project Design: New Logic



Project Design: Graphic Interface (1)

PyGame을 이용한 그래픽 요소

1. "BOARD_OPTIONS": 난이도별 보드 크기를 나타낸 *dictionary*.

2. "tile_numbers": 각 타일별 이미지 파일을 담은 *list*.

3. "Tile": 타일 객체 *class*.

4. "Board": 보드 객체 *class*.

```python
# Board options
BOARD_OPTIONS = {
    '8x8': (8, 8, 10),
    '16x16': (16, 16, 40),
    '30x16': (30, 16, 99)
}

# Leaderboard
LEADERBOARD = {
    '8x8': float('inf'),
    '16x16': float('inf'),
    '30x16': float('inf')
}

tile_numbers = []
for i in range(1, 9):
    tile_numbers.append(pygame.transform.scale(pygame.image.load(os.path.join
    ("./sprites", f"Tile{i}.png")), (TILESIZE, TILESIZE)))
```

```python
piratecat-lover, 3 hours ago | 1 author (piratecat-lover)
class Tile:
    def __init__(self, x, y, image, type, revealed=False, flagged=False):
        self.x, self.y = x * TILESIZE, y * TILESIZE
        self.image = image
        self.type = type
        self.revealed = revealed
        self.flagged = flagged

    def draw(self, board_surface):
        if not self.flagged and self.revealed:

piratecat-lover, 3 hours ago | 1 author (piratecat-lover)
class Board:
    def __init__(self):
        self.board_surface = pygame.Surface((WIDTH, HEIGHT))
        self.board_list = [[Tile(col, row, tile_empty, ".") for row in range
        (ROWS)] for col in range(COLS)]
        self.place_mines()
        self.place_clues()
        self.dug = []
```

Project Design: Graphic Interface (2)

게임에 필요한 이미지 파일 (Source: DALL-E)

1. Sprites: 지뢰, 타일, 깃발 등의 이미지.

***2. Title Screen*:** 게임 시작 화면.

Project Timeline

Date (Jul)	8	9	10	11	12	15	16	17	18	19	22	23	24	25
Ideation	All													
Code			SSL, JHK, DYK											
Graphics			SSL											
Deployment											SSL			
Presentation				JHK, DYK									SSL, DYK	

TASKS		
Code 1. Basic Functionality - JHK 2. Randomizer Logic - DYK 3. Graphic Interface - SSL	**Deployment** 1. Github - SSL 2. Executable - SSL	**Presentation** 1. Slides - All 2. Speech - JHK, DYK


Thank you for your attention!