배열-기반 그리드(Array-Backed Grids)

순천향대학교 컴퓨터공학과
이 상 정

학습 내용

- 보드 게임 등을 위한 그리드(grid)소개
- 그리드 셀 예
  - 소개
  - 2차원 배열 생성
  - 마우스 클릭 시 값 지정
  - 그리드 그리기

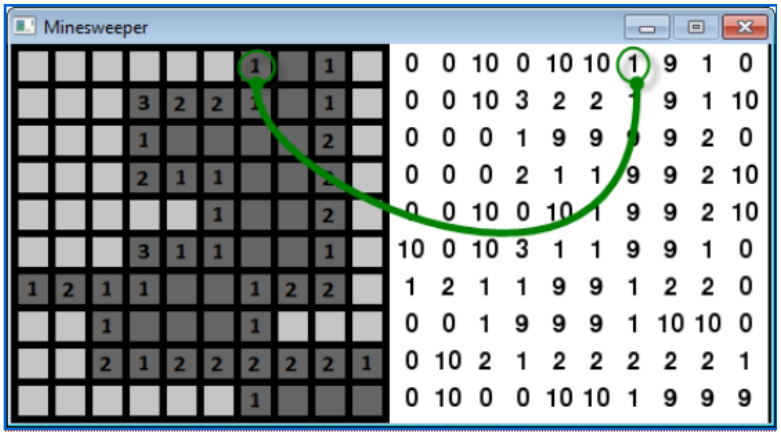
그리드 소개

- 그리드(grid)
  - 바국판, 격자 등의 눈금, 선
  - tic-tac-toe 등의 보드 게임 등에 활용
- 그리드의 셀 표현을 위해 2차원 배열 사용
  - 배열의 인덱스가 보드의 위치 표시
  - 배열 값이 표시되는 내용 표시
- Tic-tac-toe 보드 배열 값
  - 0 -> 빈 공간
  - 1 -> X
  - 2 -> O

	O	O
	X	
X		

0	2	2
0	1	0
1	0	0

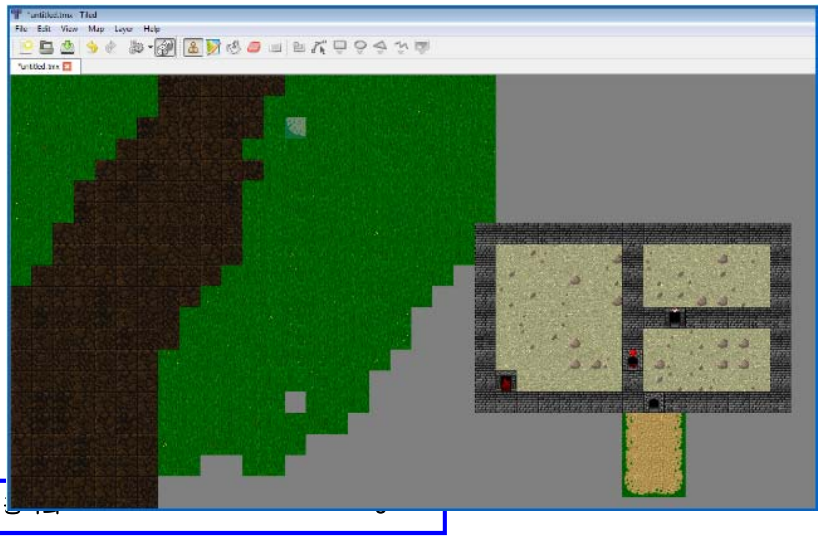
지뢰 찾기(Minesweeper) 게임 예



- 지뢰 찾기 배열 값
  - 10 -> 지뢰
  - 0 -> 클릭하지 않은 공간
  - 9 -> 클릭해서 클리어 된 공간
  - 1 ~ 8 -> 주변 사각형 주위의 지뢰 수

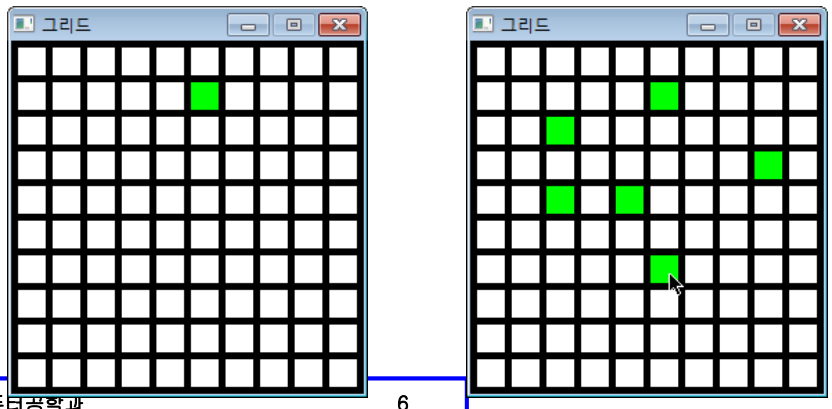
어드벤처 게임의 지도 예

- 어드벤처 게임 등에서의 지도(map) 등은 타일맵 편집기 (tiled map editor)를 사용해서 생성
  - 각 위치 영역은 특정 타입을 표시하는 숫자로 구성된 그리드
  - Tiled Qt(www.mapeditor.org)를 사용하여 생성된 지도 예



그리드 셀 – 소개

- 10 x 10 그리드 셀 예
  - 윈도우 크기는 255 x 255 픽셀
  - 한 셀의 크기는 20 x 20 픽셀: width = 20, height = 20
  - 셀 간의 여백은 5: margin = 5
  - 셀은 흰색에서 마우스 클릭하면 녹색으로 변함
    - 2차원 배열 값: 흰색 0, 녹색 1



그리드 셀 – 2차원 배열 생성

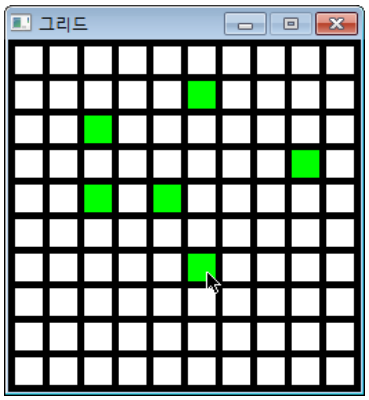
- 리스트들의 리스트로 2차열 배열 생성
  - 파이썬의 2차원 배열 생성은 다소 복잡
    - 배열 등 과학 계산용 패키지 NumPy(http://www.numpy.org/) 사용하면 쉽게 배열 구현 가능
  - 행의 리스트 생성 후 열의 리스트 생성하여 추가

```
# 2차원 배열 생성
# 2차원 배열은 리스트의 리스트
grid = []
for row in range(10):
    # 한 행 전체를 위한 빈 배열 추가,
    grid.append([])
    for column in range(10):
        grid[row].append(0)    # 열을 추가, 0으로 초기화

# 행 1, 열 5를 1로 초기화
grid[1][5] = 1
```


그리드 셀 – 마우스 클릭 시 값 지정

- 그리드 셀에 마우스 클릭하면 해당 셀의 색이 녹색으로 변함
  - 해당 셀의 2차원 배열 값을 1로 지정
  - 마우스 클릭 (x,y) 픽셀 좌표 값을 2차원 배열의 인덱스로 변환 필요
    - x // (width+margin), y // (height+margin)
      - // 는 정수 나눗셈 연산자, 정수의 몫만 계산

```
if event.type == pygame.MOUSEBUTTONDOWN:
        # 마우스 클릭 위치
        pos = pygame.mouse.get_pos()
        # x,y 스크린 좌표를 그리드 좌표로 변환
        column = pos[0] // (width+margin)
        row = pos[1] // (height+margin)
        # 그리드 값 1로 지정
        grid[row][column] = 1
        print("Click ",pos,"Grid coordinates: ",row,column)
```

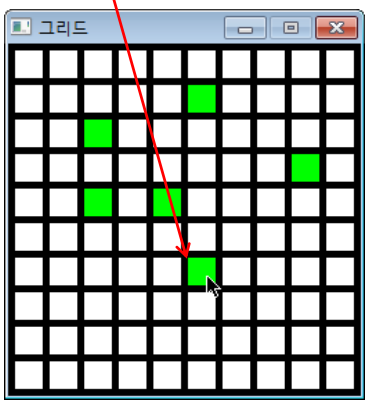


그리드 셀 – 그리드 그리기

- 2차원 배열의 값을 조사하여 해당 셀의 색을 지정
  - 해당 셀의 사각형 위치(왼쪽 위 좌표) 지정
  - 배열 인덱스에서 (x,y) 좌표 지정
    x = (margin+width)*column+margin
    y = (margin+height)*row+margin

grid[6][5],
x = (5+20)*5 + 5 = 130
y = (5+20)*6 + 5 = 155

```
# 그리드 그리기
for row in range(10):
    for column in range(10):
        color = white
        if grid[row][column] == 1:
            color = green
        pygame.draw.rect(screen,color,[(margin+width)*column+margin,
(margin+height)*row+margin, width, height])
```



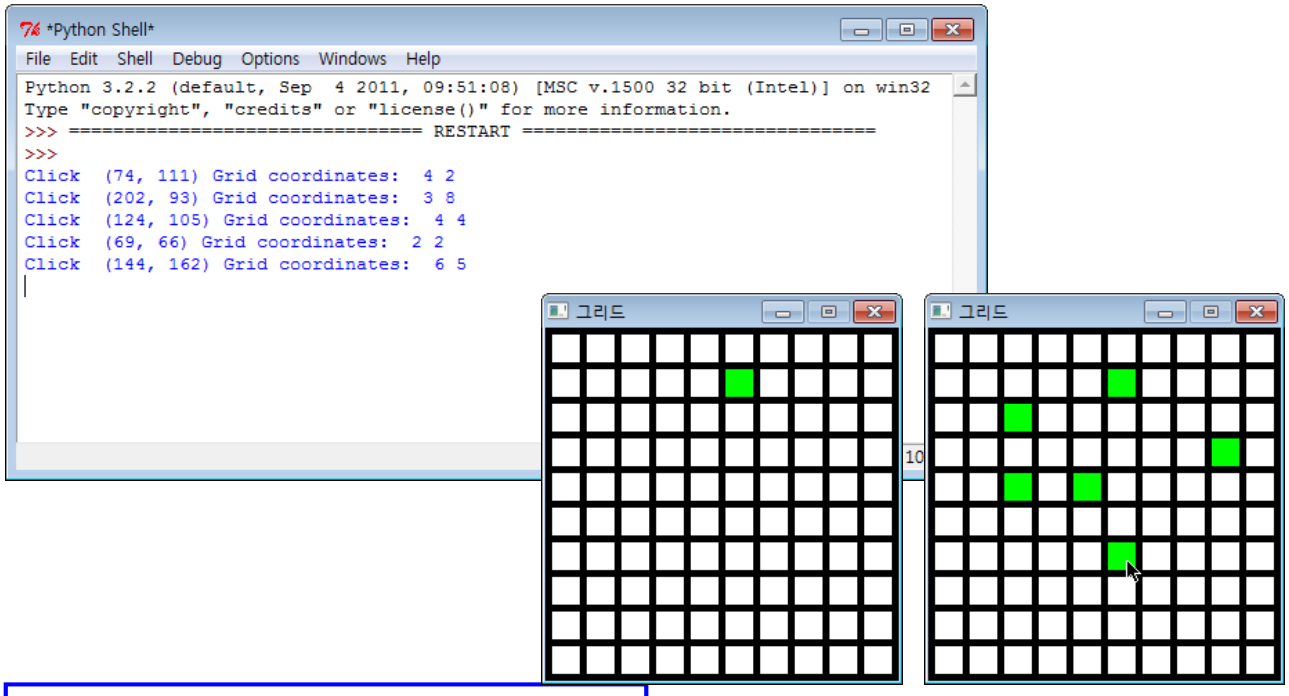
그리드 코드 (1)

```
import pygame

black = ( 0, 0, 0)
white = ( 255, 255, 255)
green = ( 0, 255, 0)
red = ( 255, 0, 0)

# 그리드 크기 지정
width=20
height=20
# 그리드 셀 간 여백
margin=5

# 2차원 배열 생성
# 2차원 배열은 리스트의 리스트
grid=[]
for row in range(10):
    # 행을 위한 빈 배열 추가
    grid.append([])
    for column in range(10):
        grid[row].append(0) # 열을 추가

# 행 1, 열 5를 1로 초기화
grid[1][5] = 1

pygame.init()
# 윈도우 설정
size=[255,255]
screen=pygame.display.set_mode(size)
pygame.display.set_caption("그리드")

done=False
clock=pygame.time.Clock()

while done==False:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True
        if event.type == pygame.MOUSEBUTTONDOWN:
            # 마우스 클릭 위치
            pos = pygame.mouse.get_pos()
```

그리드 코드 (2)

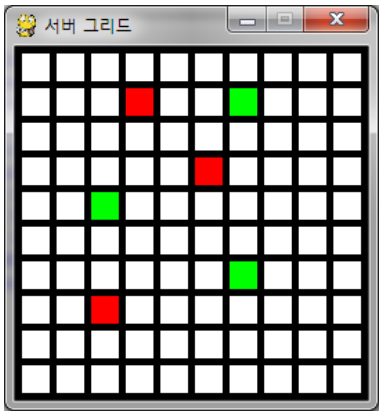
```
            # x,y 스크린 좌표를 그리드 좌표로 변환
            column=pos[0] // (width+margin)
            row=pos[1] // (height+margin)
            # 그리드 값 1로 지정
            grid[row][column]=1
            print("Click ",pos,"Grid coordinates: ",row,column)

    screen.fill(black)
    # 그리드 그리기
    for row in range(10):
        for column in range(10):
            color = white
            if grid[row][column] == 1:
                color = green
            pygame.draw.rect(screen,color, [(margin+width)*column+margin,
                                            (margin+height)*row+margin, width, height])

    clock.tick(20)
    pygame.display.flip()

pygame.quit()
```

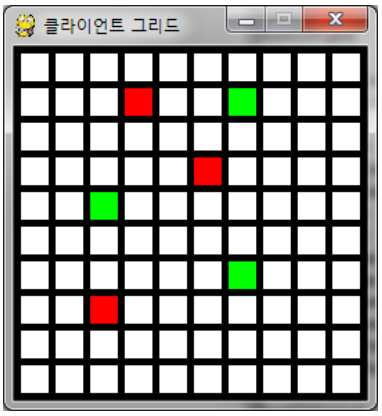

시험주행



네트워크 그리드 예 – 소개

- 배열 기반 그리드를 네트워크 버전으로 개선
  - 클릭한 셀의 2차원 배열 인덱스 값을 네트워크 데이터로 송수신
  - 클라이언트, 서버의 클릭한 그리드 값(색) 구분
    빈 공간: 0 (흰색), 서버: 1 (녹색), 클라이언트: 2 (적색)





네트워크 그리드 예 – 데이터 송신

- 마우스 클릭 시 데이터 송신
  - 송신 시 리스트를 바이트형으로 변환하여 송신

```
# 서버 송신
if event.type == pygame.MOUSEBUTTONDOWN:
    ……
    # 그리드 값 1(서버)로 지정
    grid[row][column]=1
    # 서버 인덱스를 바이트로 변환하여 송신
    serverIndex = [row, column]
     indexBytes = bytes(serverIndex)
     connectionSocket.send(indexBytes)
     ……
```

```
# 클라이언트 송신
if event.type == pygame.MOUSEBUTTONDOWN:
  ……
  # 그리드 값 2(클라이언트)로 지정
  grid[row][column]=2
  # 클라이언트 인덱스를 바이트로 변환하여 송신
  clientIndex = [row, column]
  indexBytes = bytes(clientIndex)
  clientSocket.send(indexBytes)
    ……
```

네트워크 그리드 예 – 데이터 수신

- 스레드 함수로 구현
  - 수신 시 바이트 데이터를 리스트 형으로 변환 후 그리드 표시

```
# 서버 메시지 수신 함수(스레드)
def serverRecv():
  while True:
    # 클라이언트의 클릭 배열 인덱스 수신 후
    # 리스트형으로 변환
    clientIndex = list(connectionSocket.recv(1024))
    row = clientIndex[0]
    column = clientIndex[1]
    # 그리드 값 2(클라이언트)로 지정
    grid[row][column] = 2

# 스레드 생성 및 실행
Thread(target=serverRecv).start()
```

```
# 클라이언트 메시지 수신 함수(스레드)
def clientRecv():
  while True:
    # 서버의 클릭 배열 인덱스 수신 후
    # 리스트형으로 변환
    serverIndex = list(clientSocket.recv(1024))
    row = serverIndex[0]
    column = serverIndex[1]
    # 그리드 값 1(서버)로 지정
    grid[row][column] = 1

# 스레드 생성 및 실행
Thread(target=clientRecv).start()
```


네트워크 그리드 예 – 서버 프로그램

```
### 그리드 서버 프로그램
import pygame
from socket import *
from threading import *

black = ( 0, 0, 0)
white = ( 255, 255, 255)
green = ( 0, 255, 0)
red = ( 255, 0, 0)
# 그리드 크기 지정
width=20
height=20
# 그리드 셀 간 여백
margin=5
# 서버 메시지 수신 함수(스레드)
def serverRecv():
  while True:
    # 클라이언트의 클릭 배열 인덱스 수신 후
    # 리스트형으로 변환
    clientIndex = list(connectionSocket.recv(1024))
    row = clientIndex[0]
    column = clientIndex[1]
    # 그리드 값 2(클라이언트)로 지정
    grid[row][column] = 2
    print("Receive, Grid Client Index: ",row,column)

# 2차원 배열 생성
# 2차원 배열은 리스트의 리스트, 0: 빈공간,
# 1: 서버, 2: 클라이언트
grid=[]
for row in range(10):
  # 행을 위한 빈 배열 추가
  grid.append([])
  for column in range(10):
    grid[row].append(0) # 열을 추가

# 서버 네트워크 연결 초기화
print("Server TCP initialize...")
# 호스트 주소 지정
serverPort = 7000
# 서버 소켓 생성
serverSocket = socket(AF_INET, SOCK_STREAM)
# 주소와 소켓 결합
serverSocket.bind(('', serverPort))
```

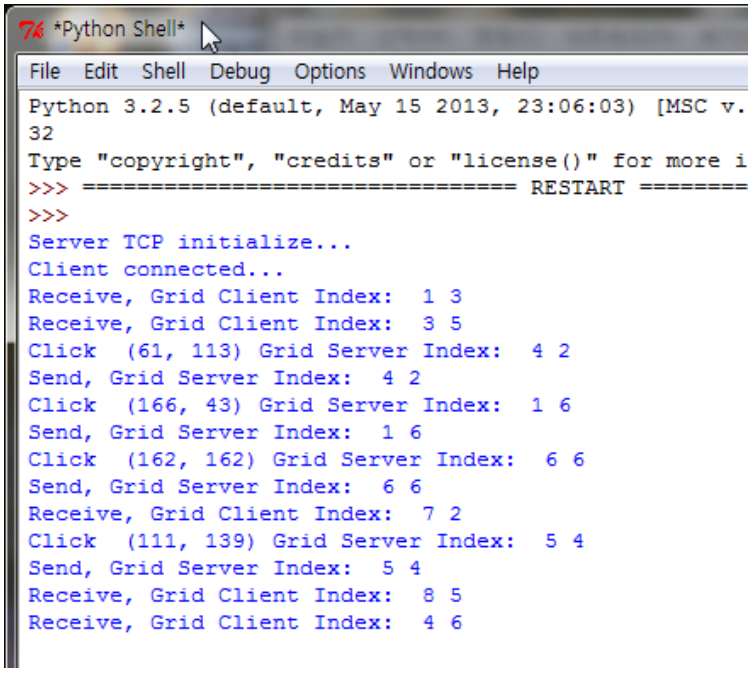
```
# 연결 요청 청취
serverSocket.listen(1)
# 연결 요청 수락, 연결 소켓 리턴
connectionSocket, addr = serverSocket.accept()
print("Client connected...", addr)
# 스레드 생성 및 실행
Thread(target=serverRecv).start()
pygame.init()
# 윈도우 설정
size=[255,255]
screen=pygame.display.set_mode(size)
pygame.display.set_caption("서버 그리드")
done=False
clock=pygame.time.Clock()

while done==False:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True
        if event.type == pygame.MOUSEBUTTONDOWN:
            # 마우스 클릭 위치
            pos = pygame.mouse.get_pos()
            # x,y 스크린 좌표를 그리드 좌표로 변환
            column=pos[0] // (width+margin)
            row=pos[1] // (height+margin)
            # 그리드 값 1(서버)로 지정
            grid[row][column]=1
            print("Click ",pos,"Grid Server Index: ",
                                        row,column)
            # 서버 인덱스를 바이트로 변환하여 송신
            serverIndex = [row, column]
            indexBytes = bytes(serverIndex)
            connectionSocket.send(indexBytes)
            print("Send, Grid Server Index: ",
                                        row,column)
    # 그리드 그리기
    screen.fill(black)
    for row in range(10):
        for column in range(10):
            color = white
            # 서버 클릭 그리드 값 색 지정
            if grid[row][column] == 1:
                color = green
            # 클라이언트 클릭 그리드 값 색 지정
            elif grid[row][column] == 2:
                color = red
            pygame.draw.rect(screen,color,
             [(margin+width)*column+margin,
             (margin+height)*row+margin,width,height])

    clock.tick(20)
    pygame.display.flip()

pygame.quit()
```

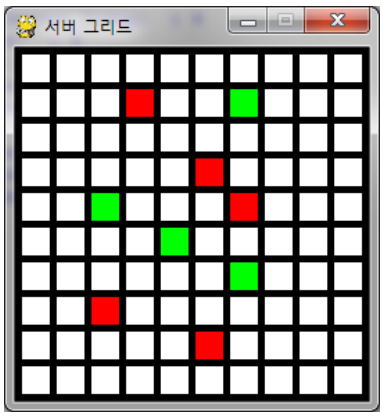

네트워크 그리드 예 – 클라이언트 프로그램

```
import pygame
from socket import *
from threading import *

black = ( 0, 0, 0)
white = ( 255, 255, 255)
green = ( 0, 255, 0)
red = ( 255, 0, 0)
# 그리드 크기 지정
width=20
height=20
# 그리드 셀 간 여백
margin=5

# 클라이언트 메시지 수신 함수(스레드)
def clientRecv():
    while True:
        # 서버의 클릭 배열 인덱스 수신 후
        # 리스트형으로 변환
        serverIndex = list(clientSocket.recv(1024))
        row = serverIndex[0]
        column = serverIndex[1]
        # 그리드 값 1(서버)로 지정
        grid[row][column] = 1
        print("Receive, Grid Server Index: ",row,column)

# 2차원 배열 생성
# 2차원 배열은 리스트의 리스트, 0: 빈공간,
# 1: 서버, 2: 클라이언트
grid=[]
for row in range(10):
    # 행을 위한 빈 배열 추가
    grid.append([])
    for column in range(10):
        grid[row].append(0) # 열을 추가

# 클라이언트 네트워크 연결 초기화
print("Client TCP initialize...")
# 연결할 서버 주소 지정
serverName = '127.0.0.1'
serverPort = 7000
# 클라이언트 소켓 생성
clientSocket = socket(AF_INET, SOCK_STREAM)
```

```
# 서버에 연결 요청
clientSocket.connect((serverName, serverPort))
print("Server connected...")
# 스레드 생성 및 실행
Thread(target=clientRecv).start()

pygame.init()
# 윈도우 설정
size=[255,255]
screen=pygame.display.set_mode(size)
pygame.display.set_caption("클라이언트 그리드")
done=False
clock=pygame.time.Clock()

while done==False:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True
        if event.type == pygame.MOUSEBUTTONDOWN:
            # 마우스 클릭 위치
            pos = pygame.mouse.get_pos()
            # x,y 스크린 좌표를 그리드 좌표로 변환
            column=pos[0] // (width+margin)
            row=pos[1] // (height+margin)
            # 그리드 값 2(클라이언트)로 지정
            grid[row][column]=2
            print("Click ",pos,"Grid Server Index: ",
                                        row,column)
            # 클라이언트 인덱스를 바이트로
            # 변환하여 송신
            clientIndex = [row, column]
            indexBytes = bytes(clientIndex)
            clientSocket.send(indexBytes)
            print("Send, Grid Client Index: ",
                                        row,column)
    # 그리드 그리기
    screen.fill(black)
    for row in range(10):
        for column in range(10):
            color = white
            # 서버 클릭 그리드 값 색 지정
            if grid[row][column] == 1:
                color = green
            # 클라이언트 클릭 그리드 값 색 지정
            elif grid[row][column] == 2:
                color = red
            pygame.draw.rect(screen,color,
            [(margin+width)*column+margin,
            (margin+height)*row+margin,width,height])
    clock.tick(20)
    pygame.display.flip()

pygame.quit()
```

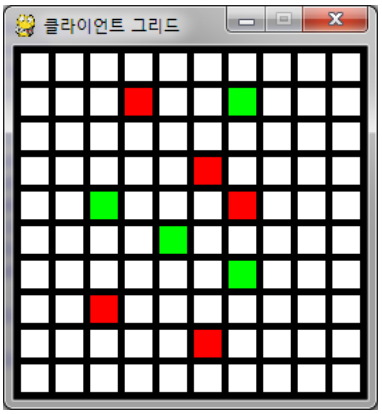

시험주행 – 서버





시험주행 – 클라이언트

```
*Python Shell*
File  Edit  Shell  Debug  Options  Windows  Help
Python 3.2.5 (default, May 15 2013, 23:06:03) [MSC
32
Type "copyright", "credits" or "license()" for mor
>>> ================================ RESTART =====
>>>
Client TCP initialize...
Server connected...
Click  (88, 39) Grid Server Index:  1 3
Send, Grid Client Index:  1 3
Click  (141, 90) Grid Server Index:  3 5
Send, Grid Client Index:  3 5
Receive, Grid Server Index:  4 2
Receive, Grid Server Index:  1 6
Receive, Grid Server Index:  6 6
Click  (63, 192) Grid Server Index:  7 2
Send, Grid Client Index:  7 2
Receive, Grid Server Index:  5 4
Click  (137, 216) Grid Server Index:  8 5
Send, Grid Client Index:  8 5
Click  (172, 113) Grid Server Index:  4 6
Send, Grid Client Index:  4 6
```



과제

1. 앞에서 소개된 그리드 프로그램을 작성하고 실행

2. 앞에서 배운 내용을 사용한 임의의 프로그램 작성
   - 프로그램 설명
   - 프로그램 소스
   - 실행 결과