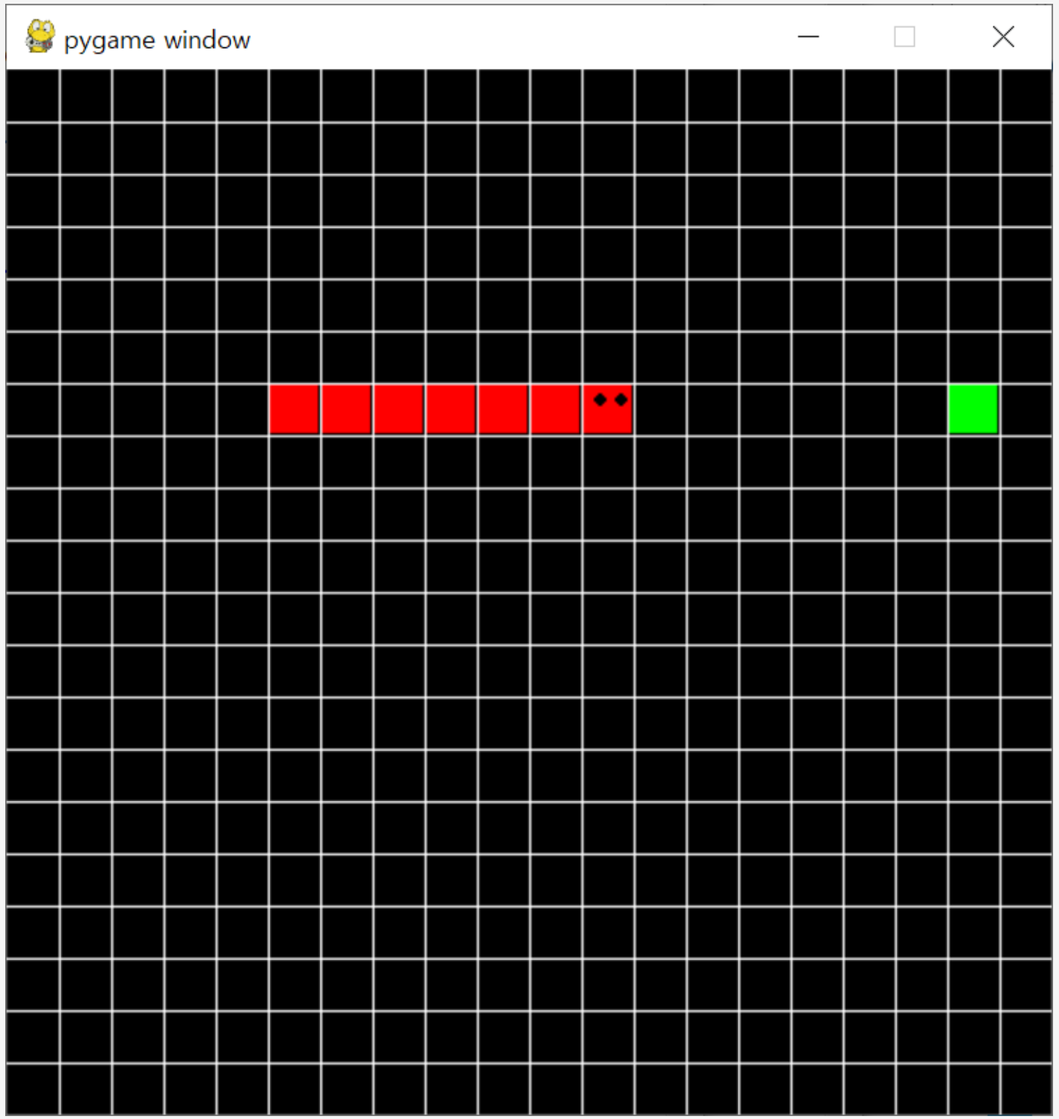
SNAKE GAME

김채원, 서현진, 허영윤

SNAKE GAME

기존 코드: https://github.com/techwithtim/Snake-Game

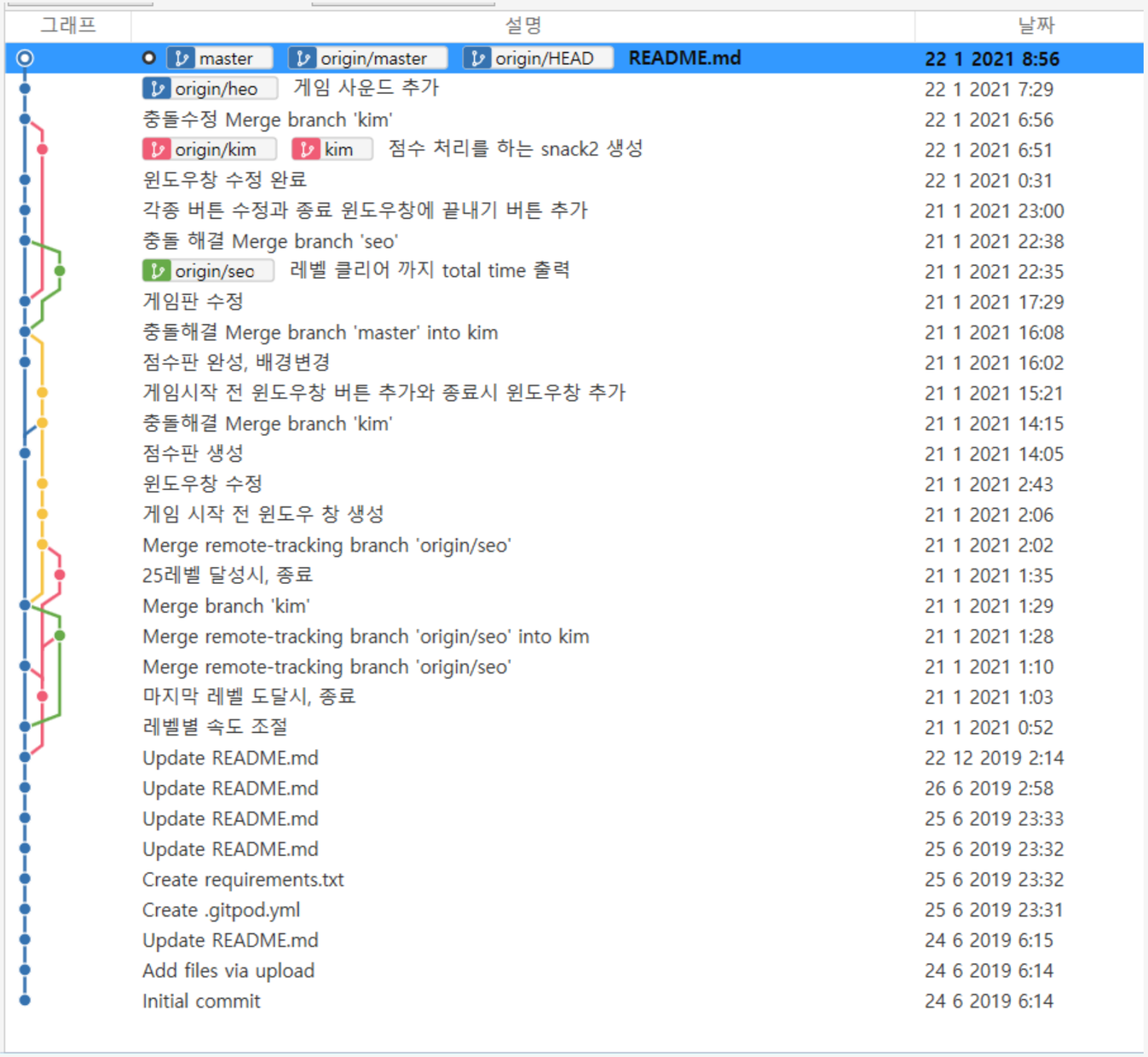
방향키로 조절하는 snake를 이용한 파이썬 게임.
화면에 랜덤으로 생성되는 snack 과 충돌하면 뱀의 body인 큐브가 추가된다.
조작 도중 뱀의 head가 자신의 body에 닿게 되거나
메인 게임 보드를 벗어나면 게임은 종료된다.
snack를 최대한 많이 먹어 body을 늘리는 것이 게임의 목표이다.

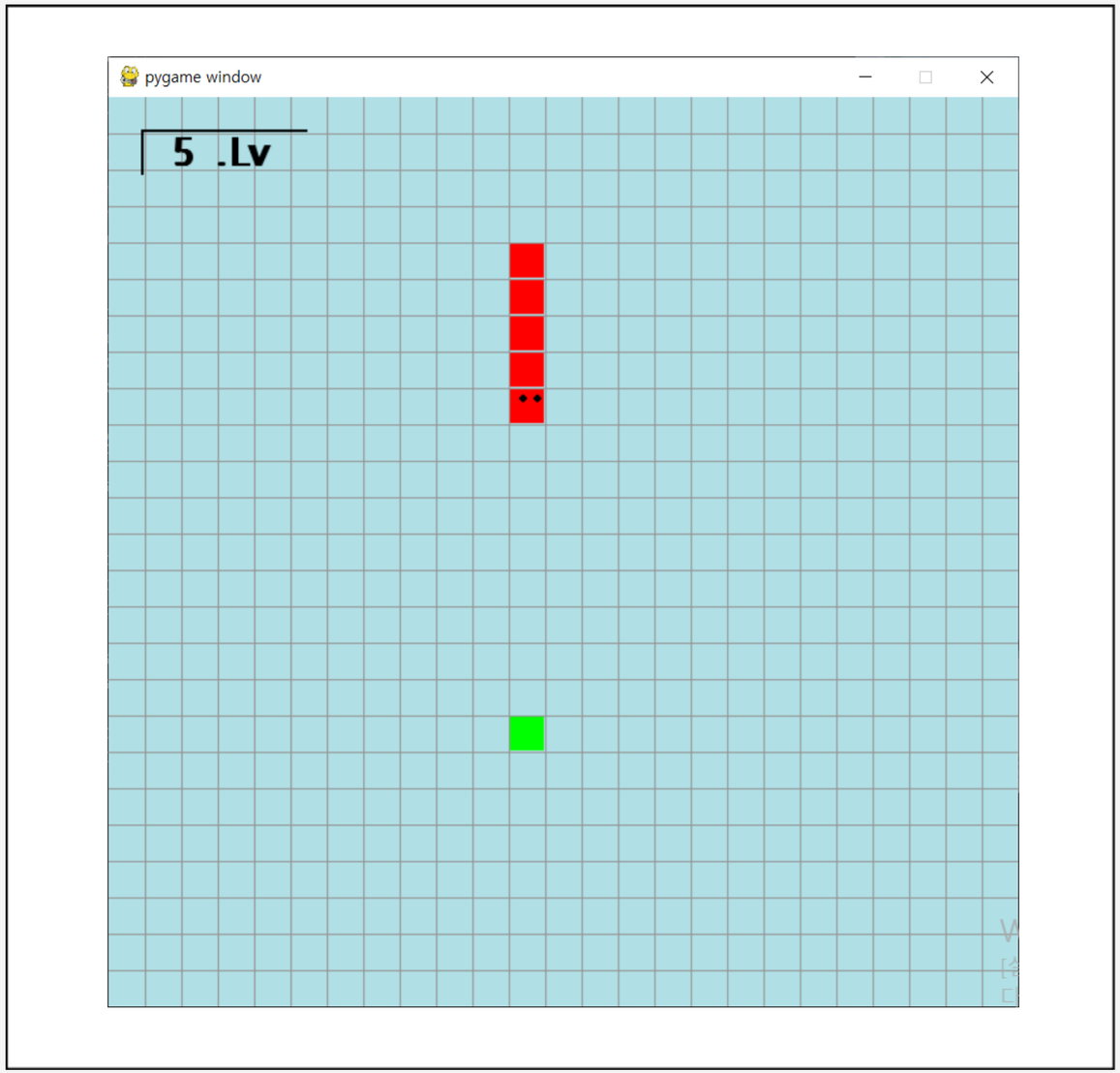


1. 게임 속도 조절
2. 최종 레벨 25로 달성 시 종료
3. 게임시작 윈도우 창 생성
4. 게임판에 레벨 출력
5. 최종 레벨 달성 시
   게임종료 윈도우 창 생성
6. 메인 게임보드 설정 조절
7. 종료시 게임 total time 출력
8. 점수가 있는 큐브 생성
9. 게임 사운드 추가

OSS 프로젝트 기말 발표

9조





1. 게임 속도 조절

기존 코드보다 속도를 늦추고,

5.Lv 씩 1.5배의 출력으로

속도를 빨라지게 설정함.

기존 코드

```
def move(self):
    for event in py
        def quit()
        quit() -> None uninitialize all pygame modules
        if event.ty
            pygame.quit()
        keys = pygame.key.get_pressed()

        for key in keys:
            if keys[pygame.K_LEFT]:
                self.dirnx = -1
                self.dirny = 0
                self.turns[self.head.pos[:]] = [self.dirnx,self.dirny]
            elif keys[pygame.K_RIGHT]:
                self.dirnx = 1
                self.dirny = 0
                self.turns[self.head.pos[:]] = [self.dirnx,self.dirny]
            elif keys[pygame.K_UP]:
                self.dirny = -1
                self.dirnx = 0
                self.turns[self.head.pos[:]] = [self.dirnx,self.dirny]
            elif keys[pygame.K_DOWN]:
                self.dirny = 1
                self.dirnx = 0
                self.turns[self.head.pos[:]] = [self.dirnx,self.dirny]
```

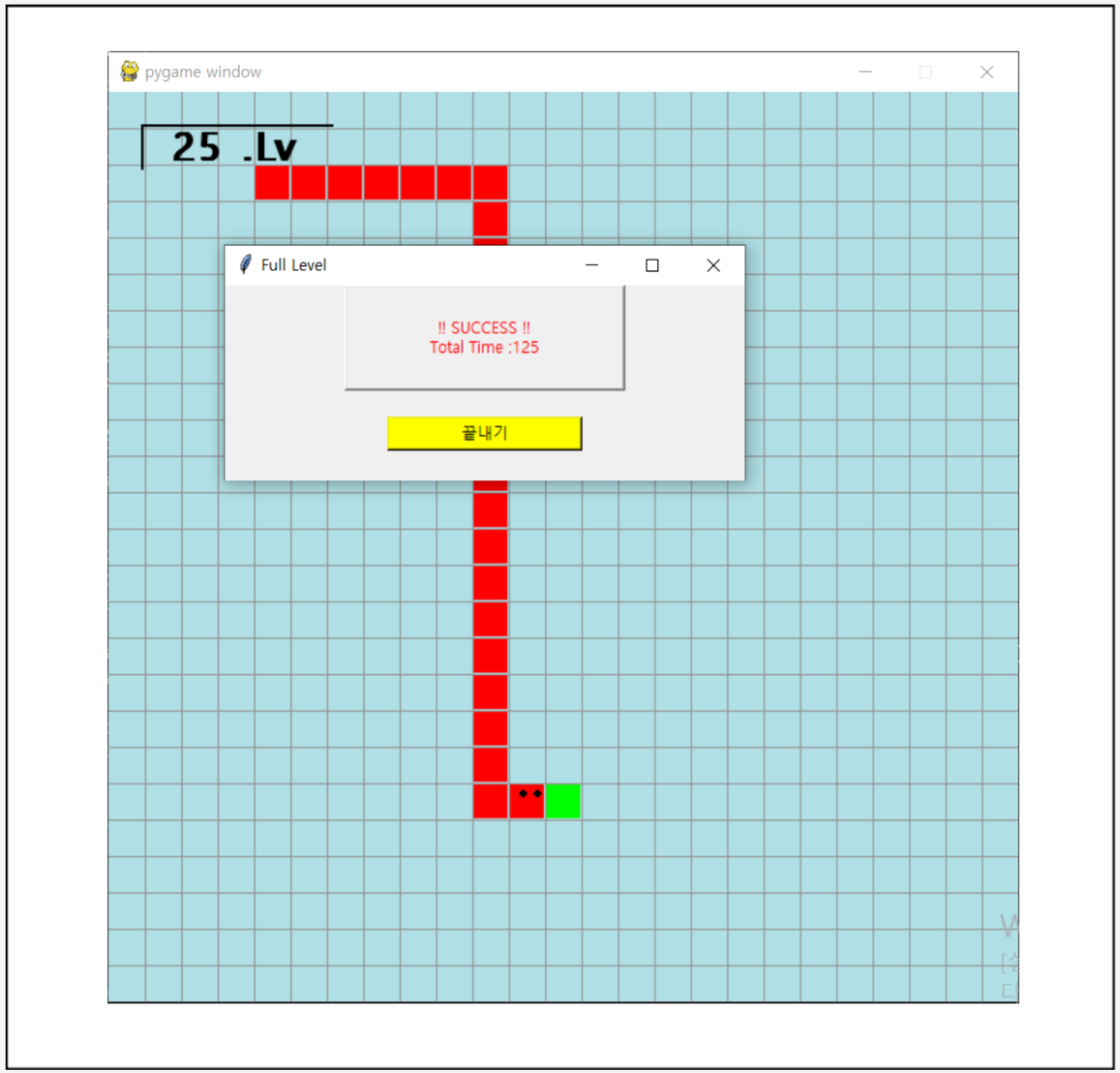
변경한 코드

```
14  pygame.init()
15  clock = pygame.time.Clock()
16  background = pygame.display.set_mode((width, height))
17
```

pygame초기화, clock초기화

```
109
110             level = int(len(self.body) / 5)
111             fps = 6 + (1.5*level)
112             clock.tick(fps)
```

class snake 안에 **move**함수 수정
level에서 5단계를 설정
fps로 기존 속도 6에서 단계별 1.5 설정
clock.tick() 함수로 속도 적용



2. 최종 레벨 25로 달성 시 종료

게임을 종료하는 함수를 정의하고

최고 레벨을 25로 설정하고

달성 시 종료 함수 실행

기존 코드

```
def main():
    global s, snack, win
    win = pygame.display.set_mode((width,height))
    s = snake((255,0,0), (10,10))
    s.addCube()
    snack = cube(randomSnack(rows,s), color=(0,255,0))
    flag = True
    clock = pygame.time.Clock()

    while flag:
        pygame.time.delay(50)
        clock.tick(10)
        s.move()
        headPos = s.head.pos
        if headPos[0] >= 20 or headPos[0] < 0 or headPos[1] >= 20 or headPos[1] < 0:
            print("Score:", len(s.body))
            s.reset((10, 10))

        if s.body[0].pos == snack.pos:
            s.addCube()
            snack = cube(randomSnack(rows,s), color=(0,255,0))

        for x in range(len(s.body)):
            if s.body[x].pos in list(map(lambda z:z.pos,s.body[x+1:])):
                print("Score:", len(s.body))
                s.reset((10,10))
                break

        redrawWindow()
```

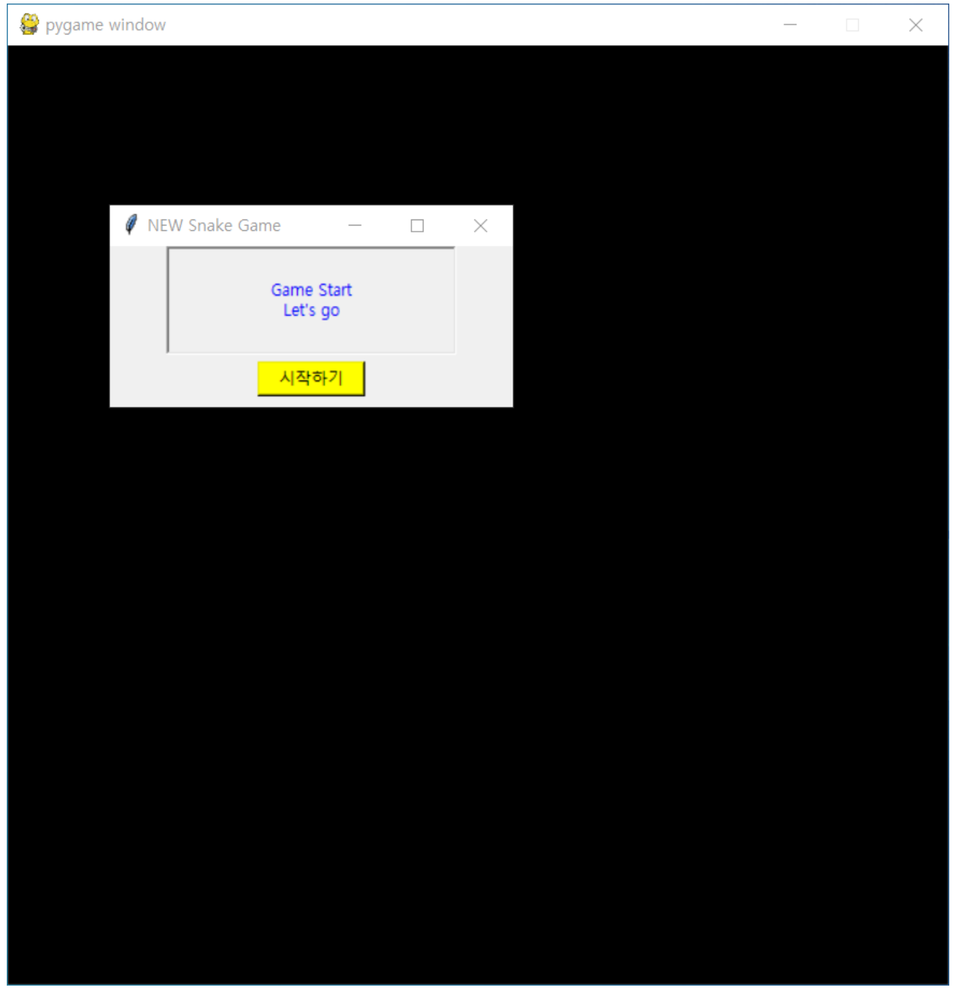
변경한 코드

```
144     def end_game():
145         print("Score:", len(self.body))
146         sys.exit(0)
```

class snake 안에 end_game함수 생성
sys.exit(0)에 의해 게임창이 닫힘

```
244             if len(s.body) > 25:
245                 finish_game(seconds)
246                 end_game()
```

기존 main함수에서 최종 레벨인
25Lv을 달성 시, 게임을 종료하는
end_game함수를 실행하는 코드 추가



3. 게임 시작 윈도우 창 생성

메인 게임 시작 전

‘Game Start Let’s go’ 라벨과

‘시작하기’ 버튼이 있는

윈도우 창 생성

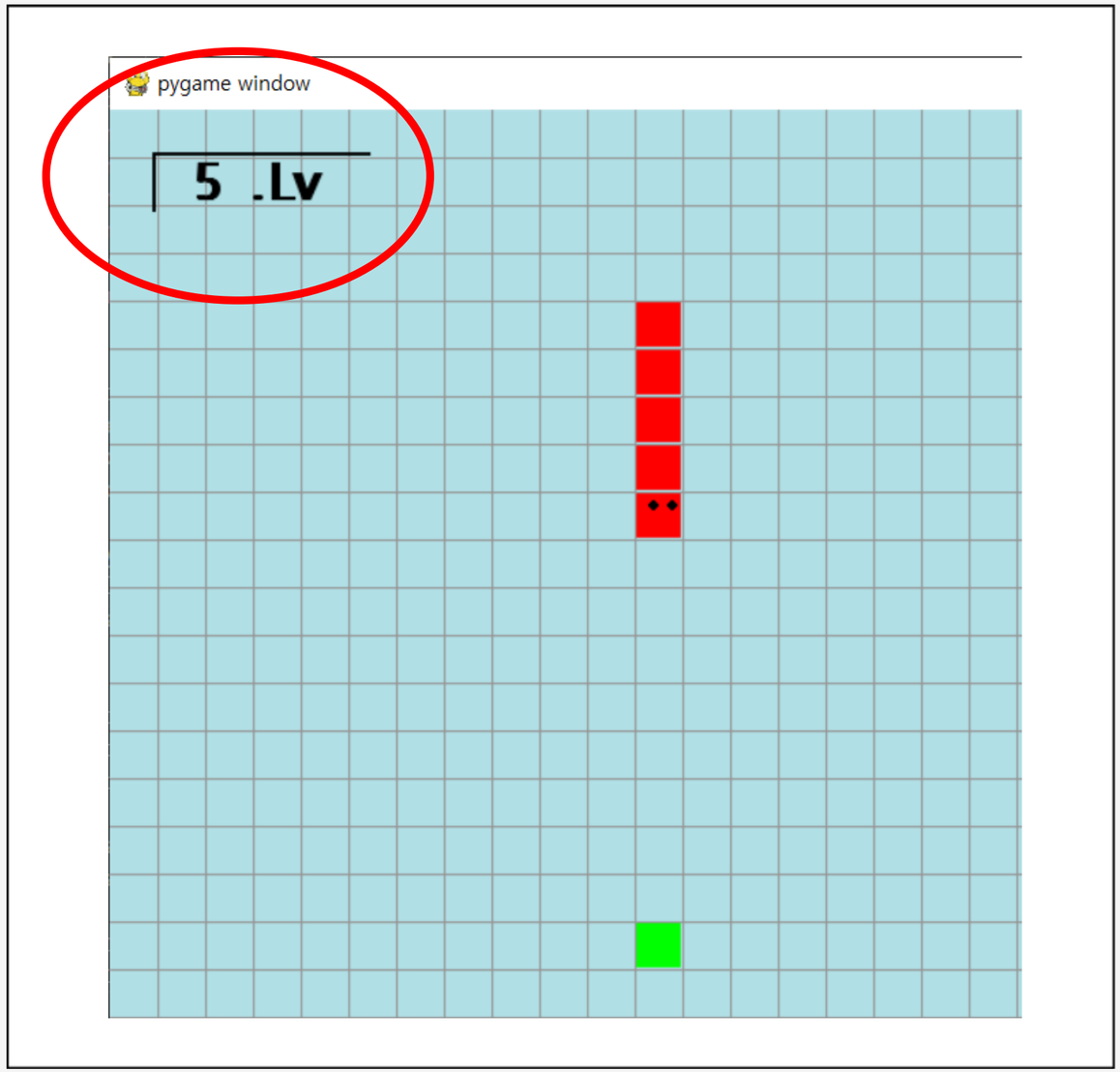
기존 코드

```
import math
import random
import pygame
import random
import tkinter as tk
from tkinter import messagebox

width = 500
height = 500

cols = 25
rows = 20


class cube():
```

변경한 코드

```
window=Tk()
window.title("NEW Snake Game")
window.geometry('300x120+470+200')

label1=Label(window, text="Game Start\nLet's go",width=30,height=5,fg="blue",relief="sunken")
label1.pack()

b1=Button(window,text="시작하기",width=10,bg='yellow',command = window.destroy)
b1.pack(padx = 10, pady = 5)

window.mainloop()
```

Tk 모듈을 통해
window라는 윈도우 창을 정의
window.mainloop()을 사용하여
윈도우 종료 될 때까지 실행



4. 게임판에 레벨 출력

메인 게임판 위에

뱀 길이에 따른 레벨이 담긴

show_info 함수를 정의하고

window에서 호출

기존 코드

```
def redrawWindow():
    global win
    win.fill((0,0,0))
    drawGrid(width, rows, win)
    s.draw(win)
    snack.draw(win)
    pygame.display.update()
    pass
```

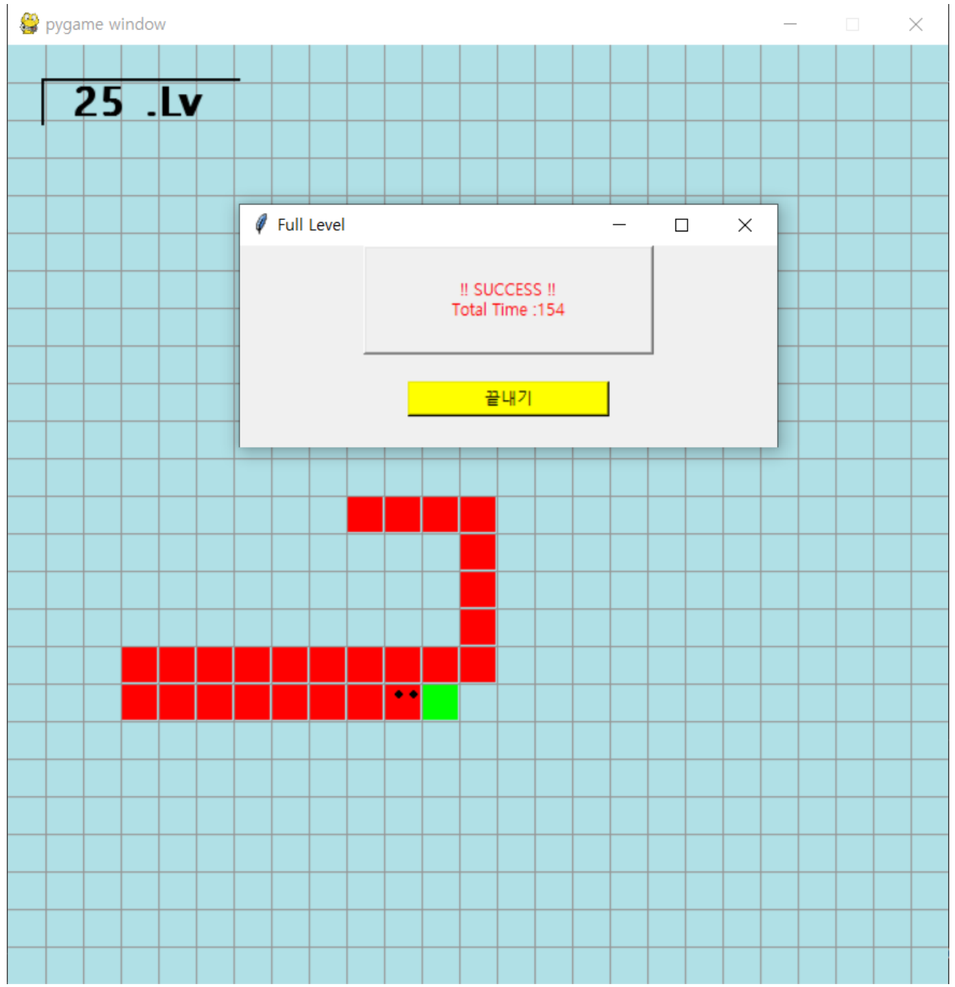
변경한 코드

```
def show_info(self):
    font = pygame.font.SysFont('malgungothic',30,bold=5)
    image = font.render(f' {len(self.body)} .Lv ', True, (0,0,0))
    pos = image.get_rect()
    pos.move_ip(20,20)
    pygame.draw.rect(image, (0,0,0),(pos.x-15, pos.y-15, pos.width, pos.height), 2)
    background.blit(image, pos)
```

class snake 안에 show_info함수 정의
뱀의 길이에 따라 .Lv를 조절하고 띄움

```
def redrawWindow():
    global win
    win.fill((176,224,230))
    drawGrid(width, rows, win)
    s.draw(win)
    snack.draw(win)
    s.show_info()
    pygame.display.update()
```

게임판 출력 함수에서 show_info를 호출



5. 최종 레벨 달성 시 게임 종료 윈도우 창 생성

게임 종료 윈도우 창을 생성하는 finish_game 함수를 정의하여 버튼을 통해 창을 종료

기존 코드

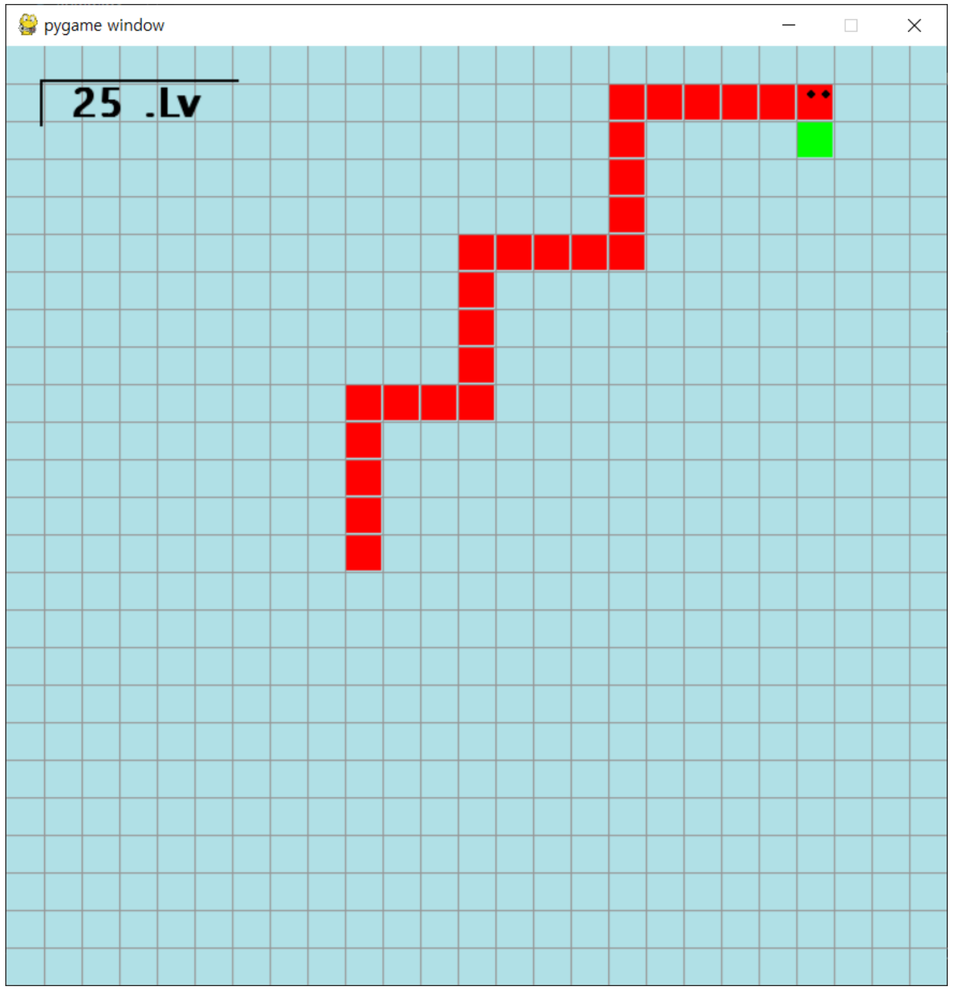
```
117     def draw(self, surface):
118         for i,c in enumerate(self.body):
119             if i == 0:
120                 c.draw(surface, True)
121             else:
122                 c.draw(surface)
123
124     # finish_game 함수 생성
125
126     def redrawWindow():
127         global win
128         win.fill((0,0,0))
129         drawGrid(width, rows, win)
130         s.draw(win)
131         snack.draw(win)
132         pygame.display.update()
133         pass
```

변경한 코드

```
def finish_game(seconds):
    fin=Tk()
    fin.title("Full Level")
    fin.geometry('400x150+500+200')
    b=int(seconds)
    print ("Total Time :", b)
    label2=Label(fin, text="!! SUCCESS !!\nTotal Time :"+str(b),width=30,height=5,fg="red",relief="raised")
    label2.pack()

    b2=Button(fin, text="끝내기",width=20,bg='yellow',command = fin.destroy)
    b2.pack(padx = 10, pady = 20)

    fin.mainloop()
```

```
if len(s.body) > 25:
    finish_game()
    end_game()

redrawWindow()
```

finish_game 함수를 정의하여 fin 윈도우 창에 라벨과 버튼을 생성하여 띄우게 함.
25레벨 달성 시 종료되는 코드안에서 호출됨.



6. 메인 게임보드 설정 조절

메인 게임보드의 크기를 키우고

전체 격자의 수를 늘림

메인 게임보드를 충돌 시 사망

설정 조절

기존 코드

```
def redrawWindow():
    global win
    win.fill((0,0,0))
    drawGrid(width, rows, win)
    s.draw(win)
    snack.draw(win)
    pygame.display.update()
    pass
```

```
def drawGrid(w, rows, surface):
    sizeBtwn = w // rows

    x = 0
    y = 0
    for l in range(rows):
        x = x + sizeBtwn
        y = y +sizeBtwn

        pygame.draw.line(surface, (255,255,255), (x, 0),(x,w))
        pygame.draw.line(surface, (255,255,255), (0, y),(w,y))
```

```
if headPos[0] >= 20 or headPos[0] < 0 or headPos[1] >= 20 or headPos[1] < 0:
    print("Score:", len(s.body))
    s.reset((10, 10))
```

변경한 코드

```
def redrawWindow():
    global win
    win.fill((176,224,230))
    drawGrid(width, rows, win)
    s.draw(win)
    snack.draw(win)
    s.show_info()
    pygame.display.update()
```

배경 색상 변경

```
width = 700
height = 700
cols = 25
rows = 25
```

```
class cube():
    rows = 25
    w = 700
```

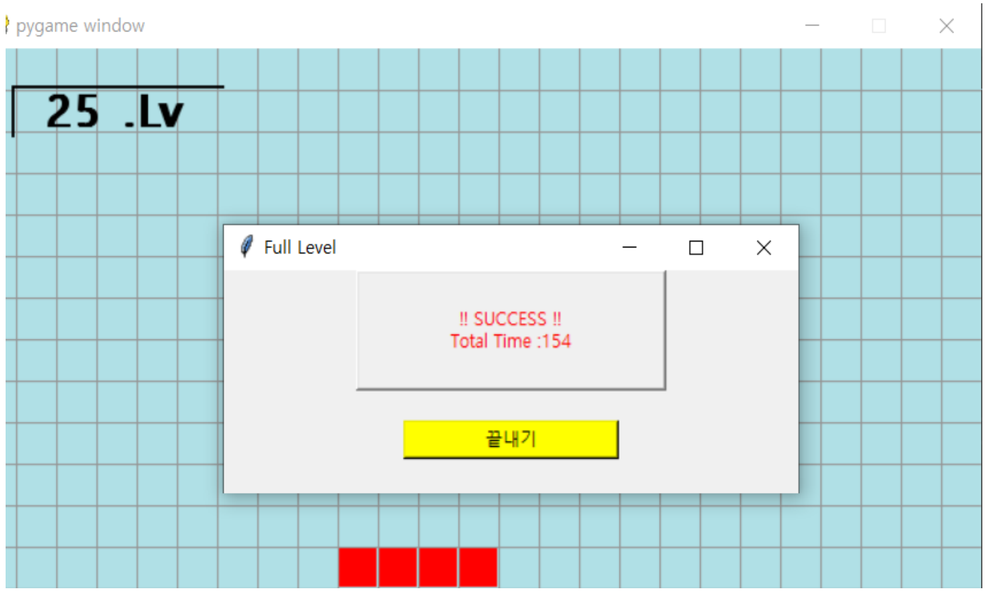
변수 크기 변경

```
def drawGrid(w, rows, surface):

    sizeBtwn = w // rows
    x = 0
    y = 0

    for l in range(rows):
        x = x + sizeBtwn
        y = y +sizeBtwn

        pygame.draw.line(surface, (150,150,150), (x, 0),(x,w))
        pygame.draw.line(surface, (150,150,150), (0, y),(w,y))
```

gird 라인 색상 변경

```
if headPos[0] >= 25 or headPos[0] < 0 or headPos[1] >= 25 or headPos[1] < 0:
```

메인 게임 보드 충돌 설정 변경



7. 종료시 게임 total time 출력

최종 25레벨 달성 시 종료 창에

레벨 달성까지 걸린 시간 출력

기존 코드

```
def main():
    global s, snack, win
    win = pygame.display.set_mode((width,height))
    s = snake((255,0,0), (10,10))
    s.addCube()
    snack = cube(randomSnack(rows,s), color=(0,255,0))
    flag = True
    clock = pygame.time.Clock()
```

```
while flag:
    pygame.time.delay(50)
    clock.tick(10)
    s.move()
    headPos = s.head.pos
    if headPos[0] >= 20 or headPos[0] < 0 or headPos[1] >= 20 or headPos[1] < 0:
        print("Score:", len(s.body))
        s.reset((10, 10))
```

```
121                 else:
122                     c.draw(surface)
123
124     # finish_game 함수 생성
125
126     def redrawWindow():
127         global win
128         win.fill((0,0,0))
129         drawGrid(width, rows, win)
```

변경한 코드

```
219     start_ticks=pygame.time.get_ticks()
220
```

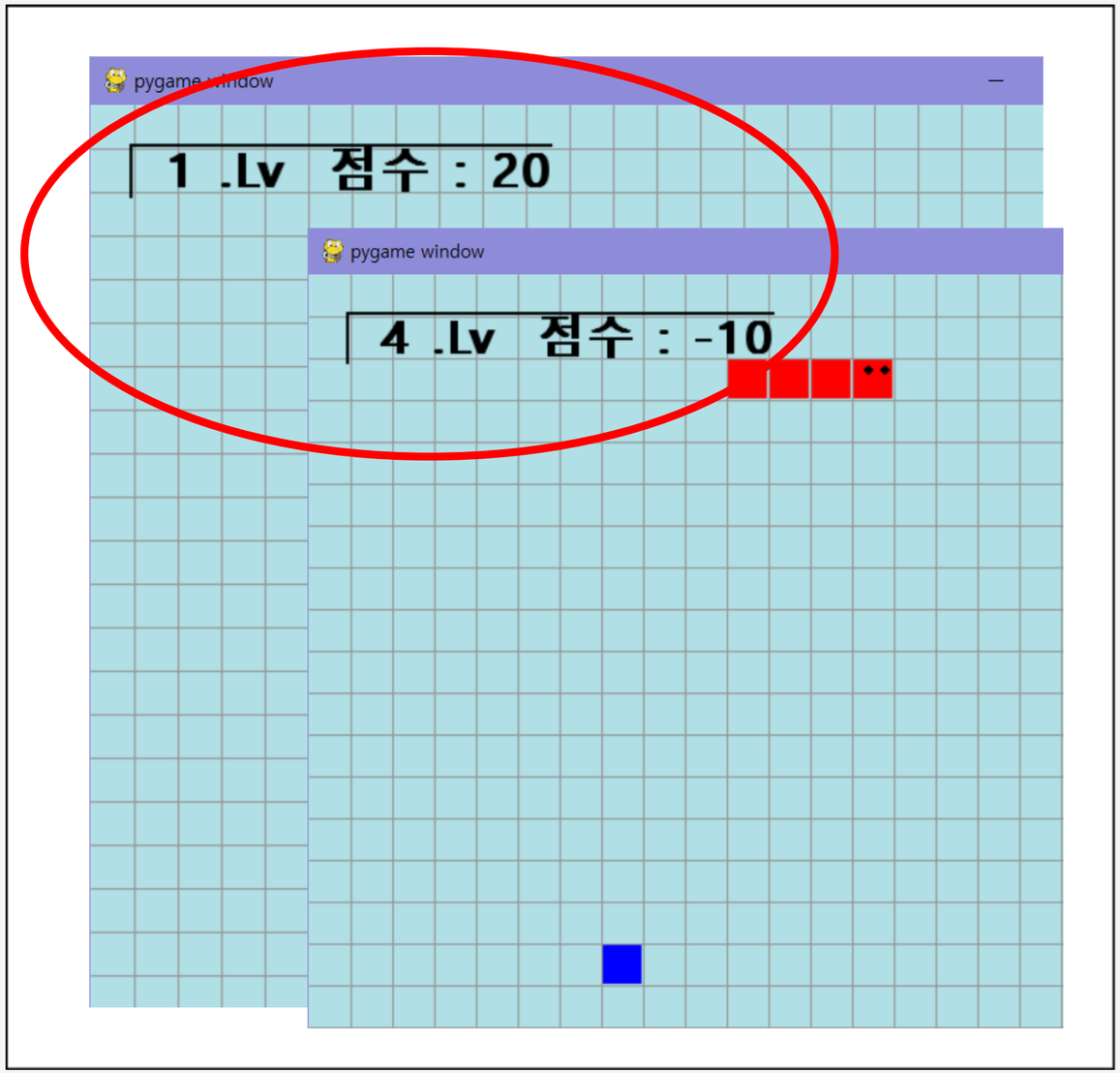
main함수가 호출된 시점의 시작시간을 tick을 통해 받아 시작 시간 start_ticks를 정의

```
227         seconds=(pygame.time.get_ticks()-start_ticks)/1000
```

while문 안에 pygame이 호출된 이후 흐르는 시간을 리턴하는 함수를 이용해 경과 시간 계산

```
b=int(seconds)
print ("Total Time :", b)
label2=Label(fin, text="!! SUCCESS !!\nTotal Time :"+str(b),width=30,height=5,fg="red",relief="raised")
label2.pack()
```

finish_game함수에서 total time 출력



8. 점수가 있는 큐브 생성

snake와 충돌 시 점수를 내는

bug(파),snack(초) 변수를

만들고 메인 게임 보드에 출력

기존 코드

```
def main():
    global s, snack, win
    win = pygame.display.set_mode((width,height))
    s = snake((255,0,0), (10,10))
    s.addCube()
    snack = cube(randomSnack(rows,s), color=(0,255,0))
    flag = True
    clock = pygame.time.Clock()
```

```
if s.body[0].pos == snack.pos:
    s.addCube()
    snack = cube(randomSnack(rows,s), color=(0,255,0))
```

변경한 코드

```
bug = cube(randomSnack(rows,s), color=(0,0,255))
```

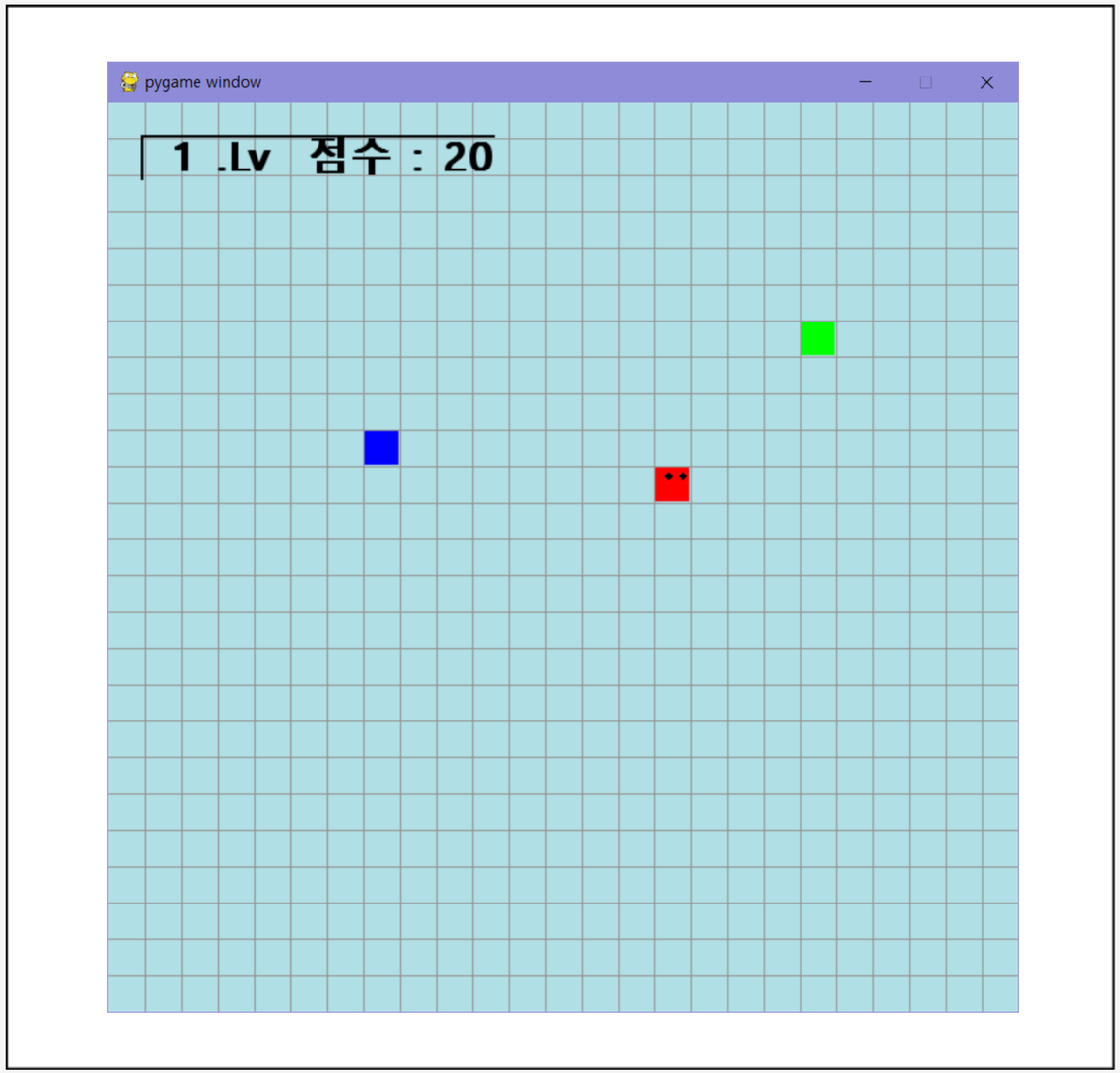
랜덤큐브을 생성하는 변수 만들기

```
if s.body[0].pos == snack.pos:
    s.addCube()
    snack = cube(randomSnack(rows,s), color=(0,255,0))
    bug = cube(randomSnack(rows,s), color=(0,0,255))
    score +=10
    redrawWindow()

if s.body[0].pos == bug.pos:
    snack = cube(randomSnack(rows,s), color=(0,255,0))
    bug = cube(randomSnack(rows,s), color=(0,0,255))
    score -= 10
    redrawWindow()
```

snack,bug 변수와 snake의 head가 충돌 시 점수를 계산

```
def show_info(self,score):
    font = pygame.font.SysFont('malgungothic',30,bold=5)
    image = font.render(f' {len(self.body)} .Lv 점수 : {score}', True, (0,0,0))
```

계산한 점수를 메인게임보드에 출력



9. 게임 사운드 추가

snake가 snack과 bug를 만났을 때와 메인게임보드 밖으로 나갈시 사운드 출력

기존 코드

```
def main():
    global s, snack, win
    win = pygame.display.set_mode((width,height))
    s = snake((255,0,0), (10,10))
    s.addCube()
    snack = cube(randomSnack(rows,s), color=(0,255,0))
    flag = True
    clock = pygame.time.Clock()
```

original코드

8번까지
변경한 코드

```
def main():

    global s, snack,bug, win, b, score

    win = pygame.display.set_mode((width,height))
    s = snake((255,0,0), (10,10))
    s.addCube()
    snack = cube(randomSnack(rows,s), color=(0,255,0))
    bug = cube(randomSnack(rows,s), color=(0,0,255))
    flag = True
    clock = pygame.time.Clock()
    start_ticks=pygame.time.get_ticks()

    score=0
```

변경한 코드

```
227     eatsnack = pygame.mixer.Sound('grow.wav')
228     eatsnack.set_volume(0.8)
229     eatbug = pygame.mixer.Sound('shrink.wav')
230     eatbug.set_volume(0.8)
231     gameoversound = pygame.mixer.Sound('gameover.wav')
232     gameoversound.set_volume(0.8)
```

main 함수 안에 Sound 객체를 만든다

```
243                 gameoversound.play()
248                     eatsnack.play()
256                     eatbug.play()
```

만들어진 객체를 필요한곳에 넣고
객체명.play()로 사운드 출력

감사합니다

김채원, 서현진, 허영윤

Github : https://github.com/cloudisme99/Snake-Game