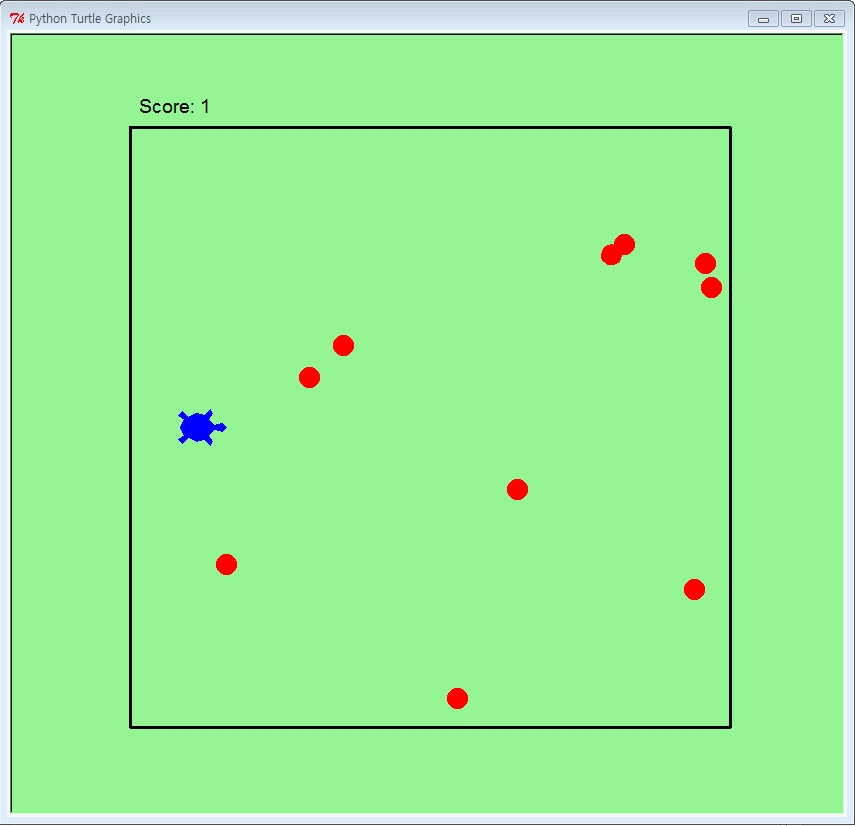
프로그래밍

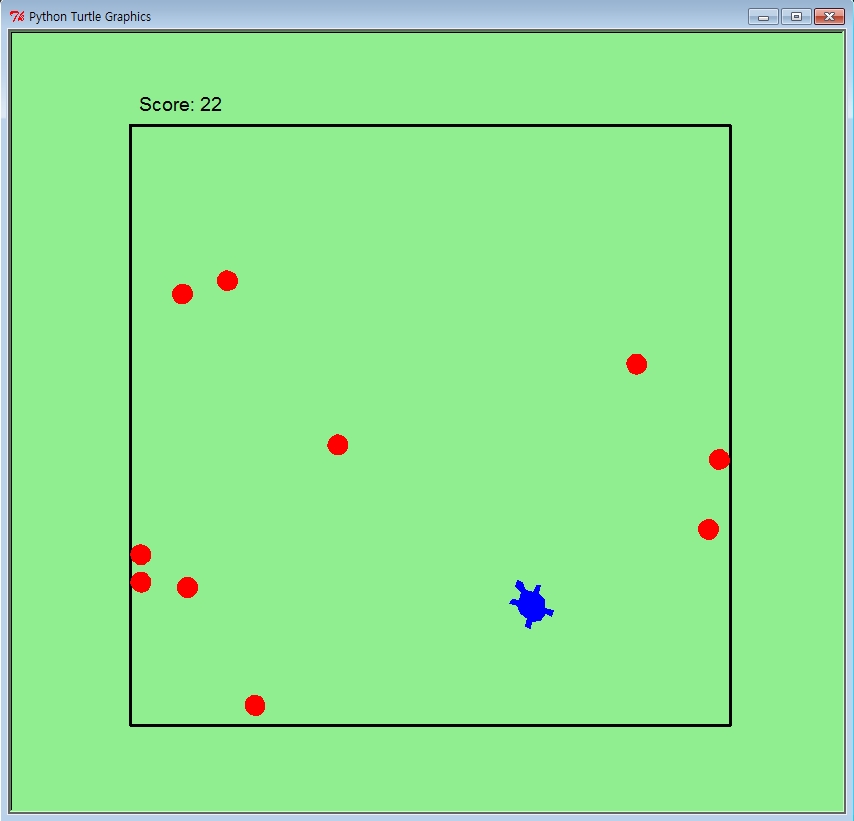
미디어소프트웨어 200911208 김종인

**목차**

1. **실행화면 및 설명**
2. **요구사항**
3. **구현내용**
4. **실행코드**

**1. 실행화면 및 설명**

  
거북이(파란색)가 거북이의 먹이(빨간색, Points)를 먹으면 왼쪽상단의 Score가 올라간다.

  
키보드 입력을 이용해 거북이의 이동방향 및 속도를 변경한다.

**2. 요구사항  
 1) 사용자가 플레이를 하기전 게임에 대한 설정.**

사용자가 플레이할 거북이의 크기 및 색을 정한다.

먹이(Point)의 색과 크기, 그리고 그 위치를 정한다.

사용자가 플레이 할 공간을 만들고 그 밖을 나가지 못하게 한다.

**2) 사용자가 플레이를 하면서 일어나는 변화를 설정**

사용자로 하여금 먹이(Point)를 먹었을 때의 점수 변화를 기록한다.

사용자가 Point를 획득 했을 경우 일어나는 변화를 표현한다.

사용자로 하여금 거북이의 움직임을 기존의 배운 것들을 응용해 표현한다.

사용자가 플레이 할 거북이 및 먹이가 플레이 할 공간 밖으로 나가지 못하게 하는

경계선과 만났을 때의 변화를 표현한다.

**3.구현내용**

**#거북이 게임**

**#import random 은 point의 위치를 랜덤하게 나타내기 위함.**

import turtle

import math

import random

**#거북이를 화면에 표시한다.**

wn=turtle.Screen()

wn.bgcolor("lightgreen")

**#프레임을 낮춰서 게임을 부드럽게 하기 위함.**

wn.tracer(3)

**#거북이가 플레이할 공간의 경계선(stadium)을 만들기를 시작한다.**

stadium = turtle.Turtle()

stadium.penup()

stadium.setposition(-300,-300)

stadium.pendown()

stadium.pensize(3)

**# stadium의 사각형 공간을 만듬.**

for side in range(4):

stadium.forward(600)

stadium.left(90)

stadium.hideturtle()

**# 사용자가 플레이를 할 거북이를 만듬**

player = turtle.Turtle()

player.color("blue")

player.shape("turtle")

player.shapesize(2,2)

player.penup()

player.speed(0)

**#점수를 기록하기 위한 score를 생성**

score = 0

**#거북이 먹이(point)를 만들어 리스트에 넣음**

maxPoints = 10

points = []

**#획득할 먹이의 모양과 나타날 위치를 랜덤으로 지정**

for count in range(maxPoints):

points.append(turtle.Turtle())

points[count].color("red")

points[count].shape("circle")

points[count].penup()

points[count].speed(0)

points[count].setposition(random.randint(-300,300), random.randint(-300,300 ))

**# 기본 속도 speed를 지정**

speed = 1

**# 키보드입력으로부터 거북이를 움직이게 할 함수들을 만듬**

def turnleft():

player.left(30)

def turnright():

player.right(30)

**# 키보드의 Up버튼을 누르면 거북이의 속도가 점차 증가함.**

def increasespeed():

global speed

speed +=1

**#math를 이용해 거북이가 포인트를 먹었을 때의 변화를 설정하는 함수**

def isCollision(t1,t2):

d = math.sqrt(math.pow(t1.xcor()-t2.xcor(),2) +

math.pow(t1.ycor()-t2.ycor(),2))

if d <20:

return True

else:

return False

**#키보드 입력으로 거북이를 움직임**

turtle.listen()

turtle.onkey(turnleft,"Left")

turtle.onkey(turnright,"Right")

turtle.onkey(increasespeed,"Up")

**#거북이(player)가 벽에 부딪히면 뒤를 돌아보게 설정**

while True:

player.forward(speed)

if player.xcor() > 300 or player.xcor() < -300:

player.right(180)

if player.ycor() > 300 or player.ycor() < -300:

player.right(180)

**#거북이 먹이를 움직이게 만듬**

for count in range(maxPoints):

points[count].forward(3)

**#먹이도 벽에 부딪히면 뒤를 돌아봄**

if points[count].xcor() > 290 or points[count].xcor() < -290:

points[count].right(180)

if points[count].ycor() > 290 or points[count].ycor() < -290:

points[count].right(180)

**#거북이가 먹이를 먹었을 때 다시 나타날 장소를 랜덤으로 지정한다.**

if isCollision(player,points[count]):

points[count].setposition(random.randint(-300,300), random.randint(-300,300 ))

points[count].right(random.randint(0,360))

score += 1

#좌측 상단에 점수를 기록

stadium.undo()

stadium.penup()

stadium.hideturtle()

stadium.setposition(-290,310)

scorestring = "Score: %s" %score

stadium.write(scorestring, False, align = "left", font =("Arial",14,"normal"))

**4. 실행코드**

|  |
| --- |
| #거북이게임 |
|  | import turtle |
|  | import math |
|  | import random |
|  |  |
|  |  |
|  | wn=turtle.Screen() |
|  | wn.bgcolor("lightgreen") |
|  |  |
|  | #프레임을 낮춰서 게임을 부드럽게 함 |
|  | wn.tracer(3) |
|  |  |
|  |  |
|  |  |
|  | #사각형 게임공간을 만듬 |
|  | stadium = turtle.Turtle() |
|  | stadium.penup() |
|  | stadium.setposition(-300,-300) |
|  | stadium.pendown() |
|  | stadium.pensize(3) |
|  | for side in range(4): |
|  | stadium.forward(600) |
|  | stadium.left(90) |
|  | stadium.hideturtle() |
|  |  |
|  |  |
|  |  |
|  |  |
|  | #플레이를 할 거북이를 만듬 |
|  | player = turtle.Turtle() |
|  | player.color("blue") |
|  | player.shape("turtle") |
|  | player.shapesize(2,2) |
|  | player.penup() |
|  | player.speed(0) |
|  |  |
|  | #점수 |
|  | score = 0 |
|  |  |
|  | #거북이 먹이(포인트)를 만들어 리스트에 넣음 |
|  | maxPoints = 10 |
|  | points = [] |
|  |  |
|  |  |
|  | #획득할 먹이의 모양과 나타날 위치를 랜덤으로 지정 |
|  | for count in range(maxPoints): |
|  | points.append(turtle.Turtle()) |
|  | points[count].color("red") |
|  | points[count].shape("circle") |
|  | points[count].penup() |
|  | points[count].speed(0) |
|  | points[count].setposition(random.randint(-300,300), random.randint(-300,300 )) |
|  |  |
|  |  |
|  | #기본 속도 지정 |
|  | speed = 1 |
|  |  |
|  | #키보드입력으로부터 거북이를 움직이게 할 함수들을 지정 |
|  |  |
|  | def turnleft(): |
|  | player.left(30) |
|  |  |
|  | def turnright(): |
|  | player.right(30) |
|  |  |
|  | def increasespeed(): |
|  | global speed |
|  | speed +=1 |
|  |  |
|  | def isCollision(t1,t2): |
|  | #math를 이용해 거북이가 포인트를 먹었을 때의 변화를 설정하는 함수 |
|  | d = math.sqrt(math.pow(t1.xcor()-t2.xcor(),2) + |
|  | math.pow(t1.ycor()-t2.ycor(),2)) |
|  | if d <20: |
|  | return True |
|  | else: |
|  | return False |
|  |  |
|  |  |
|  | #키보드 입력으로 거북이를 움직임 |
|  | turtle.listen() |
|  | turtle.onkey(turnleft,"Left") |
|  | turtle.onkey(turnright,"Right") |
|  | turtle.onkey(increasespeed,"Up") |
|  |  |
|  |  |
|  |  |
|  | while True: |
|  | player.forward(speed) |
|  |  |
|  | #거북이(player)가 벽에 부딪히면 뒤를 돌아보게 설정 |
|  | if player.xcor() > 300 or player.xcor() < -300: |
|  | player.right(180) |
|  |  |
|  | if player.ycor() > 300 or player.ycor() < -300: |
|  | player.right(180) |
|  |  |
|  |  |
|  |  |
|  | #거북이 먹이를 움직이게 만듬 |
|  | for count in range(maxPoints): |
|  | points[count].forward(3) |
|  |  |
|  | #먹이도 벽에 부딪히면 뒤를 돌아봄 |
|  | if points[count].xcor() > 290 or points[count].xcor() < -290: |
|  | points[count].right(180) |
|  |  |
|  | if points[count].ycor() > 290 or points[count].ycor() < -290: |
|  | points[count].right(180) |
|  |  |
|  | #거북이가 먹이를 먹었을 때 다시 나타날 장소를 랜덤으로 지정 |
|  | if isCollision(player,points[count]): |
|  | points[count].setposition(random.randint(-300,300), random.randint(-300,300 )) |
|  | points[count].right(random.randint(0,360)) |
|  | score += 1 |
|  | #좌측 상단에 점수를 기록 |
|  | stadium.undo() |
|  | stadium.penup() |
|  | stadium.hideturtle() |
|  | stadium.setposition(-290,310) |
|  | scorestring = "Score: %s" %score |
|  | stadium.write(scorestring, False, align = "left", font =("Arial",14,"normal")) |