1. Cocos 예제 (chapter2\_01.py) 수정

- 4개의 NPC들은 아래와 같은 동작을 수행

. 임의의 고유 속력을 가지고 있으며, 진행 방향(초기에 임의의 진행 방향을 가짐)은 약간씩 수정된다.

. 좌/우 화면을 벗어나면 반대편 우/좌 화면을 나타난다 (상/하도 동일)

. 항상 4개의 NPC가 존재하도록, player와 충돌한 NPC가 발생하면 새로운 NPC를 추가한다.

|  |
| --- |
| Description]  NPC가 player와 충돌하면 새로운 NPC가 화면의 테두리 4변의 중심 중 하나에서 랜덤하게 생성된다.  모든 구체의 기본 속력은 300이고, 새로 생성된 NPC는 200~300 사이의 랜덤한 속력을 갖는다.  NPC의 진행 방향은 매 update마다 랜덤한 값을 더해주어 조금씩 변한다.  구체의 중심이 화면의 좌/우를 벗어나면 반대편 우/좌 화면에서 나타난다(상/하도 동일함) |
| Code]  import cocos import cocos.collision\_model as cm import cocos.euclid as eu import random import math  from collections import defaultdict from pyglet.window import key   class Actor(cocos.sprite.Sprite):  def \_\_init\_\_(self, x, y, color):  super(Actor, self).\_\_init\_\_('ball.png', color=color)  self.position = pos = eu.Vector2(x, y)  self.cshape = cm.CircleShape(pos, self.width/2)  self.velocity = self.normalization((random.uniform(-1, 1), random.uniform(-1, 1)))  self.speed = 300.0   def change\_speed(self, s):  self.speed = s   def change\_velocity(self, v):  v = self.normalization(v)  self.velocity = v   def normalization(self, v):  if v[0] is not 0 or v[1] is not 0:  magnitude = math.sqrt(v[0]\*\*2 + v[1]\*\*2)  v = (v[0]/magnitude, v[1]/magnitude)  return v  return (0, 0)   def move(self, dt):  pos = self.position  new\_x = pos[0] + self.speed \* self.velocity[0] \* dt  new\_y = pos[1] + self.speed \* self.velocity[1] \* dt  self.position = (new\_x, new\_y)  self.cshape.center = eu.Vector2(self.position[0], self.position[1])   class MainLayer(cocos.layer.Layer):  is\_event\_handler = True   def \_\_init\_\_(self):  super(MainLayer, self).\_\_init\_\_()  self.player = Actor(320, 240, (0, 0, 255))  self.player.change\_velocity((0, 0))  self.add(self.player)  for pos in [(100, 100), (540, 380), (540, 100), (100, 380)]:  self.add(Actor(pos[0], pos[1], (255, 0, 0)))   cell = self.player.width \* 1.25  self.collman = cm.CollisionManagerGrid(0, 640, 0, 480,  cell, cell)  self.pressed = defaultdict(int)  self.schedule(self.update)   def on\_key\_press(self, k, m):  self.pressed[k] = 1   def on\_key\_release(self, k, m):  self.pressed[k] = 0   def update(self, dt):  self.collman.clear()  for \_, node in self.children:  self.collman.add(node)  for other in self.collman.iter\_colliding(self.player):  self.remove(other)  pos = random.choice([(320, 0), (320, 480), (0, 240), (680, 240)])  npc = Actor(pos[0], pos[1], (255, 0, 0))  npc.change\_speed(random.randint(200, 300))  self.add(npc)   x = self.pressed[key.RIGHT] - self.pressed[key.LEFT]  y = self.pressed[key.UP] - self.pressed[key.DOWN]  self.player.change\_velocity((x, y))   for \_, node in self.children:  node.move(dt)  node.change\_velocity((node.velocity[0]+random.uniform(-0.1, 0.1), node.velocity[1]+random.uniform(-0.1, 0.1)))  if node.position[0] < 0:  node.position = (640, node.position[1])  if node.position[0] > 640:  node.position = (0, node.position[1])  if node.position[1] < 0:  node.position = (node.position[0], 480)  if node.position[1] > 480:  node.position = (node.position[0], 0)   if \_\_name\_\_ == '\_\_main\_\_':  cocos.director.director.init(caption='Hello, Cocos')  layer = MainLayer()  scene = cocos.scene.Scene(layer)  cocos.director.director.run(scene) |
| Game Shot]  컴퓨터, 실내, 노트북, 앉아있는이(가) 표시된 사진  자동 생성된 설명 |