1. 공을 ball2.png로 변경

- tkinter.PhotoImage, Canvas.create\_image 사용

- 그림의 Canvas.coords의 결과는 중심 좌표

. 이전 소스에서 사용된 oval의 coords는 left, top, right, bottom

- 방법1)

. 좌표 사용되는 위치에서 모두 수정

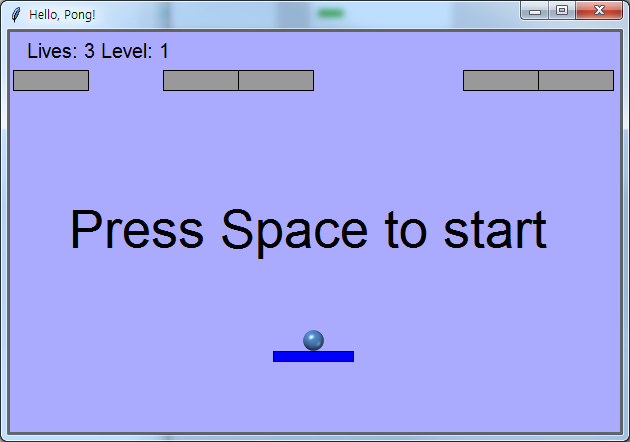
. Ball.\_\_init\_\_, Ball.update, Ball.collide, Game.check\_collisions 수정

- 방법2)

. Ball.get\_position을 overriding

|  |
| --- |
| Description]  get\_position 함수를 overriding함 |
| Code]  def \_\_init\_\_(self, canvas, x, y):  self.radius = 10  self.direction = [1, -1]  self.speed = 10  self.ballimage = tk.PhotoImage(file='ball2.png') # PhotoImage를 멤버변수로 저장하지 않으면 사진이 할당해제되어 이미지가 보이지 않음  image = canvas.create\_image(x, y, anchor=tk.CENTER, image=self.ballimage)  super(Ball, self).\_\_init\_\_(canvas, image  def get\_position(self):  center = self.canvas.coords(self.item)  left = center[0] - self.radius  right = center[0] + self.radius  top = center[1] - self.radius  bottom = center[1] + self.radius  return [left, top, right, bottom] |
| Game Shot] |

2. Level 추가, 벽돌 층수 변경, 임의의 벽돌만 생성



- Game.\_\_int\_\_에서 self.level = 1로 초기화

- Game.game\_loop에서 num\_bricks == 0인 경우에 level 증가, <space> binding 등의 추가 필요

- Game.update\_lives\_text에 level 표시: ‘%d %d’ % (x, y)

- Level 시작할 때, 벽돌 생성을 함수로 변경

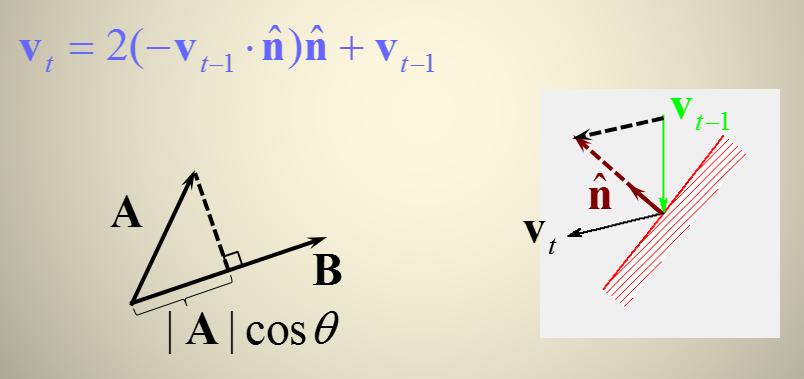
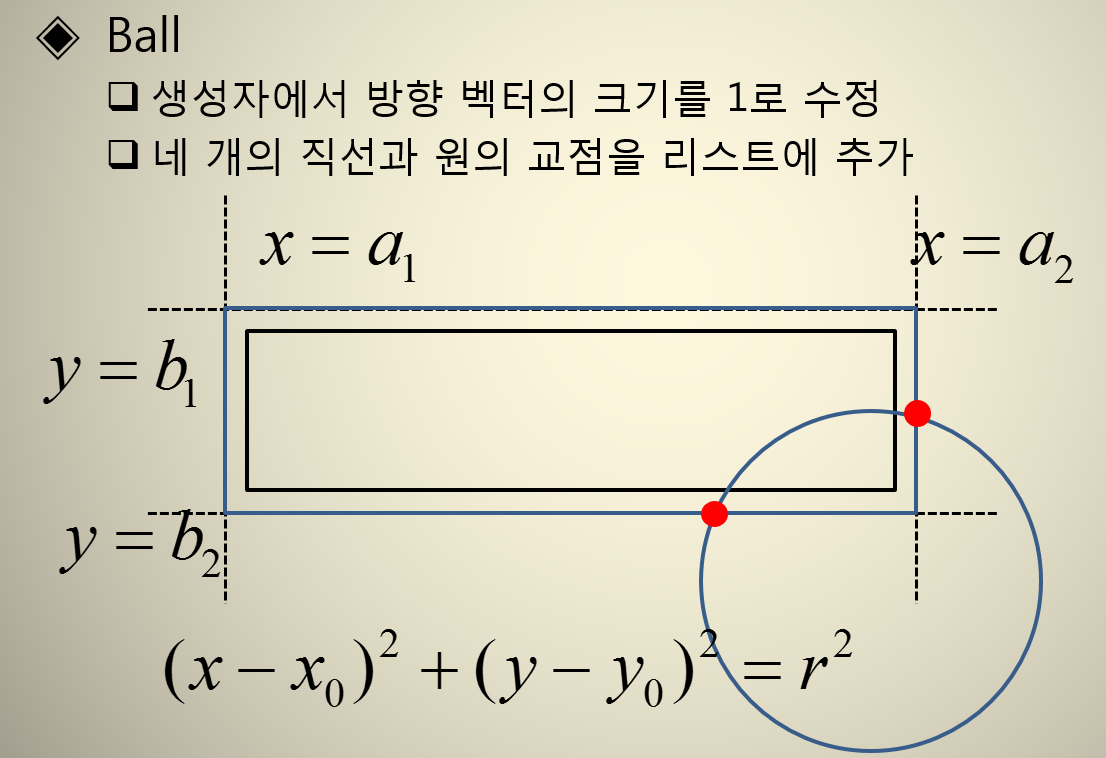
ex) Game.setup\_level(self)

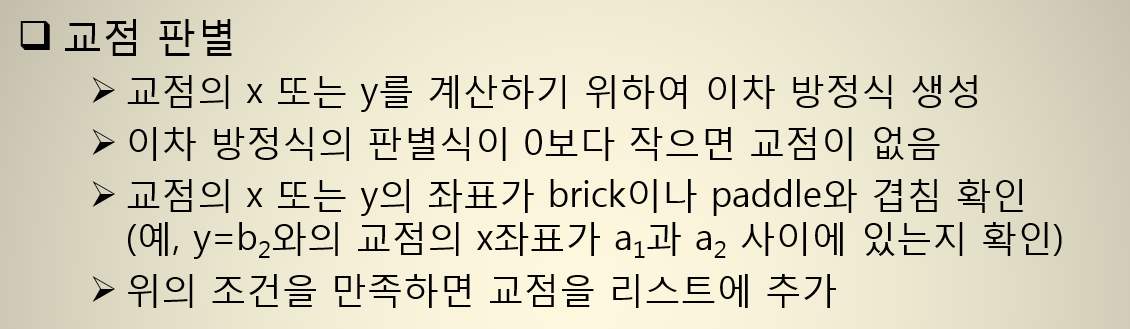
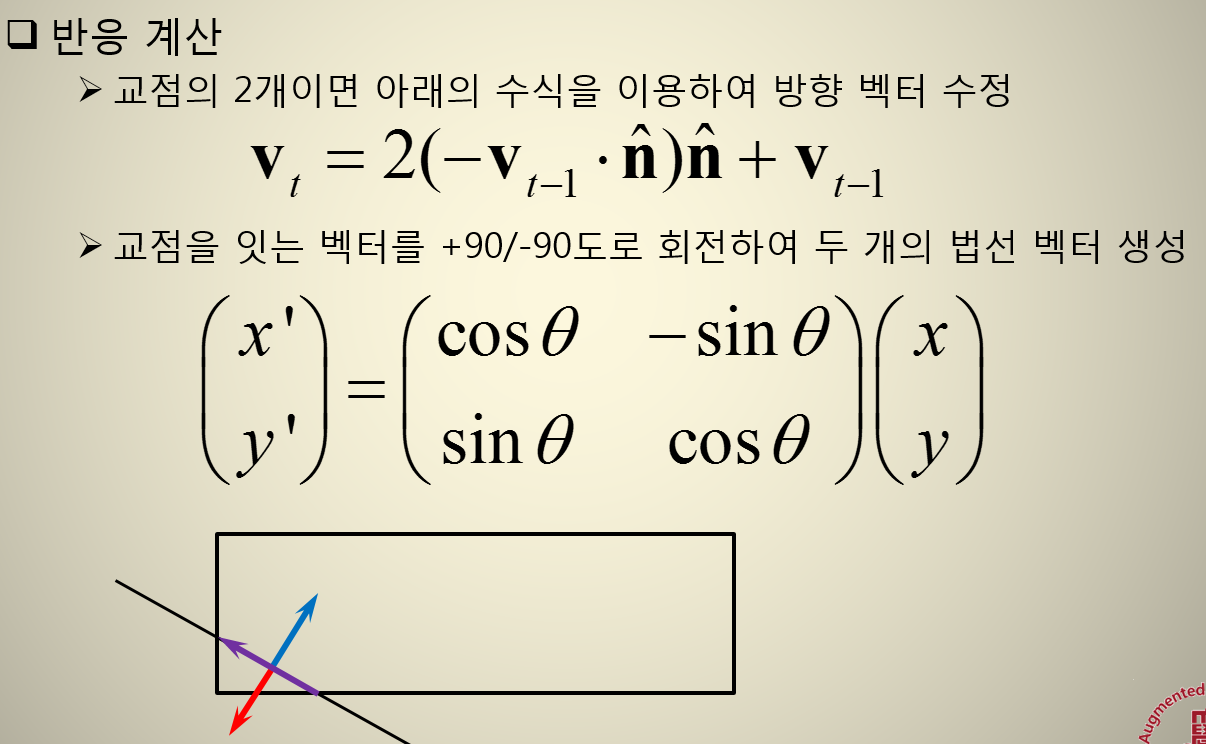
. Game.level에 따른 벽돌 층수 변경 및 임의의 벽돌만 생성

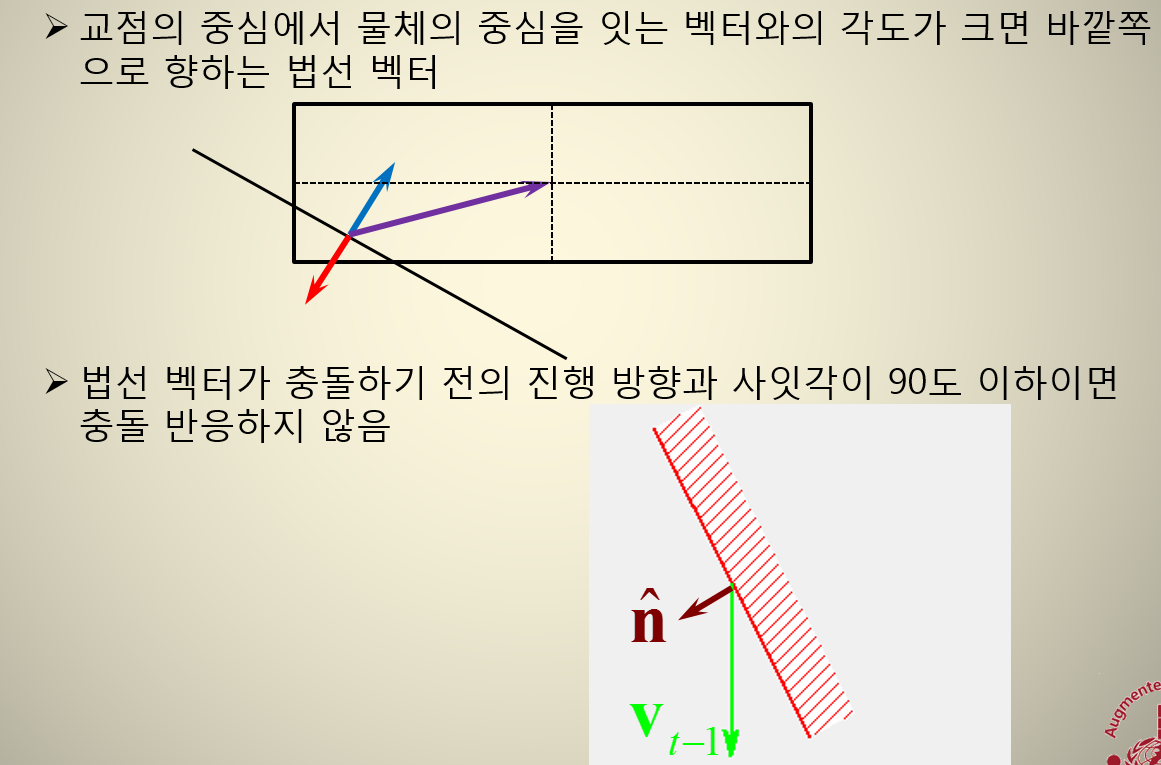
. random.randint(0, 9) 사용

|  |
| --- |
| Description]  level 5가 마지막 level, level이 증가할수록 hits와 블록 개수가 많아짐 |
| Code]  self.hits = ((1,0,0), (1,1,0), (1,1,1), (2,1,1), (2,2,1), (2,2,2))  def setup\_level(self):  for x in range(5, self.width - 5, 75):  if random.randint(0, 9) < 2self.level:  self.add\_brick(x + 37.5, 50, self.hits[self.level-1][0])  if random.randint(0, 9) < 2self.level:  self.add\_brick(x + 37.5, 70, self.hits[self.level-1][1])  if random.randint(0, 9) < 2self.level:  self.add\_brick(x + 37.5, 90, self.hits[self.level-1][2])  def setup\_game(self):  self.add\_ball()  self.update\_lives\_text()  if self.lives == 3:  self.setup\_level()  self.text = self.draw\_text(300, 200,  'Press Space to start')  self.canvas.bind('<space>', lambda \_: self.start\_game())  def add\_brick(self, x, y, hits):  if hits != 0:  brick = Brick(self.canvas, x, y, hits)  self.items[brick.item] = brick  def game\_loop(self):  self.check\_collisions()  num\_bricks = len(self.canvas.find\_withtag('brick'))  if num\_bricks == 0:  self.ball.speed = None  if self.level >= 5:  self.draw\_text(300, 200, 'You win!')  else:  self.level += 1  self.lives = 3  self.setup\_game()  def add\_brick(self, x, y, hits):  if hits != 0:  brick = Brick(self.canvas, x, y, hits)  self.items[brick.item] = brick |
| Game Shot] |

3. 모서리에 충돌한 공의 충돌 반응 처리



|  |
| --- |
| Description] |
| Code] |
| Game Shot] |