Ex.No:13 **Date:**	**SIMULATE BOUNCING BALL IN PYGAME**

AIM:

To write a python program to simulate bouncing ball inpygame.

ALGORITHM:

STEP 1:Define the class Ball and initialize the screen background, image and the circle for the ball.

STEP 2: Define the functions for update and for checking the boundary for the ball to hit

STEP 3: Define the main function for the actual boumcing ball simulation

PROGRAM:

```
""" bounce.py
   bounce on screen boundary contact
   only change is in ball.checkBounds.
"""

import pygame
pygame.init()

class Ball(pygame.sprite.Sprite):
def __init__(self, screen, background):
      pygame.sprite.Sprite.__init__(self)
self.screen = screen
self.background = background

self.image = pygame.Surface((30, 30))
self.image.fill((255, 255, 255))
pygame.draw.circle(self.image, (0, 0, 255), (15, 15), 15)
self.rect = self.image.get_rect()

self.rect.center = (320, 240)

self.dx = 5
self.dy = 5

def update(self):
oldCenter = self.rect.center
self.rect.centerx += self.dx
self.rect.centery += self.dy
pygame.draw.line(self.background, (0, 0, 0), oldCenter, self.rect.center)
```

```
self.checkBounds()

defcheckBounds(self):
        """ bounce on encountering any screen boundary """

        if self.rect.right>= self.screen.get_width():
self.dx *= -1
        if self.rect.left<= 0:
self.dx *= -1
        if self.rect.bottom>= self.screen.get_height():
self.dy *= -1
        if self.rect.top<= 0:
self.dy *= -1

def main():
    screen = pygame.display.set_mode((640, 480))
pygame.display.set_caption("Boundary-checking: bounce")

    background = pygame.Surface(screen.get_size())
background.fill((255, 255, 255))
screen.blit(background, (0, 0))

    ball = Ball(screen, background)
allSprites = pygame.sprite.Group(ball)

    clock = pygame.time.Clock()
keepGoing = True
    while keepGoing:
clock.tick(30)
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
keepGoing = False

allSprites.clear(screen, background)
allSprites.update()
allSprites.draw(screen)\

pygame.display.flip()

if __name__ == "__main__":
    main()
```

RESULT:

Thus the python program bouncing ball is executed using pygame.