I210, IUB

The following code (avoid_pizza.py) is modified from the pizza game we have seen, to demonstrate how one can implement the following tasks:
1) dealing with screen boundaries
2) adding a simple level counter
3) adding simple instructions
4) setting is_collideable = False to avoid collisions between the texts and moving sprites
5) changing the speed of a moving sprite

Questions:
1) What’s the parent class of Pizza and Pan? Pizza and Pan override what methods?
2) Now circle the main statements in the code for each of the tasks listed above.

from livewires import games, color

games.init(screen_width = 640, screen_height = 480, fps = 50)

class Pizza(games.Sprite):
    """ A bouncing pizza. """
    cycles = 0

    quit_label = games.Text(value = "Press Q to Quit", size = 25, color = color.black,
                            top = 5, right = 130)
    games.screen.add(quit_label)

    instructions = games.Text(value = "Avoid the pizza!!!", size = 60, color = color.red,
                               top = games.screen.height - 80, right = games.screen.width - 100,
                               is_collideable = False)
    games.screen.add(instructions)

    level_label = games.Text(value = "Level: ", size = 25, color = color.black, top = 5,
                              right = games.screen.width - 20, is_collideable = False)
    games.screen.add(level_label)

    level = games.Text(value = 1, size = 25, color = color.black, top = 5,
                        right = games.screen.width - 10, is_collideable = False)
    games.screen.add(level)

    def update(self):
        """ Reverse a velocity component if edge of screen reached. """
        if self.right > games.screen.width or self.left < 0:
            self.dx = -self.dx

        if self.bottom > games.screen.height or self.top < 0:
            self.dy = -self.dy

        if games.keyboard.is_pressed(games.K_q):
            self.game_over()
        Pizza.cycles += 1

        if Pizza.cycles == 300:
            Pizza.level.value += 1

            if self.dx > 0:
                self.dx += 1
            else:
self.dx -= 1

if self.dy > 0:
    self.dy += 1
else:
    self.dy -= 1

Pizza.cycles = 0

def game_over(self):
    """ End the game. ""
    end_message = games.Message(value = "Game Over",
                                size = 90,
                                color = color.red,
                                x = games.screen.width/2,
                                y = games.screen.height/2,
                                lifetime = 5 * games.screen.fps,
                                after_death = games.screen.quit)

    games.screen.add(end_message)
    self.destroy()

class Pan(games.Sprite):
    """ A pan controlled by player to catch falling pizzas. ""
    image = games.load_image("pan.gif")

    def __init__(self):
        super(Pan, self).__init__(image = Pan.image, x = games.mouse.x,
                                   bottom = games.screen.height)

        def update(self):
            """ Move to mouse x position. ""
            self.x = games.mouse.x
            self.y = games.mouse.y

            self.check_catch()

        def check_catch(self):
            """ Check if catch pizzas. ""
            for pizza in self.overlapping_sprites:
                pizza.game_over()

def main():
    wall_image = games.load_image("wall.gif", transparent = False)
    games.screen.background = wall_image

    pizza_image = games.load_image("pizza.gif")
    the_pizza = Pizza(image = pizza_image,
                       x = games.screen.width/2,
                       y = games.screen.height/2,
                       dx = 1,
                       dy = 1)
    games.screen.add(the_pizza)

    the_pan = Pan()
    games.screen.add(the_pan)

    games.mouse.is_visible = False

    games.screen.mainloop()

    # kick it off!
    main()