Game Development 3 – Obstacles

Last week we added movement so that we can move the bird around the screen. This week, we will add obstacles that we have to avoid.

Task 1: Obstacles

Currently, Birdie can freely move around the world. This does not make for a very exciting game which is why we will be adding obstacles next.

Add the following function to the code. What do you think it allows you to do?

```
def add_obstacle():
    top = random.randrange(20, 250)
    top_obstacle = Obstacle(True, top)
    bottom_obstacle = Obstacle(False, top + 150)

    top_obstacle.add(obstacles)
    bottom_obstacle.add(obstacles)
```

Once you know what this function does, add a comment above it to explain it to the reader of the code. Now call the function in the right place of the code and run the code. Close the game and run it again – can you see a difference?

Task 2: Moving Obstacles

The next step is to move the obstacles. To do this we will write the following function:

```
# Update the position of all obstacles
def handle_obstacles():
    for obstacle in obstacles:
        obstacle.update()
```

Can you explain in your own words what this function does?
The next step is to actually call this function. We do this after handling movement of the bird in the main event loop:

```
while True:
    ….
    handle_movement()
    handle_obstacles()
    …
```

When you run this, you will notice that the obstacle now moves across the screen. However, it would be better if it appeared outside of the screen, move into and then out of the scene. To do this, find the following variable and assign the value 0 to it:

```
OBSTACLE_RIGHT_OFFSET = 0
```

Important:
Notice how the obstacle now disappears from the screen. This does not mean that it no longer exists, though. However, we can safely remove it from the group of obstacles by adapting the handle_obstacles() function:

```
# Update the position of all obstacles
def handle_obstacles():
    for obstacle in obstacles:
        obstacle.update()
        if not obstacle.is_visible():
            obstacle.remove(obstacles)
```

Task 3: Creating New Obstacles

So far, we only have one obstacle in the game. We will now add more periodically. To do this, we will use a familiar mechanism: events. The provided code already causes a "`LAUNCH_NEW_OBSTACLE`" to be sent every 4 seconds. Can you find the code that is responsible for this?

Now add logic to the handle_events() function to add a new obstacle whenever the "`LAUNCH_NEW_OBSTACLE`" event is sent.

Advanced: Can you think of any problems we could have run into had we not removed obstacles in the previous task that could no longer be seen?

Task 4: Advanced Extension – Classes

(This is a difficult extension! Do not worry if you don't understand the code here but please ask if you are interested. The material that is covered here is part of most first year university courses in computer science.)

As an extension, read some more of the provided code. In particular, look at what we call the "definition of Bird":

```
class Bird(pygame.sprite.Sprite):
    ...
```

Find the move_up, move_down, etc. functions and investigate what the bird actually does when you call one of these functions.
Now, think about what a "`class`" might be.

Next time we will handle collision between Birdie and the obstacles.