Game Development 1 – Your First Game

Today, we will start to make a very simple game called “Birdie”. This will take us a few weeks overall by the end of which you will be able to program simple games and show them off to your friends.

Today’s lesson is about getting a simple window with a bird to show up. **The lesson will be difficult and you will likely have many questions by the end. Don’t worry though as we will discuss the concepts from today in more detail in the coming weeks**.

Task 1:

Copy the provided birdie program into your folder. Try running the program. So far it does not do anything yet. To change this, let’s create a window.

Look for the comment in the provided code that says “Task 1” and write down the following line:

```
screen = pygame.display.set_mode(size)
pygame.display.set_caption("Birdie")
```

How would you read out the first statement?
What do you think the second line achieves?

Now run the program, what can you see?

Task 2:

Now that we have a window, we can start drawing the landscape. Birdie, our main character, loves to fly and so we will use clouds as a background image.
In your folder you should have a “cloud.jpg” image file, which shows one cloud. We will use this as the background.

```
cloud = pygame.image.load("cloud.jpg")
```

Run this program. We cannot see anything yet. This is because we have not told the computer to show the cloud in the window. To solve this problem, find the main event loop in the program and add the missing line:

```
while True:
    clock.tick(30)
    handle_events(pygame.event.get())
    screen.blit(cloud, cloud.get_rect())
    pygame.display.flip()
```

Think about what the last two lines might do here (do not worry about the two preceding lines at this point).

Task 3:

We now have a window with a cloudy background but our main character is nowhere to be seen. Let’s add the bird now.
Think about whether you want to create a bird only once or every time we run the code in the main event loop. What happens if you create a bird every time the code in the main event loop is run?

To create a bird, add the following:

```
bird = Bird()
sprite_group = pygame.sprite.Group(bird)
```

and add the following line to the main loop after the drawing of the background:

```
sprite_group.draw(screen)
```

Can you explain what we just did? What can you see when you run this program?

Task 4 (Difficult Extension! We will revisit this next time):

Birdie does flap his wings yet. As we have learnt, the main event loop must process all of the events. As it turns out, the code that you have already sends an event every time Birdie should move its wings. All that is left to do is to forward this information to the bird.

Take a look at the block of code starting at

```
def handle_events(events):
```

Currently, this block of code stops the program when the player clicks the cross because an event of type “pygame.QUIT” is sent.
Every time the bird should flap its wings an event of type “ANIMATE_BIRD_E” is sent. You can make Birdie flaps his wings using

```
bird.flap()
```

Can you animate the bird?

Next time we will be teaching Birdie how to move around.