

University of Hull
Department of Computer Science

Writing Games with Pygame

Vsn. 1.0 Rob Miles 2014

Week 2: Making a Game

These notes describe the practical elements of the course. They are to be used in conjunction with the slide decks.

Practical Break 1: Moving some cheese

In this break we are going to make some cheese move down the screen.

Getting started

We are going to use the programs that we wrote last week as the basis of the work this week. You will also need to find the `cheese.png` file and add that to the folder where you store your programs. If you didn't get as far as drawing the cheese, you can find the cheese image here:

<https://rob-miles.squarespace.com/s/cheese.png>



Before you go any further; perform the following:

1. Log into your system with the username and password that you know and love from the past.
2. Make sure you have a cheese image. If you don't have this, follow the link above in your browser, right click the image and then save it with the filename `cheese.png` in the same location as your program file.
3. Start the Idle environment.

Draw some moving cheese

The first thing we are going to do is make some cheese move down the screen a little way. This is the code that we have been working with during the presentation.

```
import pygame
```

```
class MovingCheese:
```

```
    def drawCheese(self):
```

```
        pygame.init()
```

```
        width=800
```

```
        height=600
```

```
        size = (width,height)
```

```
        surface = pygame.display.set_mode(size)
```

```
        cheeseImage = pygame.image.load("cheese.png")
```

```
        cheeseX = 40
```

```
        cheeseY = 60
```

```
        for i in range(1,50):
```

```
pygame.time.Clock().tick(30)
surface.fill((255,255,255))
cheeseX = cheeseX + 1
cheeseY = cheeseY + 1
cheesePos = (cheeseX,cheeseY)
surface.blit(cheeseImage, cheesePos)
pygame.display.flip()
```

```
fun = MovingCheese()
fun.drawCheese()
```



Before you go any further; perform the following:

4. Create a new Python file (you could call it MovingCheese) and then enter the above program into it. Hint: You can copy it out of the PDF file and paste it directly into the IDLE to make things a bit quicker.
5. Run the program and note that the cheese moves down the screen 50 places.

We now know the principles of games. We have an update behaviour (the bit that adds 1 to the X and Y position) and a draw behaviour (the bit that blits the cheese onto the screen). In a larger game these would be performed by different methods and on a really powerful computer with multiple processors the game would be engineered so that update and draw could be performed simultaneously. But for our needs we can work in the manner above.

Investigating Tick Speed

At the moment the program draws frames at a rate of 30 times a second. Let's investigate the effect of changing the speed.



Before you go any further; perform the following:

6. Change the program to draw frames at a rate of 60 times a second.
7. Run the program and note what happens.
8. Now try a frame rate of 15 frames a second.

When you increase the frame rate everything happens more quickly. It takes half as long for the cheese to move the 50 pixels down the screen. When you decrease the rate the game runs more slowly.

Moving off the Screen

At the moment the cheese only moves 50 pixels, which is not very far.



Before you go any further; perform the following:

9. Change the program so that the cheese moves 500 pixels.
10. Run the program. Note what happens when the cheese moves off the screen.

The Pygame system will only draw objects that are visible in the window. If you move the object off the screen the program will not generate an error, but it won't draw your objects either.

Getting Stuck

When you make the game move the cheese a long way you might notice the game getting stuck, especially if you move a window over the pygame display. This is because actions involving the

window cause events to occur in the game and we are not dealing with events at the moment. The events pile up and eventually Pygame becomes upset and stops running the program.

Practical Break 2: Cheese Steering

Now that we know how to deal with keyboard events we can use them to steer cheese around the screen:

```
import pygame

class MovingCheese:
    def drawCheese(self):
        pygame.init()
        width=800
        height=600
        size = (width,height)
        surface = pygame.display.set_mode(size)
        cheeseImage = pygame.image.load("cheese.png")
        cheeseX = 40
        cheeseY = 60
        cheeseYSpeed = 2
        while True:
            pygame.time.Clock().tick(30)
            for e in pygame.event.get():
                if e.type == pygame.KEYDOWN:
                    if e.key == pygame.K_ESCAPE:
                        return
                    if e.key == pygame.K_w:
                        cheeseY = cheeseY-cheeseYSpeed
                    if e.key == pygame.K_x:
                        cheeseY = cheeseY+cheeseYSpeed
            surface.fill((255,255,255))
            cheesePos = (cheeseX,cheeseY)
            surface.blit(cheeseImage, cheesePos)
            pygame.display.flip()

fun = MovingCheese()
fun.drawCheese()
```

This code moves the cheese up and down the screen. If the player presses **w** the cheese moves up. If the player presses **x** the cheese moves down. Each time the cheese is moved by an amount given in the **cheeseYSpeed** variable.



Before you go any further; perform the following:

11. Open a new file in Idle and enter the program into it (use cut and paste from the PDF).
12. Run the program. Note what happens when you press up and down. Is this going to make for good gameplay?

You will have noticed that when you press the characters the cheese moves a tiny amount each time. This is not going to be the basis of great gameplay. We want the cheese to move when the key is held down, and then stop moving when we release the key. How do you think we might do that?

We could use the `pygame.KEYDOWN` event to start the cheese moving and the `pygame.KEYUP` event to stop the moving.

```
def drawCheese(self):
    pygame.init()
    width=800
    height=600
    size = (width,height)
    surface = pygame.display.set_mode(size)
    cheeseImage = pygame.image.load("cheese.png")
    cheeseX = 40
    cheeseY = 60
    cheeseYSpeed = 2
    cheeseMovingUp = False
    while True:
        pygame.time.Clock().tick(30)
        for e in pygame.event.get():
            if e.type == pygame.KEYDOWN:
                if e.key == pygame.K_ESCAPE:
                    return
                if e.key == pygame.K_w:
                    cheeseMovingUp = True
            if e.type == pygame.KEYUP:
                if e.key == pygame.K_w:
                    cheeseMovingUp = False
        if cheeseMovingUp:
            cheeseY = cheeseY-cheeseYSpeed
        surface.fill((255,255,255))
        cheesePos = (cheeseX,cheeseY)
        surface.blit(cheeseImage, cheesePos)
    pygame.display.flip()
```



Before you go any further; perform the following:

13. Replace the **drawCheese** method in your game with the one above.
14. Run the game and notice that when you hold the **w** key down the cheese moves up. This is because the keydown event for **w** sets a flag called **cheeseMovingUp** and the keyup event clears it. The program only moves the cheese when the **cheeseMovingUp** flag is set to **True**.

We can use this every time we want to control movement by holding a key down. Of course sometimes you don't want this form of action, if you want to fire a missile when the player presses a key the program just needs to deal with the keyDown event.



Before you go any further; perform the following:

15. Add a **CheeseMovingUp** flag to your game that is set when the cheese is moving up.
16. Add code to the event processor so that when the player holds down the **x** key the cheese moves down.
17. What should happen if the player presses down and up at the same time? Test your program to find out.
18. Now add further code so that when the player holds down the **a** key the cheese moves left and when they hold down the **d** key the cheese moves right.

At this point you should be able to steer the cheese around the screen. Unfortunately you can also steer the cheese off the screen entirely, which is not good.

Clamping the Cheese

Stopping a value from going outside a particular range is called *clamping*. You see it a lot in video games. The game does not "know" to stop the player from moving off the screen, you have to make the program do the test for you. For example

```
if cheeseY<=0:  
    cheeseY = 0
```

This clamps the cheese on the screen so that it can't move off the top (remember that when y is zero the cheese is at the top of the screen). The code would run each time that we have calculated a new value of **cheeseY**. If it becomes less than zero it is *clamped* at 0.



Before you go any further; perform the following:

19. Clamp your cheese so that it can't move off the screen. You might find it useful to know that the screen is 800 pixels wide and 600 pixels high. The cheese image is 100 pixels square.

Rob Miles June. 2014