

What to do When Love Slows Down?

So far we have a spaceship that can fly around the screen. This doesn't take Love very long to draw each time. In fact Love can finish the instructions in `love.draw()` and `love.update()` about 180 times in one second. This is called frames per second or FPS for short.

When we want to turn the rocket we add 0.017 to the `spaceship.direction` each time Love calls `love.update()`. If Love is running at 180 FPS then in one second we will increase the value of `spaceship.direction` by

$$\text{FPS} \times \text{Increment} = 180 \times 0.017 = 3.06$$

When we add more things to draw, like a moon, Love takes longer to draw them. Now instead of 180 FPS, Love may only be able to call `love.draw()` and `love.update()` about 60 times per second. That means that in one second we will only increase `spaceship.direction` by

$$\text{FPS} \times \text{Increment} = 60 \times 0.017 = 1.02$$

This means that the rocket will a third as fast. If we want the rocket to behave the same when Love runs slower, we have to increase the turn value as Love gets slower.

Fixing the Problem

`Love.update(dt)`

`Love.update()` has an optional argument. The value **dt** tells us how long it's been (in seconds) since the last call to `love.update()`.

If Love is running at 180 FPS, then each loop takes 1/180 or about 0.0055 seconds. So when Love calls `love.update()` the value of `dt` will be 0.0055 seconds.

If Love is running at 60 FPS, then each loop takes 1/60 or about 0.0166 seconds. So when Love calls `love.update()` the value of `dt` will be 0.0167 seconds.

The trick is to multiply the `spaceship.direction` increment by `dt`.

At 180 FPS in one second we add

$$\text{FPS} \times \text{Increment} \times \text{dt} = 180 \times 3.06 \times 0.0055 = \mathbf{3.06}$$

At 60 FPS in one second we add

$$\text{FPS} \times \text{Increment} \times \text{dt} = 60 \times 3.06 \times 0.0167 = \mathbf{3.06}$$

So now, it doesn't matter how long it takes Love to execute `love.draw()` and `love.update()`. The rocket will turn at the same speed.

This problem isn't limited to turning speed. It also affects acceleration and movement. Anytime we increment a value in `love.update` we have to multiply the update value by `dt`.

Here are the **changes**. I also made the `turnSpeed` a bit bigger than 3.06 for faster turning and lowered the acceleration to 200. I also changed the key used for acceleration to 'W'. This will let us use the space bar for something else later.

In `love.load()`

```
spaceship.acceleration = 200
spaceship.turnSpeed = 3.4
```

In `love.draw()` add as the first line

```
love.graphics.print("FPS " .. love.timer.getFPS(),10,10)
```

This will display the FPS that Love is running at. You can remove the line displaying the moon and see how this number changes when you run the program,

In `love.update()`

Change `love.update()` to `Love.update(dt)`

```
spaceship.posX = spaceship.posX + spaceship.velX * dt
spaceship.posY = spaceship.posY + spaceship.velY * dt
```

```
if love.keyboard.isDown("a") then
    spaceship.direction = spaceship.direction - spaceship.turnSpeed * dt
end
```

```
if love.keyboard.isDown("d") then
    spaceship.direction = spaceship.direction + spaceship.turnSpeed * dt
end
```

```
if love.keyboard.isDown("w") then
```

```
    --some trigonometry magic
    spaceship.velX = spaceship.velX + math.sin(spaceship.direction) * spaceship.acceleration * dt
    spaceship.velY = spaceship.velY + math.cos(spaceship.direction) * -spaceship.acceleration * dt
end
```