

# SIMULATION & PHYSICS – PRACTICAL 3

STEERING  
SJORS GIELEN  
500765899

### 1. repeat the assignment you are implementing;

Create a ship that steers towards it's goal location.

Once the ship gets close ensure it neatly slows down to a halt.

### 2. explain your approach;

For the ship I created a steering force and a rocket force. The rocket force is to propel the ship in the direction it is currently facing while the steering force adds a relatively small vector with the goal to turn the ship in the direction of the orb. There is a small downside to this approach and that is when the ball is directly behind the ship the steering vector may be fully nullified due to the rocketforce. A better implementation would be to ensure the steering force is never being applied to the ship in a larger angle than half  $\pi$  compared to the rocketforce.

### 3. describe your code;

To ensure the slowing down behaviour I added three if statements. One for being far away, one for being too close (and thus fully standing still) and one for slowing down if neither of the above conditions are true. This is mostly done through handling with a maxSpeed float. This float is used to clamp the currentSpeed float between 0 and the maxSpeed. In most conditions the currentSpeed is simply getting increased by the accel float divided by the mass.

To calculate out the steeringForce I simply took the goalVelocity (the vector that represents the ship too the ball) and truncated it to the maxSteering value and divided that by the mass.

#### 4. show (relevant) code snippets;

```
Vector2 goalVelocity = target - this.position;

Vector2 steeringForce = Truncate(goalVelocity, maxSteering) / mass;

// arriving and stopping
float arrivingRadiusSquared = arrivingRadius * arrivingRadius;
float goalsquared = goalVelocity.LengthSquared();
if (goalsquared > arrivingRadiusSquared)
{
    //far away get there faster
    currentSpeed += accel / mass;
    this.maxSpeed = this.storedMaxSpeed;
}
else if (goalsquared < arrivingRadiusSquared / 10)
{
    //too close, stop
    currentSpeed = 0;
}
else
{
    //getting close, slow down captain
    maxSpeed = goalVelocity.Length() / arrivingRadius * this.storedMaxSpeed;
}
currentSpeed = MathHelper.Clamp(currentSpeed, 0, maxSpeed);

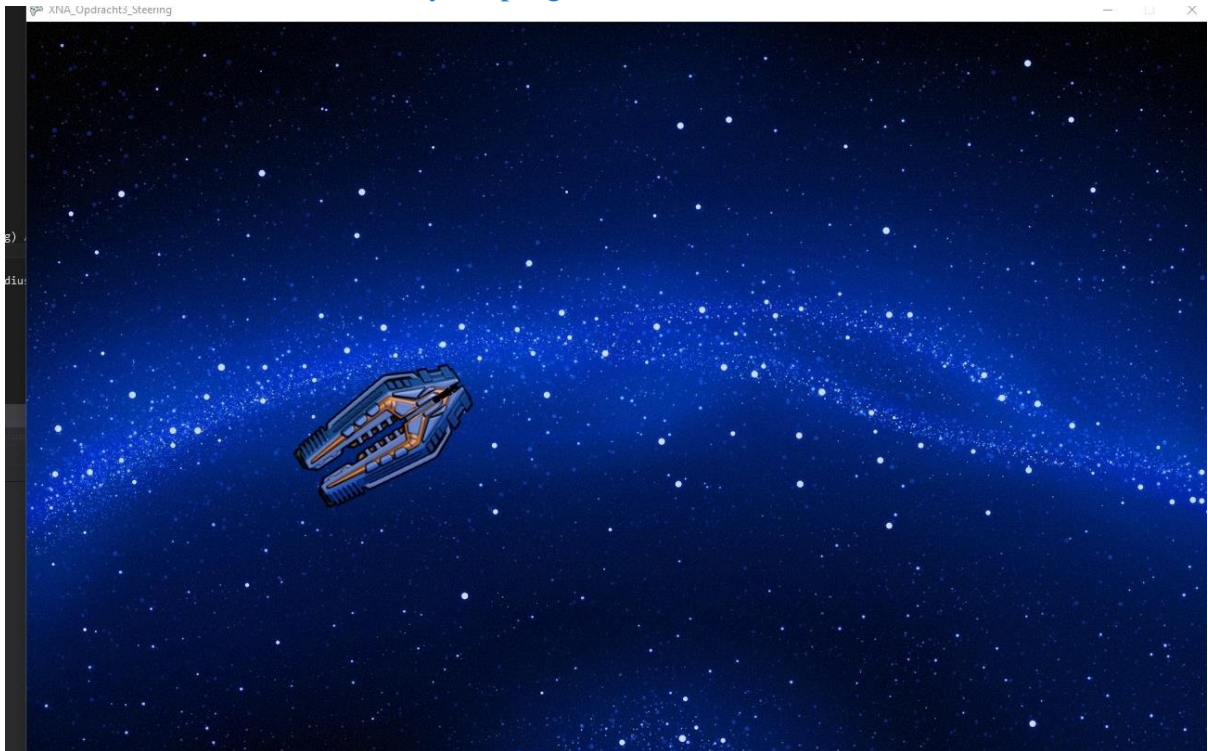
Vector2 rocketForce = new Vector2((float)Math.Cos(rotation), (float)Math.Sin(rotation)) * currentSpeed;

//add forces
velocity += rocketForce;
velocity += steeringForce;

//ensure speed does not exceed value
velocity = Truncate(velocity, currentSpeed);
// apply rotation
if (velocity != Vector2.Zero) {
    var angle = (float)Math.Atan2(velocity.Y, velocity.X);
    rotation = angle;
}
```

The ships update code.

## 5. include a screenshot of your program



Again I have made video's for the class as well These are found here:

<https://www.youtube.com/watch?v=TdETIUrTzHM&t=3684s&list=PLARkMALdMekM6EMkY0gcQSKvADVAX9zK5&index=5>