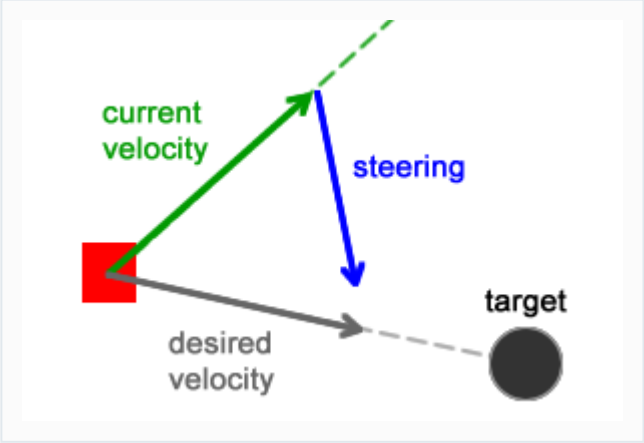
Flee

The seek behavior described previously is based on two forces that push the character towards the target: the desired velocity and the steering.

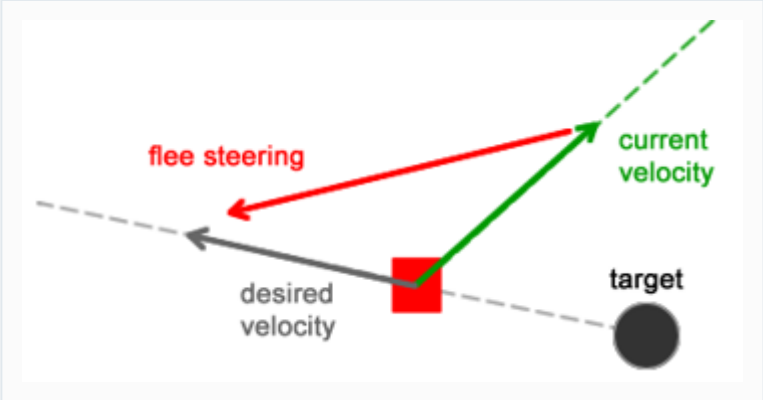
```
desired_velocity = normalize(target - position) * max_velocity
steering = desired_velocity - velocity
```

The `desired_velocity`, in this case, is the shortest path between the character and the target. It is calculated by subtracting the target's position from the character's position. The result is a force vector that goes from the *character* towards the *target*.



Seek behavior

The flee behavior uses those same two forces, but they are adjusted to make the character run *away* from the target:



Flee behavior

That new `desired_velocity` vector is calculated by subtracting the *character's* position from the *target's* position, which produces a vector that goes from the *target* towards the *character*.

The resulting forces are calculated almost the same way as in the seek behavior:

```
desired_velocity = normalize(position - target) * max_velocity
steering = desired_velocity - velocity
```

The `desired_velocity` in that case represents the easiest escaping route the character can use to run away from the target. The steering force makes the character abandon its current route, pushing it towards the direction of the desired velocity vector.

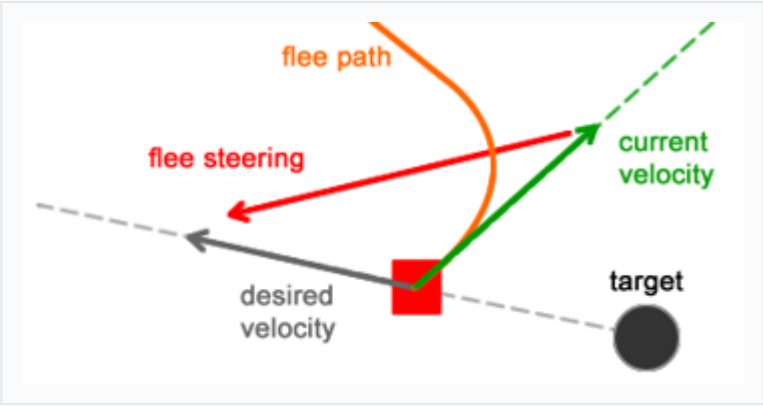
Comparing the desired velocity vector of the flee behavior with the desired velocity vector of the seek behavior, the following relation can be established:

```
flee_desired_velocity = -seek_desired_velocity
```

In other words, one vector is the negative of the other.

Adding Escape Forces

After the steering force is calculated it must be added to the character's velocity vector. Since that force is pushing the character away from the target, every frame the character will stop moving towards the target and start moving away from it, producing a **flee path** (the orange curve in the figure below):



The addition of those forces and the final velocity/position calculation are handled in the same way as before. Below is a demo showing several characters performing the flee behavior:

Move the mouse to move the target.

Every character is placed at the center of the moving area with a random velocity. They will try to flee from the target (the mouse cursor). The addition of all forces makes each character smoothly abandon its current route and flee the target.

Currently the target affects every character, ignoring the distance from them; it could have been limited to an "effect area", where the character would flee only if it is close enough to the target.