

Steer2D

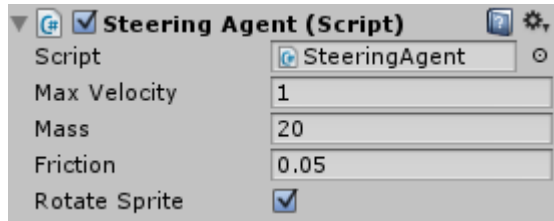
A Simple 2D Steering Library for Unity

1 CONTENTS

1	Quick Start.....	3
2	Steering Behaviors	3
2.1	Seek.....	4
2.2	Flee.....	4
2.3	Arrive.....	4
2.4	Pursue	5
2.5	Evade.....	5
2.6	Follow Path	6
2.7	Flock	7
3	Building Your Own Behaviors.....	7

1 QUICK START

To create a steering agent, simply add the 'Steering Agent' component to your Game object. This component is required for all steering agents and provides fields to alter the speed and agility of your agents.



- **Max Velocity** – The maximum velocity the agent can reach.
- **Mass** – The mass of an agent, defines how quickly the agent can accelerate and how quickly they can change direction. Setting a higher mass means an agent will change directions much more slowly making for smoother corners.
- **Friction** – The amount of friction applied to the agent. This is generally used to slow down and stop the agent when no force is being actively applied to it.
- **Rotate Sprite** – Whether or not the game object should be rotated to the direction it is currently heading.

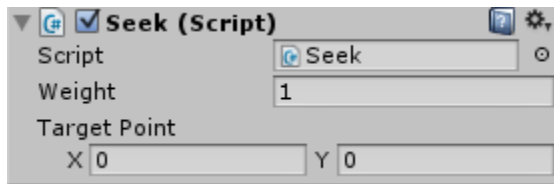
To add specific behaviors to the agent, add one of the following behavior components. Each provides its own movement behavior and various parameters to customize it. Multiple behaviors can be added and combined in a single agent.

- **Seek** – The agent will seek towards a target.
- **Flee** – The agent will flee from a target.
- **Pursue** – The agent will chase another agent, taking into consideration the future predicted position of the target agent.
- **Evade** – The agent will flee from a target agent, taking into consideration the future predicted position of the target agent.
- **Follow Path** – The agent will follow a path of points.
- **Flock** – The agents will form flocks with nearby agents.

2 STEERING BEHAVIORS

A single agent may contain multiple steering behaviors that will be combined to generate the final output. All steering behaviors have a 'weight' parameter. This parameter controls how much of an effect that particular behavior will have on the overall movement of the agent. You can alter the weight parameters of several steering behaviors to combine them to create more complex movement patterns for the agent.

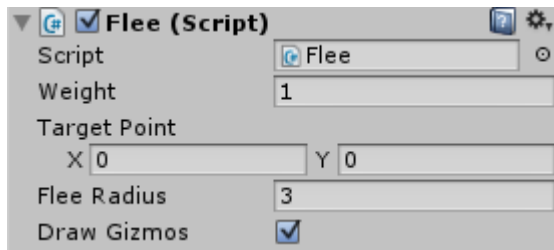
2.1 SEEK



The seek behavior simply moves the agent towards a specified target point.

- **Target Point** – The point the agent will seek towards.

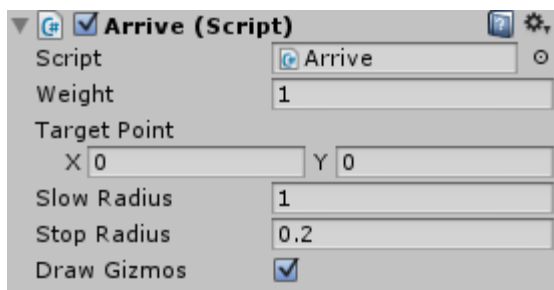
2.2 FLEE



The flee behavior moves the agent away from a specified target point while it is still inside the flee radius.

- **Target Point** – The point the agent will flee from.
- **Flee Radius** – The distance from the target at which the agent should stop fleeing.
- **Draw Gizmos** – If checked, a circle will be drawn in the editor denoting where the flee radius reaches.

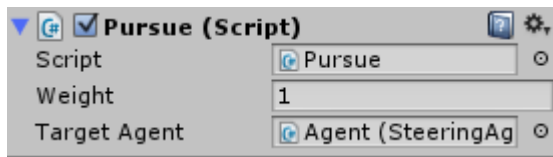
2.3 ARRIVE



The arrive behavior seeks towards a point and comes to a smooth stop as it gets within a certain distance of the point.

- **Target Point** – The point to seek towards.
- **Slow Radius** – The distance to the point at which the agent should begin to slow down.
- **Stop Radius** – The distance to the point at which the agent should come to a complete stop.
- **Draw Gizmos** – If checked, circles will be drawn in the editor denoting the slow and stop radiuses.

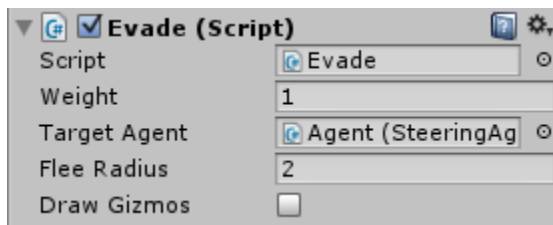
2.4 PURSUE



The pursue behavior seeks towards a target agent, taking into consideration the target agents future predicted position.

- **Target Agent** – The target steering agent to pursue.

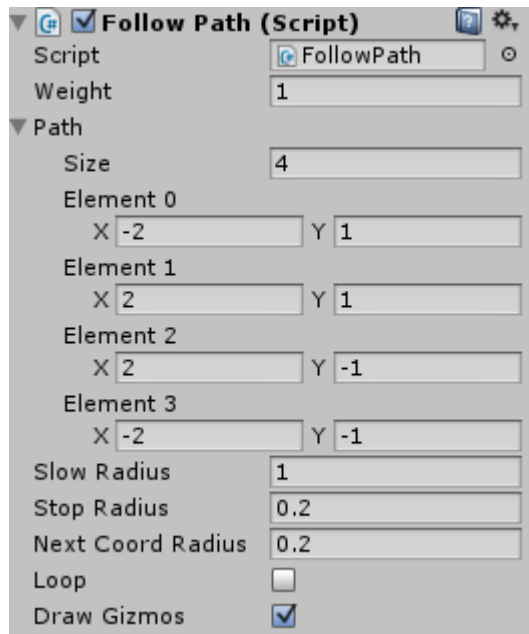
2.5 EVADE



The evade behavior attempts to flee from a target agent, taking into consideration the target agents future predicted position.

- **Target Agent** – The target steering agent to flee from.
- **Flee Radius** – The distance from the target agent at which the agent will stop fleeing.
- **Draw Gizmos** – If checked, a circle denoting the flee radius will be drawn in the editor around the target agent.

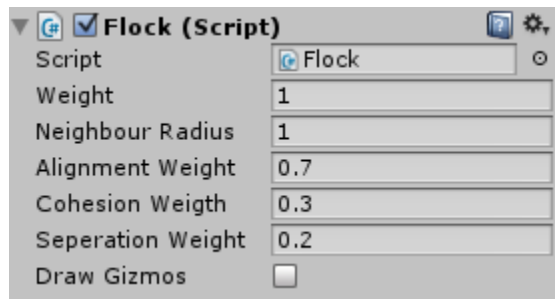
2.6 FOLLOW PATH



The follow path will seek the agent to each point in a list, then smoothly come to a stop at the final point.

- **Path** – An array of vectors specifying the path to follow.
- **Slow Radius** – The distance at which the agent should begin to slow down upon arriving at the final point.
- **Stop Radius** – The distance at which the agent should stop upon arriving at the final point.
- **Next Coord Radius** – The distance to a particular point that the agent must reach before it moves onto the next coordinate.
- **Loop** – If checked, agent will simply loop around the path continuously.
- **Draw Gizmos** – If checked, a line will be drawn in the editor denoting the path of the agent, as well as circles to denote the current coordinate and the slowing radiuses.

2.7 FLOCK



The flock behavior is a complex behavior that will cause several agents that are near each other to flock together and move in groups based off the parameters given. You may need to mess around with the various force weights to achieve the desired effect.

- **Neighbor Radius** – The distance between neighboring agents that the agent will attempt to flock towards. Any agents outside the neighbor radius are ignored by the agent.
- **Alignment Weight** – The weighting for the alignment force. This force attempts to turn agents so that they are all facing the same direction.
- **Cohesion Weight** – The weighting for the cohesion force. This force attempts to pull agents together into a group.
- **Separation Weight** – The weighting for the separation force. This force attempts to slightly separate the agents and stop them from overlapping.

3 BUILDING YOUR OWN BEHAVIORS

To create your own steering behaviors, create a class that derives from the 'SteeringBehaviour' abstract class. Then override the GetVelocity() function to return your own desired velocity. The SteeringBehavior parent class also exposes the 'agent' member which contains the SteeringAgent component for the current game object. You can use the 'agent' member to access the current velocity of the agent as well as its other values. A global list of agents in the scene is maintained in the public static member: 'SteeringAgent.AgentList'.

View the source files for the other steering behaviors to see examples of how you might implement your own.