Version 1.0.0

Project Dawn | Local Avoidance | Tool

Table of Contents

Overview

This package contains fast lightweight solution for local avoidance. It is developed with DOTS in mind, as result it takes advantage of Unity latest technology stack like SIMD mathematics, Jobs and Burst compiler.

Manual

This local avoidance uses custom solution called SonarAvoidance. Idea is very simple. Interest point constructs sonar volume shaped in cricle, that scans for nearby obstacles. Each obstacle cuts into the volume. Once all obstacles are scanned, the best direction can be calculated from it.

The solution is designed to be lightweight so it could be moved into any design. Thus the whole funcionality is contained of single struct called `SonarAvoidance`.

Firstly, structure needs to be constructed

```
var sonar = new SonarAvoidance(position, direction, innerRadius, outerRadius, new
SonarDynamics(sonarVelocity));
```

Next, each obstacle can be included

```
bool success = sonar.InsertObstacle(position, velocity, radius);
```

After all obstacles included the desired direction can be calculated

```
bool success = sonar.FindClosestDirection(out desiredDirection);
```

Finally once it is finished, resouces can be released with simply dispose interface

```
sonar.Dispose();
```

Debug

There is few very handy methods for drawing sonar volume. These methods use gizmos so they must be called from MonoBehaviour.OnGizmos message and also editor only.

Drawing sonar shape

```
sonar.DrawSonar();
```

Drawing the closest direction with the inner radius

```
sonar.DrawClosestDirection();
```

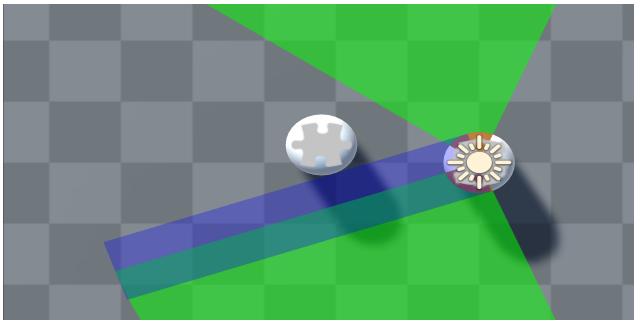
Obstacle Velocity

For obstacle avoidance that has velocity, `SonarAvoidance` uses trick of extending obstacle's shape towards the velocity. As result directions that would intersect with obstacle's velocity direction are less likely to be chosen. Sadly, this algorithm is not as accurate as static object's avoidance, for this reason there is settings to tweak according the needs. The settings can be passed into `SonarAvoidance` constructor with type `SonarDynamics`.

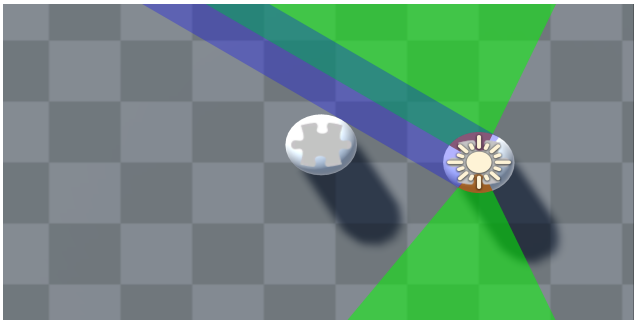
Velocity Scale

Velocity scale settings allows controling how much forward and back of obstacle will be extended. Here is few examples of it:

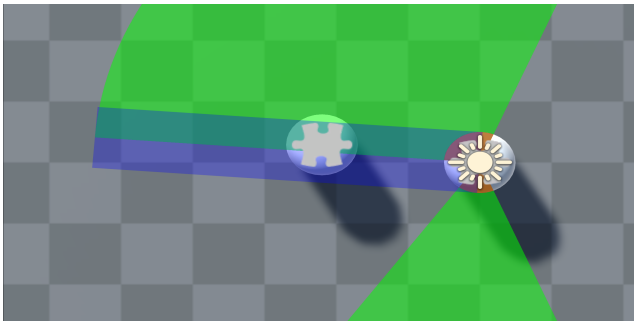
Default



With `ForwardVelocityScale` set to 1



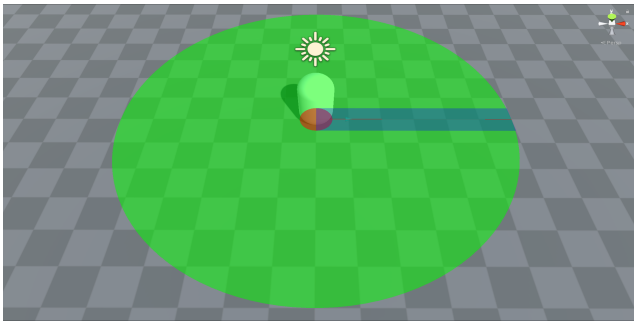
With `ForwardVelocityScale` set to 1 and `BackVelocityScale` set to 0.5



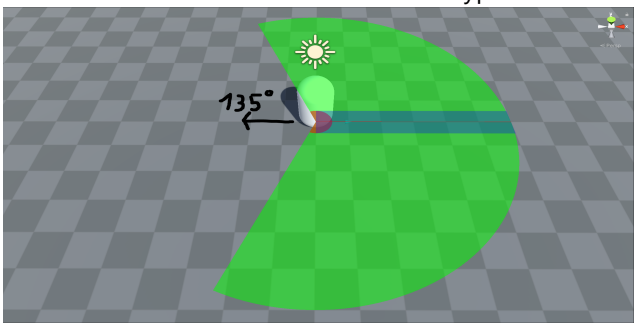
Obstacle Shapes

There is few different insertable obstacle shapes that can be used with sonar avoidance.

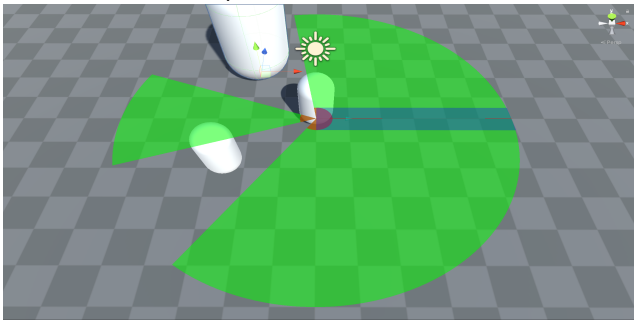
No obstacles



Radius obstacle with normal and radius. Typical used for cutting back



Circle obstacle with position and radius



Dependencies

- Tested with Unity 2019.4
- Package com.unity.mathematics@1.2
- Package com.unity.collections@0.9
- Package com.unity.burst@1.4

Support

If you have questions, bugs or feature requests use Discord.