UnitFormation

Table of Contents

Introduction

This package is a simple helper to calculate positions of the units. To start using it in your own code, first you must add "using TRavljen.UnitFormation" at the top of the C# file. From there you can start using the "FormationPositioner" class that will calculate the position based on the formation your provide. You can use of the existing ones that can be found in "Unit Formation/Scripts/Formations/" path.

Script Reference

FormationPositioner.cs

Namespaces

TRavljen.UnitFormation

```
namespace TRavljen.UnitFormation
```

Classes

FormationPositioner

```
public class FormationPositioner
```

Class responsible for providing unit positions in formation on a target position facing the respective angle. Use method 'GetAlignedPositions' when you are manually calculating facing angle of the formation. Use method 'GetPositions' when you wish for angle calculation to be done manually. It will be calculated based on magnitude of the position change and the angle from units center position to the new target position.

Methods

GetAlignedPositions

```
public static List<Vector3> GetAlignedPositions(
     int unitCount,
     IFormation formation,
     Vector3 targetPosition,
     float targetAngle)
```

Returns aligned units formation positions that are facing the passed angle.

unitCount: Amount of units in formation.
formation: Formation that units will position in.
targetPosition: Position of the formation.
targetAngle: Facing angle for the formation.

Returns: Returns aligned positions of the units in formation.

GetPositions

```
public static UnitsFormationPositions GetPositions(
     List<Vector3> currentPositions,
     IFormation formation,
     Vector3 targetPosition,
     float rotationThreshold = 4.0f)
```

Finds new positions for the passed positions and the formation. If distance from current positions center is less than rotation threshold, units formation will not be rotated around the target. New rotation angle is calculated from center position of all current positions and the target positions.

currentPositions: Current unit positions.
formation: Formation used on units
targetPosition: Position to where the units will be moved.
rotationThreshold: Threshold used to specify when the unit formation should be rotated around target position (pivot).

Returns: Returns list of the new unit positions and their new facing angle

UnitsFormationPositions

```
public struct UnitsFormationPositions
```

Data structure that represents the units new formation positions and angles.

Constructors

UnitsFormationPositions

```
public  UnitsFormationPositions(
     List<Vector3> unitPositions,
     float finalRotation)
```

Variables

UnitPositions

```
public List<Vector3> UnitPositions
```

Specifies the new positions that units can move to new formation.

FacingAngle

```
public float FacingAngle
```

Specifies the units facing angle (loot at direction) for the new position.

Formations

CircleFormation.cs

Namespaces

TRavljen.UnitFormation.Formations

```
namespace TRavljen.UnitFormation.Formations
```

Classes

CircleFormation

```
struct CircleFormation
```

Formation that positions units in a circle with specified angle and spacing between units.

Constructors

CircleFormation

```
public  CircleFormation(
     float spacing,
     float circleAngle = 360f)
```

Instantiates circle formation.

spacing: Specifies spacing between units in cricle
circleAngle: Specifies angle for units to be placed, 360 degree means that the units will go entire path around the circle and 180 degree angle means that only half of the circle will be formed.

Methods

GetPositions

```
public List<Vector3> GetPositions(
     int unitCount)
```

IFormation.cs

Namespaces

TRavljen.UnitFormation.Formations

```
namespace TRavljen.UnitFormation.Formations
```

Classes

IFormation

```
public interface IFormation
```

Defines the contact that all formations must implement. Formation should be generated or provided on the fly by calling GetPositions(int).

Methods

GetPositions

```
List<Vector3> GetPositions(
     int unitCount)
```

LineFormation.cs

Namespaces

TRavljen.UnitFormation.Formations

```
namespace TRavljen.UnitFormation.Formations
```

Classes

LineFormation

```
public struct LineFormation
```

Formation that positions units in a straight line with specified spacing.

Constructors

LineFormation

```
public  LineFormation(
     float spacing)
```

Instantiates line formation.

spacing: Specifies spacing between units.

Methods

GetPositions

```
public List<Vector3> GetPositions(
     int unitCount)
```

RectangleFormation.cs

Namespaces

TRavljen.UnitFormation.Formations

```
namespace TRavljen.UnitFormation.Formations
```

Classes

RectangleFormation

```
public struct RectangleFormation
```

Formation that positions units in a rectangle with specified spacing and maximal column count.

Constructors

RectangleFormation

```
public  RectangleFormation(
     int columnCount,
     float spacing,
     bool centerUnits = true)
```

Instantiates rectangle formation.

columnCount: Maximal number of columns per row (there are less rows if number of units is smaller than this number).
spacing: Specifies spacing between units.
centerUnits: Specifies if units should be centered if they do not fill the full space of the row.

Properties

ColumnCount

```
public int ColumnCount
     { get; }
```

Returns the column count which represents the max unit number in a single row.

Methods

GetPositions

```
public List<Vector3> GetPositions(
     int unitCount)
```

TriangleFormation.cs

Namespaces

TRavljen.UnitFormation.Formations

```
namespace TRavljen.UnitFormation.Formations
```

Classes

TriangleFormation

```
public struct TriangleFormation
```

Formation that positions units in a triangle with specified spacing.

Constructors

TriangleFormation

```
public  TriangleFormation(
     float spacing,
     bool centerUnits = true)
```

Instantiates triangle formation.

spacing: Specifies spacing between units.
centerUnits: Specifies if units should be centered if they do not fill the full space of the row.

Methods

GetPositions

```
public List<Vector3> GetPositions(
     int unitCount)
```