EASY SPLINE PATH 2D

Scripting API

Properties:

float spacing:

Distance between points in the evenly spaced calculated points on the curve (The lower this value the larger amount of points).

The calculations of the positions along the curve are calculated by dividing the spline in equidistant points, this is the distance between each said point. This needs to be adjusted to the scale you are working on. The smaller the scale, the lowest this value will be.

Default: *0.1*.

float resolution:

In the calculations this number multiplies the times each segment is subdivided, therefore the higher its value the more precise the resultant position will be.

Default: *1*.

Vector2[] points:

An array containing the equidistant points of the curve. Calculated in the function *SetUp*() and used in other operations to make calculations.

bool autoUpdate;

When set to true every operation that modifies the Spline will call the function *SetUp*() to update it.

Default: *false*.

Methods:

public void CreatePath()

Description: Initializes the *SplinePath2D* with the current default node type.

public void Reset()

Description: Reset the Spline and updates values, the current default node type will be used.

public Vector2 GetPointByDistance(float dist, bool loop, Space space)

Description: Gets the point on the curve corresponding to distance 'dist' in the specified space.

Returns: The point on the curve corresponding to distance *'dist'*. If loop is set to true it will start at the beginning when it gets to the end.

Parameters:

dist: Distance along the curve.
loop: If set to true loop.
space: If set to *Space*.Self it will return the local position if the point. If set to *Space.World* it will return the world position.

public Vector2 GetPointByPercent(float percent, bool loop, Space space)

Description: Gets the point on the curve corresponding to percent *'percent'* in the specified space.

Returns: The point on the curve corresponding to distance *'dist'*. If loop is set to true it will start at the beginning when it gets to the end.

Parameters:

percent: Percent of the curve. Range from 0 to 1.
loop: If set to true loop.
space: If set to Space.Self it will return the local position if the point. If set to *Space.World* it will return the world position.

public float PercentToDistance(float percent)

Description: Utility function to get the corresponding distance along the curve given a percentage.

Returns: Distance corresponding to the percent.

Parameters:

percent: Percent of the curve (0, 1.0).

public float DistanceToPercent(float dist)

Description: Given a distance, it returns the corresponding percent of the curve.

Returns: Percent of the curve corresponding to the distance.

Parameters:

dist: Distance along the curve.

public void AddSegment(Vector2 anchorPos)

Description: Adds a new anchor point in the specified location and connects it to the last anchor point in the Spline.

Parameters:

anchorPos: Position of the new anchor point, in world space.

public void SplitSegment(Vector2 anchorPos, int segmentIndex)

Description: Adds a new anchor point in the specified position and connects it to the anchor points corresponding to the segment.

Parameters:

anchorPos: Position of the new anchor point, in world space.
segmentIndex: Index of the segment to split.

public Vector2[] GetPointsInSegment(int segmentIndex)

Description: Get the control points of the spline corresponding to the segment.

Returns: An array containing the control points of the segment.

Parameters:

segmentIndex: The index of the segment.

public Vector2 GetPoint(int pointIndex, Space space)

Description: Get the position of the specified control point.

Returns: he position of the control point in the specified space..

Parameters:

pointIndex: The index.
space: If set to *Space*.Self it will return the local position if the point. If set to *Space.World* it will return the world position.

public void MovePoint(int pointIndex, Vector2 position, bool force = false)

Description: Move the specified control point to the a position. If force is set to false all affected points will be updated.

Parameters:

pointIndex: The index of the control point to be moved.
position: New position of the point.
force: If set to true other points will not be affected.

public void MoveSegment(int segmentIndex, Vector2 movement, Vector2 origin)

Description: Moves the handles corresponding to the specified segment along the vector 'movement', the origin vector is used to calculate how much each vector moves.

Parameters:

segmentIndex: Index of the segment.
movement: Movement vector to be applied.
origin: Position from where the segment is being dragged.

public void SetUp()

Description: Calculates the points of the curve and the total length.
Can be used in run time to update the Spline after modifying it.

public Vector2[] GetControlPoints(Space space = Space.World)

Gets the position of all control points in the given space. By default the space is Space.World.

Parameters:

space: If set to *Space*.Self it will return the local position if the point. If set to *Space.World* it will return the world position.

public Vector2[] GetAnchorControlPoints(Space space = Space.World)

Gets the position of all anchor control points in the given space. By default the space is Space.World.

Parameters:

space: If set to *Space*.Self it will return the local position if the point. If set to *Space.World* it will return the world position.

public void GetAnchorType(int indx, SplinePath2D.AnchorStatus type, bool setUp = false)

Change the type of a given anchor (AUTO, CORNER, FREE or LOCK). If the property *'autoUpdate'* or if the param *'setUp'* is set to true it will call 'SetUp()'.

Parameters:

indx: The index of the anchor.
type: The type to assign to the anchor.
setUp: If set to true it will call *'SetUp()'* after assigning the anchor type.

public SplinePath2D.AnchorStatus ChangeAnchorType(int indx)

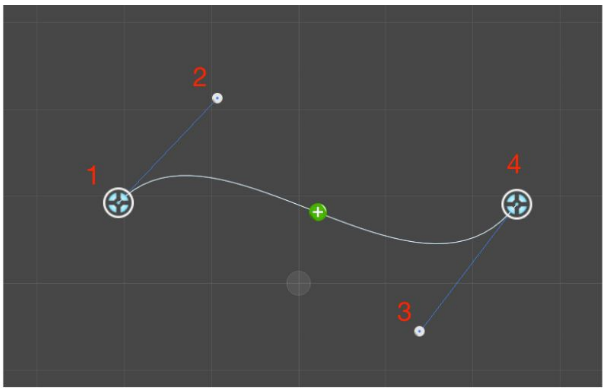
Get the type of a given anchor (AUTO, CORNER, FREE or LOCK).

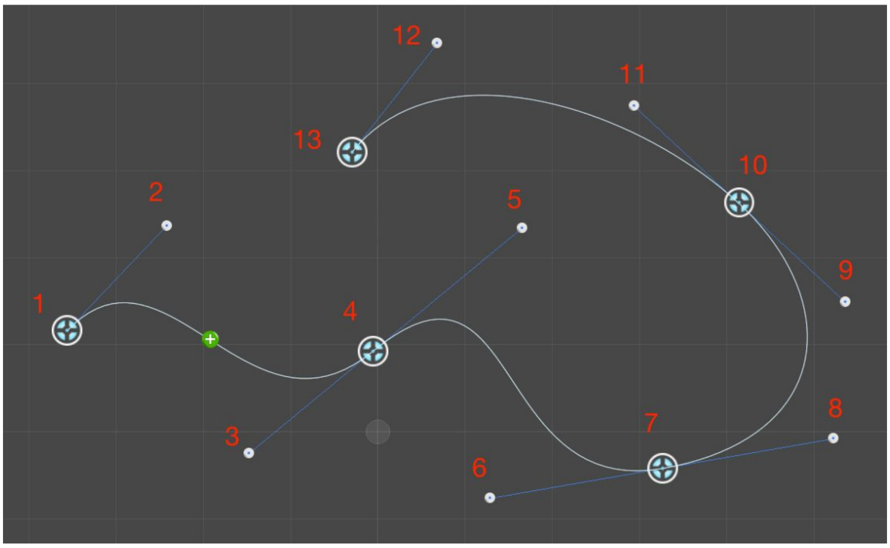
Parameters:

indx: The index of the anchor.

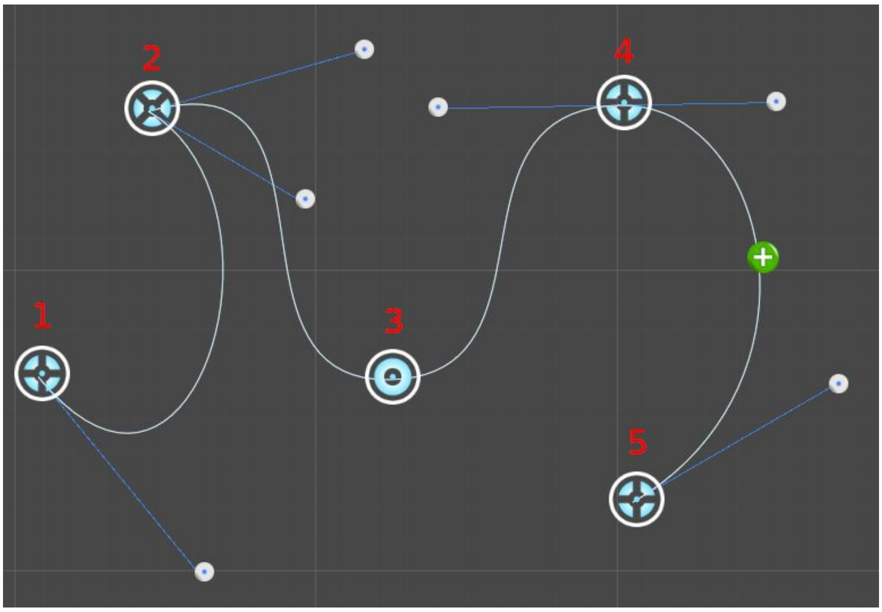
Point Numeration:

For every function that takes the index of a control point, the numeration of the points in the spline is as shown in the following pictures:





For every function that takes the index of an anchor control point, the numeration in the spline is as shown in the following picture:



For support contact: dcassetsuy@gmail.com