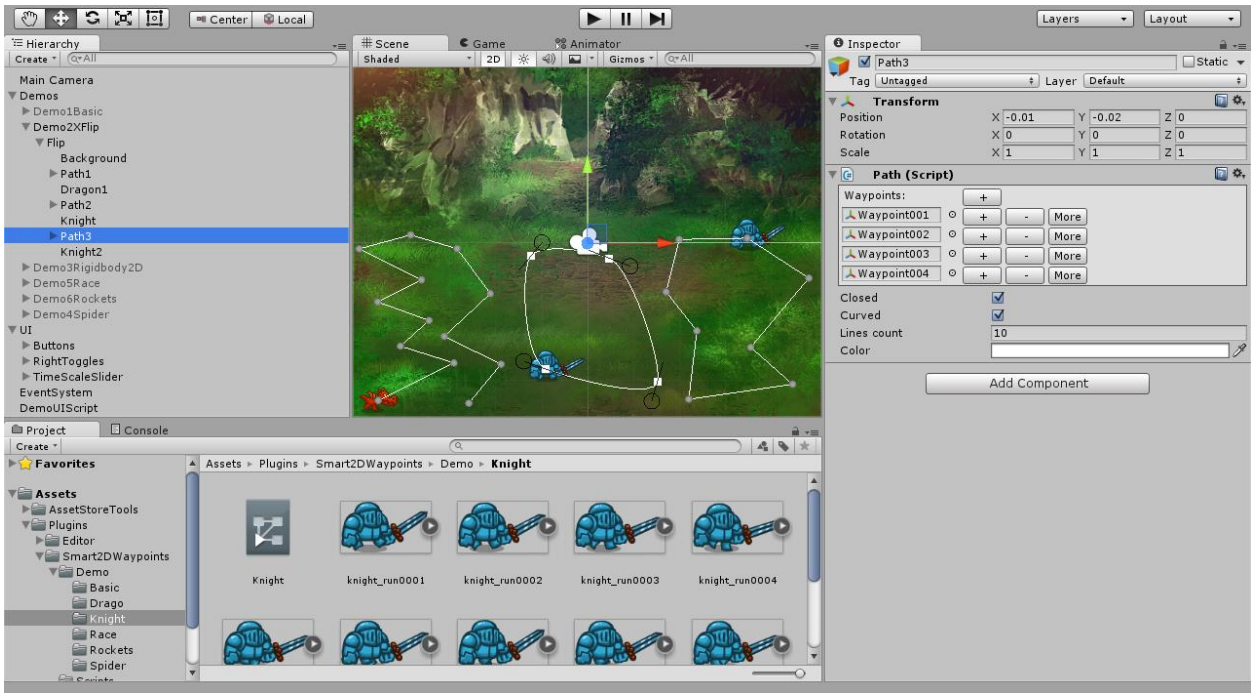
Smart 2D Waypoints Tutorial v1.6

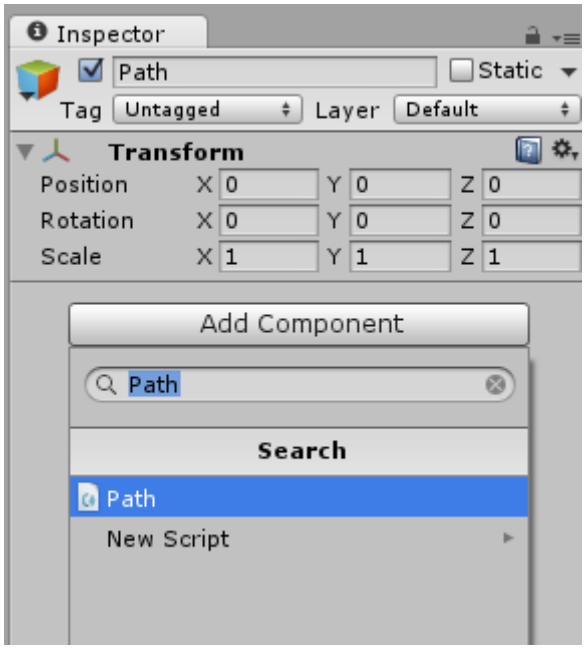
1. Demo.

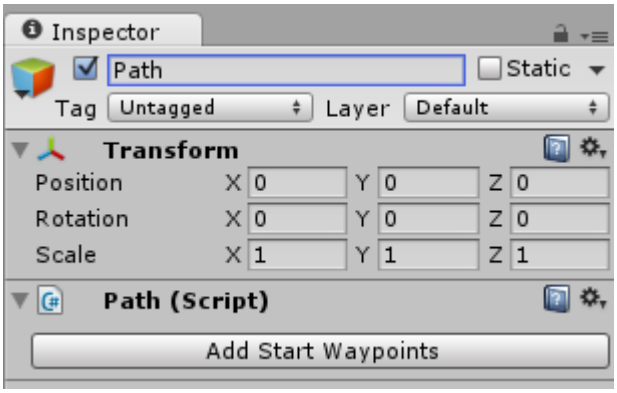
Open "Demo" scene from "Assets/Plugins/Smart2DWaypoints/Demo" directory and check several examples, which show how you can use plugin.
And you can check following web demo: http://nubick.github.io/plugins/waypoints.html



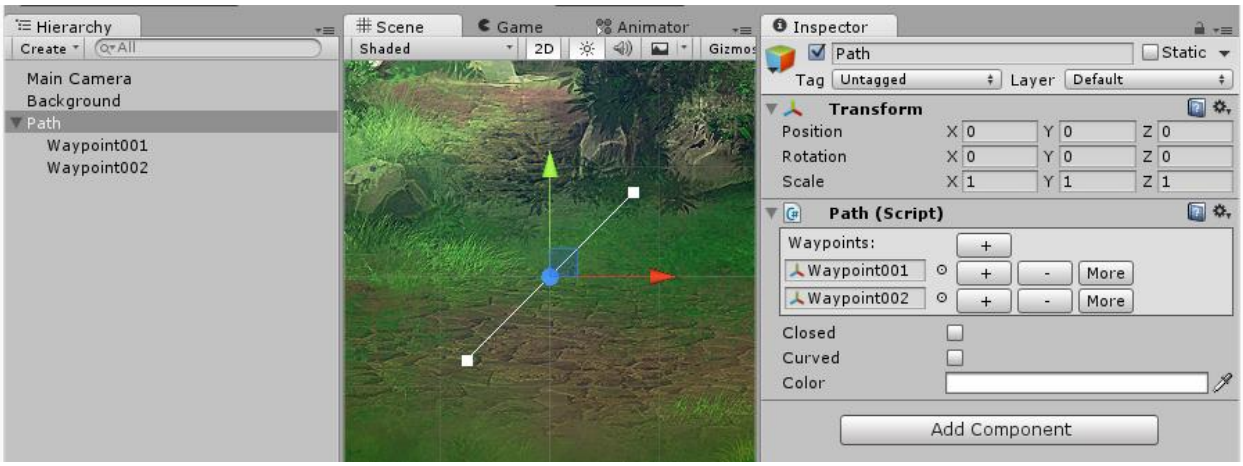
2. Path and waypoints.

3.1. Create empty game object in the game scene, name it “Path” and add “Path” component to it.





3.2. Click “Add Start Waypoints” button. Two waypoints will be created. These waypoints will appear in scene hierarchy under “Path” game object. You can select any waypoint using square handler in Scene View and move it.

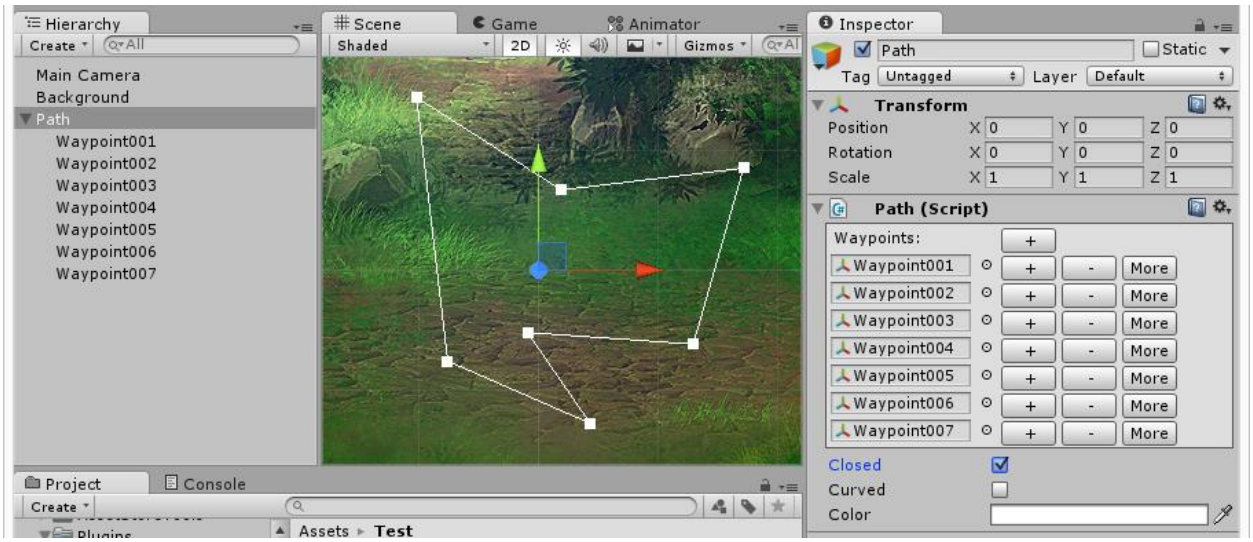


Go to inspector and add more waypoints using plus buttons.

You can try following settings from “Path” component:

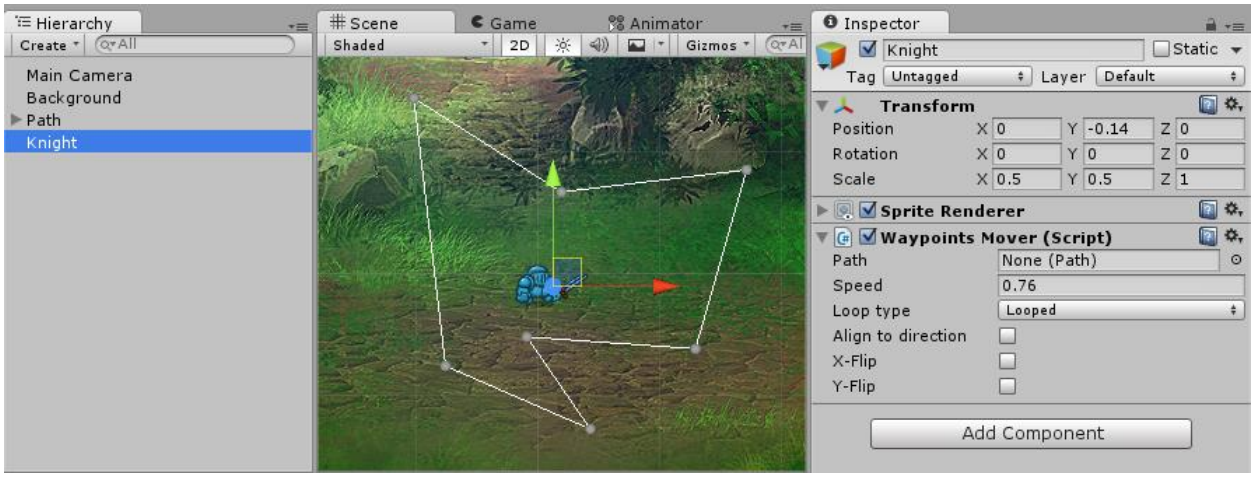
- “Plus” button add new waypoint under corresponding waypoint.
- “Minus” button delete corresponding waypoint.
- “Closed” checkbox make path closed.
- “Curved” checkbox make path curved.
- “Color” field set path color in scene view.
- “More” button add to waypoint additional behaviors.

Go to scene view and make path as you want, dragging waypoints. Now you get required path. Let us go further.



3. Waypoints mover.

Add to scene game object which you want to move (we use sprite knight as example). Add “Waypoints Mover” component to this game object.



Drag and drop “Path” game object from Hierarchy to “Path” field in “Waypoints Mover” component. Moving game object will placed to first waypoint. If you run the game, you will see how your game object move by path.

Check other following settings in “Waypoint Mover” component:

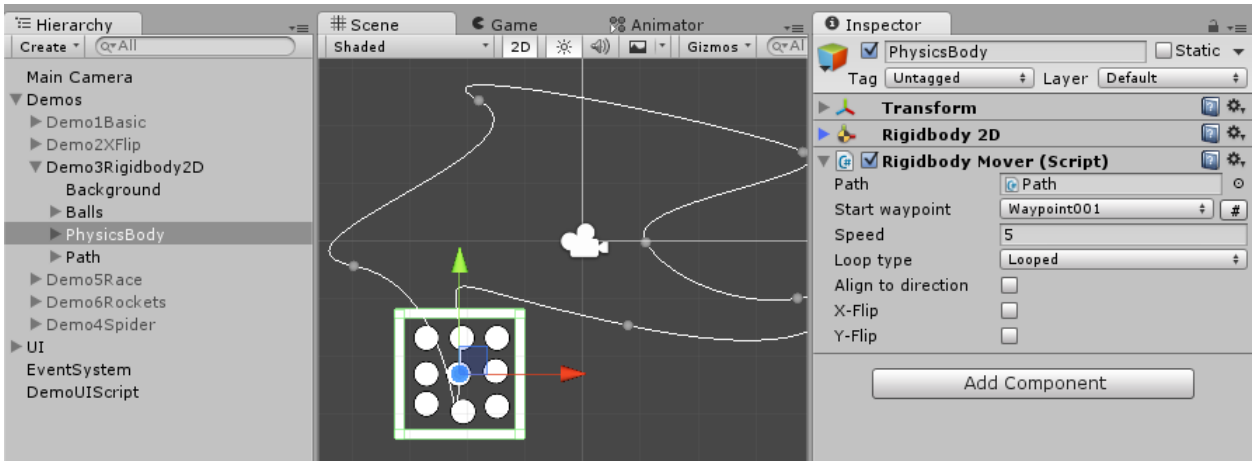
- “Speed” field set how fast game object is moving.
- “Loop type” drop down allows select one of three possible moving types:
  - “One Way” – stop when the last waypoint is reached.
  - “Looped” – when the last waypoint is reached, go to first waypoint and start moving again.
  - “Ping pong” – when the last waypoint is reached change direction and start moving again.
- “Align to direction” – rotate game object in the direction of movement.
- “Rotation offset” – available for “Align to direction” option. It makes offset of rotation in degrees.
- X-Flip – flip game object left/right according movement direction.
- Y-Flip – flip game object up/down according movement direction.
- “Destroy when finished” - This option is available when “One Way” loop type selected. When game object reach last waypoint it will be destroyed after that.

WaypointsMover has following additional **API** method:

- public void Pause() - pause game object movement.
- public void Resume() – resume game object movement if it was paused before.
- public bool IsPaused() – return if game object paused or not.
- public float Speed {get; set;} – use this property to change game object speed on air.

4. Rigidbody mover.

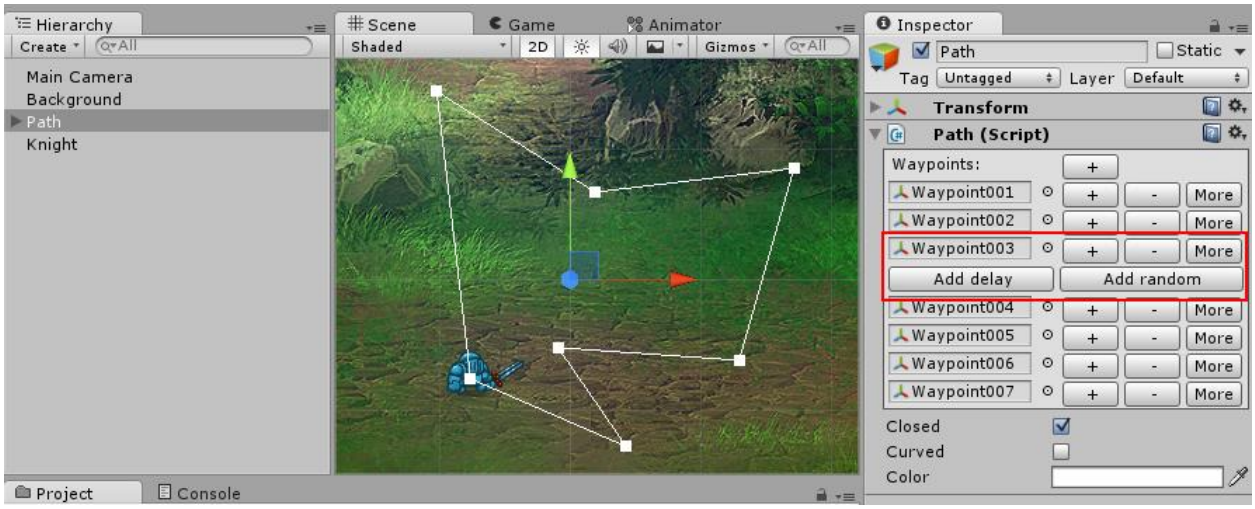
“RigidbodyMover” component needed for moving Rigidbody2D object by path. Change Transform.position for Rigidbody2D object, when you want to move it, is wrong solution. For this case, use “RigidbodyMover” component, which work similar “WaypontsMover” component but use Rigidbody2D.velocity and Rigidbody2D.angularVelosity parameters instead.



5. Delay waypoints.

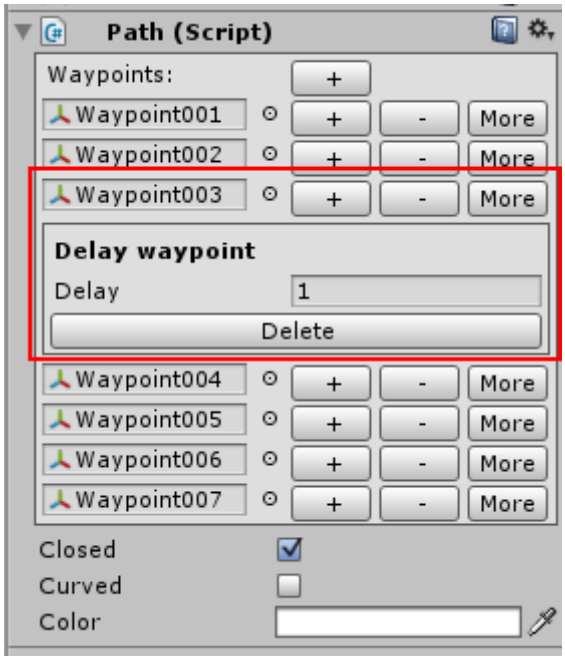
Delay waypoint stops moving object when it pass this waypoint and continue object movement after pause.

To add delay waypoint click “More” and after “Add delay” buttons. This will add “DelayWaypoint” component to waypoint.



Additional options will be available after that:

- “Delay” field set how long delay will be in second.
- “Delete” button delete delay from waypoint.



6. Support.

For any questions write to me: nubick@gmail.com