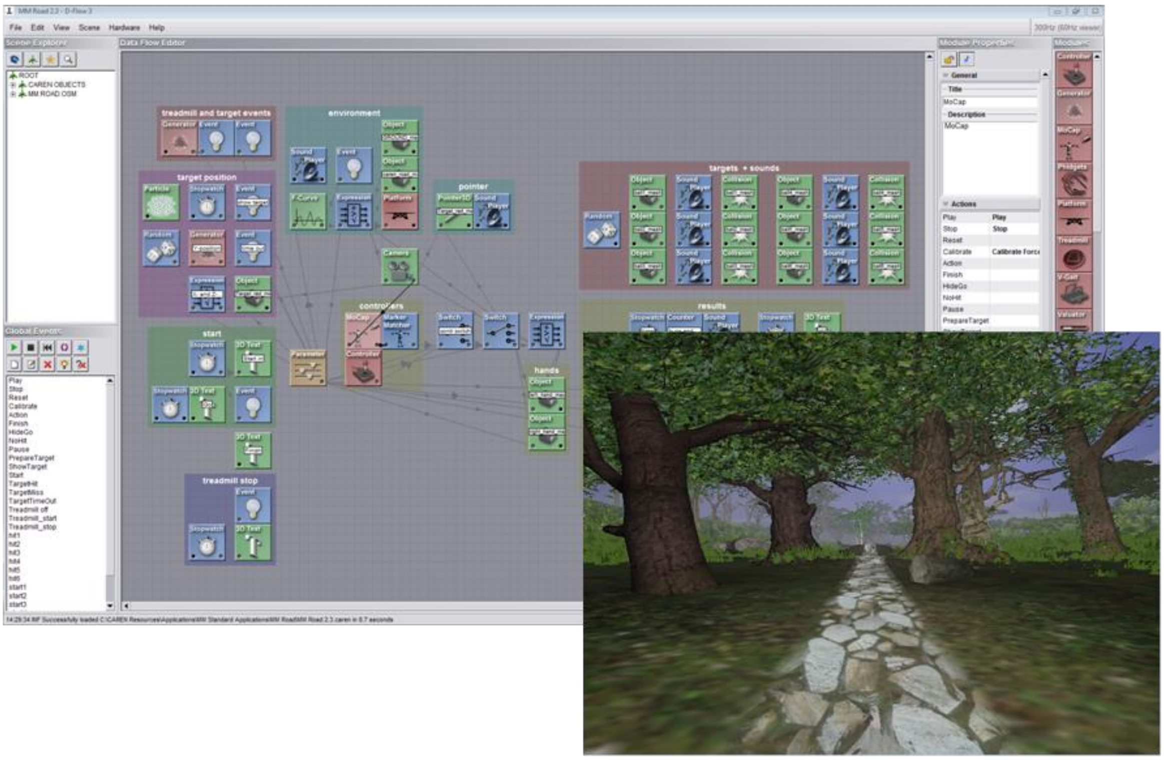
D-Flow

Gait tutorials



1 Preface

D-Flow is a software system designed for the development of interactive and immersive virtual reality applications, for the purpose of clinical research and rehabilitation. The key concept of the D-Flow software system is that the subject is regarded as an integral part of a real-time feedback loop, in which multi-sensory input devices measure the behavior of the subject, while output devices return motor-sensory, visual and auditory feedback to the subject. The D-Flow software system allows an operator to define feedback strategies through a flexible and extensible application development framework, based on visual programming.
The D-Flow Training Syllabus was designed by Motek Medical B.V. to provide new D-Flow users with the knowledge they need to start using the software.

This syllabus contains specific gait tutorials as an addition to the basics of D-Flow.

We hope you enjoy discovering all D-Flow's possibilities.

Regards,

Motek Medical B.V.

Website:
http://www.motekforcelink.com

Support:
support@motekforcelink.com
+31 (0)20 301 30 23

Knowledge Base:
https://hocoma.teachable.com/purchase?product_id=750298

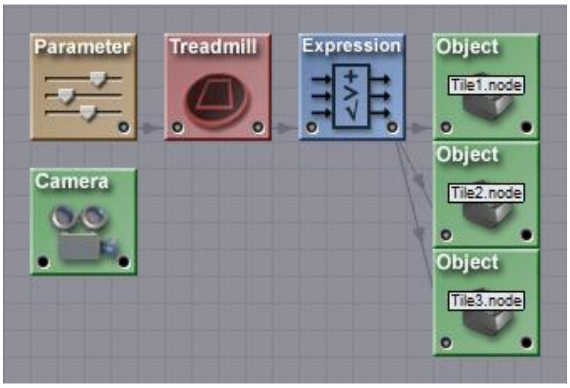
2 Contents

3 Gait tutorials

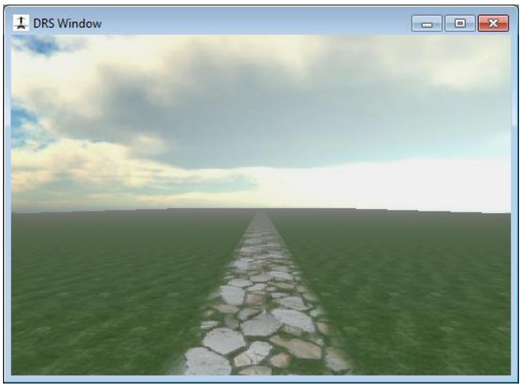
3.1 Gait Tutorial – Part 1: Creating an endless road

Introduction
In this tutorial we will demonstrate how to create a road that will go on forever. Making an endless road is a useful tool to create a realistic visual flow and will probably be the basis for most of your gait applications. It will be used in combination with the treadmill module that can either run in self-paced mode or on a fixed speed. In this example the speed will be controlled in the Runtime Console.
Files
Endless road tutorial files: 1. Download the scene called 'D-Flow tutorials.zip' from the teachable website and unpack the folder somewhere on your computer. 2. Run EndlessRoadScene.exe and select C:\Caren Resources as destination folder. 3. Place the "HBM2_Normal walking.txt" in C:\CAREN Resources\Data

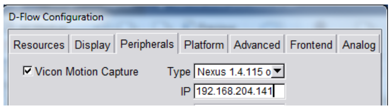
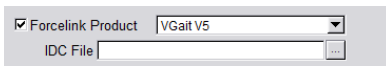
Preview of the result after completing this tutorial

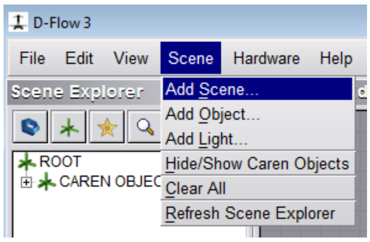
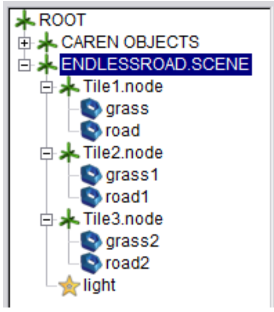
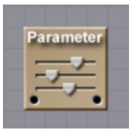
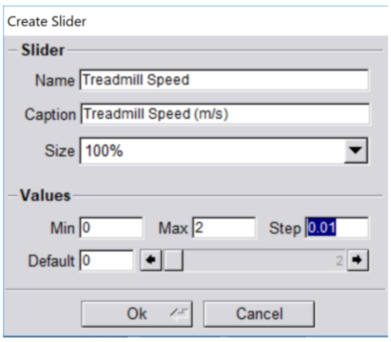


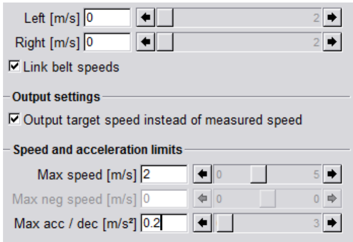
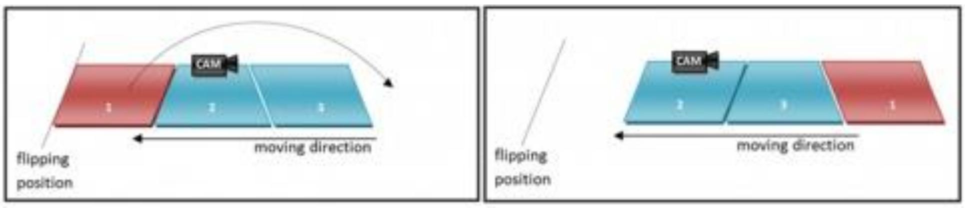
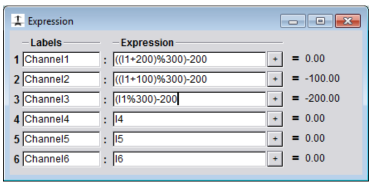
Data Flow Editor

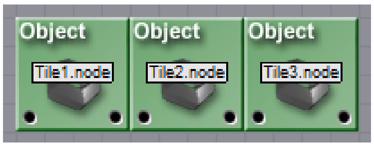


DRS Window

Procedure	Explanation	Illustration
In the first steps, the configuration of D-Flow is checked and if needed changed.		
1. Open the D-Flow Config tool: Windows | All programs | D-Flow 3.XX.X | D-Flow Tools | Configure D-Flow	With the D-Flow configuration tool, it is possible to select which hardware components are part of the system.	
2. In the Peripherals tab, enable the checkbox "Forcelink Product" and select VGait V5 (or	VGait V5 is for GRAIL systems with Pitch & Sway, for CAREN systems or M-Gait without pitch & sway select Treadmill	

Procedure	Explanation	Illustration
Treadmill 5) from the list.	V5	
3. Click Accept to save the configuration.		
Now the appropriate scene for this application will be loaded.		
4. (Re-)Start D-Flow and add the scene using the "Scene-->Add Scene" menu item.		
5. Go to the scene explorer and unfold the node called 'ENDLESSROAD.SCENE'.	Notice that there are three 'Tile' nodes. When you unfold all nodes, you will see that each tile holds the same objects: 'grass', 'road', 'wall_left' and 'wall_right'. This similarity is important to create a smooth (invisible) transition to the next tile.	
6. Activate the DRS Window by clicking on it once.		
7. Hit 'A' and 'G' on your keyboard.	This will hide the axes and grid respectively.	
First we will make sure we can control the treadmill speed from the Runtime Console.		
8. Drag a Parameter module from the module section into the Data Flow Editor.	With a Parameter module it is possible to create different parameters for the GUI of the Runtime Console.	
9. Open the Parameter module and create a new slider.		
10. Set the name to: 'Treadmill Speed' and the caption to: 'Treadmill Speed (m/s)'. Set the 'Max' value to '2'.	The treadmill belt speed will be able to vary between 0 and 2 m/s.	

Procedure	Explanation	Illustration
11. Drag a Treadmill module into the Data Flow Editor.	If the treadmill module is not available, step 1-3 is not done (correctly). Please perform step 1-3 again.	
12. Open the Treadmill module and enable ‘Link belt speeds’ and check if the ‘Max acc/dec [m/s2]’ is set to ‘0.2’.		
13. Create a connection between the Parameter module and the Treadmill module and connect the ‘Treadmill Speed’ to the ‘LeftBelt.Speed’	Since the belt speeds are linked, it does not matter which belt (i.e. ‘LeftBelt.Speed’ or ‘RightBelt.Speed’) is connected to the ‘treadmill speed’ slider in the Runtime Console.	
We will let the tiles move in reference to the camera. This will create the effect that the camera is moving over the road. Then, when the first tile reaches a certain position, it will be repositioned behind the third tile. When the second tile will reach this same position, it will be repositioned behind the first, and so on. 		
14. Drag an Expression module in to the Data Flow Editor.		
15. Double click on the Expression module and enter the following set of expressions: Channel1: ‘((I1+200)%300)-200’ Channel2: ‘((I1+100)%300)-200’ Channel3: ‘(I1%300)-200’	Each tile has a length of 100 m. The first tile has an offset of 0 0m, the second tile has an offset of 100 m and the third tile has an offset of 200 m. After moving 300 m (%300) the tile is set back to its starting position. (Tip: Use ‘Ctrl+C’ and ‘Ctrl+V‘ to copy and paste expressions from one row to another.)	

Procedure	Explanation	Illustration
16. Create a connection from the Treadmill module to the Expression module	The speed of the treadmill module will determine the speed of the scene passing by.	
17. Open the Connection Editor and delete the connections that were automatically created and connect the 'LeftBelt.Distance' to the first input of the Expression module.	Comparable to the belt speed, 'LeftBelt.Distance' and 'RightBelt.Distance' will output the same value when belt speeds are linked.	
18. Create three Object modules by dragging each 'Tile.node' from the Scene Explorer in to the Data Flow Editor.		
19. Create a connector from the Expression module to each of the Tile Object modules.		
20. Open the Connection Editor for each connector and create a connection between the correct output and the 'Object.PosZ' input of the correct Object module. Channel1 of the expression should be connected to Tile1, channel2 to tile 2 and channel3 to tile3.		
The objects should now be moving as described in the diagram above. The last step is to place a camera in the scene to define the optimal viewpoint for the scene.		
21. Drag a Camera module in to the Data Flow Editor.		
22. Double click on the Camera module and set the Y-position in the 'Translation offset [m]' section to '1.6' and the Z-position to '3'.		
23. Open the Runtime		

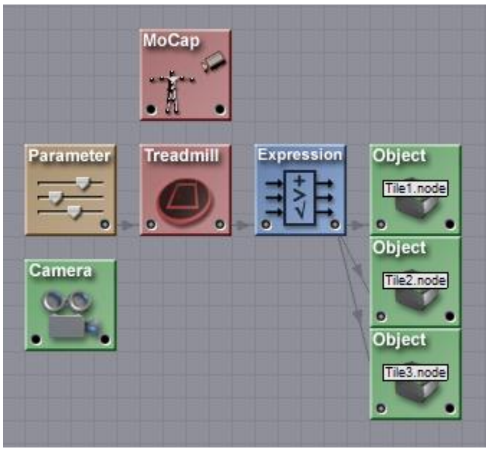
Procedure	Explanation	Illustration
Console (F2) set a speed using the 'Treadmill Speed (m/s)' slider and hit Play.		
24. Adjust the treadmill speed and note that the scene moves according to the speed.		
25. Go to 'File | Save application' and save the application as 'Endless Road'.	We will need this application in the following parts of the tutorial.	

3.2 Gait Tutorial – Part 2: Visualizing the subject

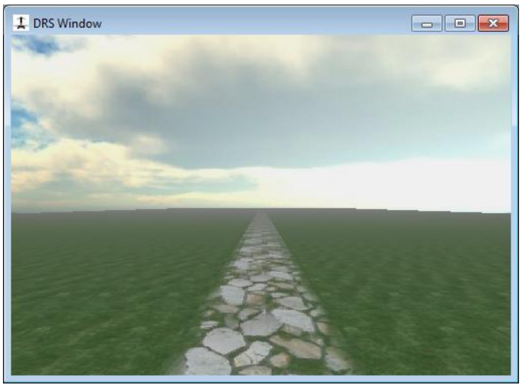
Introduction

In this tutorial we will demonstrate how to set up the MoCap module for gait analysis and visualize the subject in the DRS Window. For now we will use a file to demonstrate this functionality.

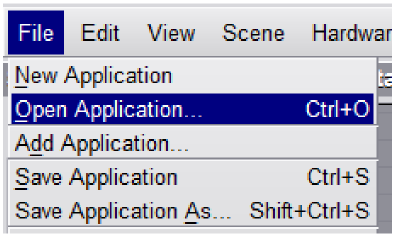
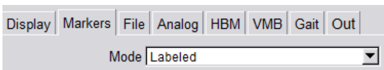
Preview of the result after completing this tutorial

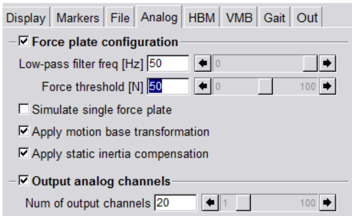
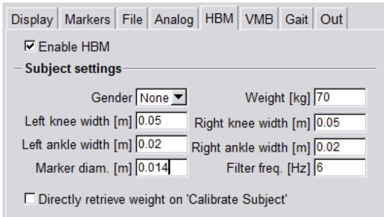
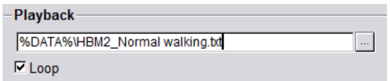
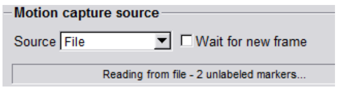


Data Flow Editor



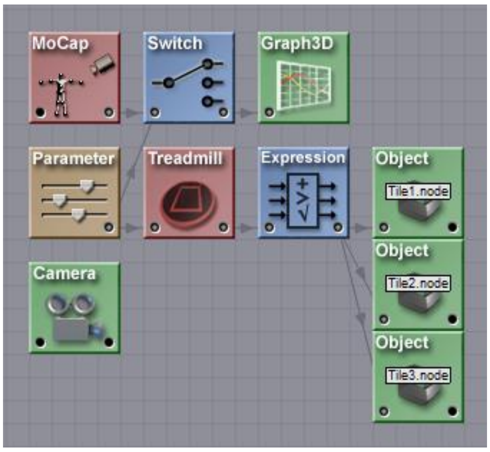
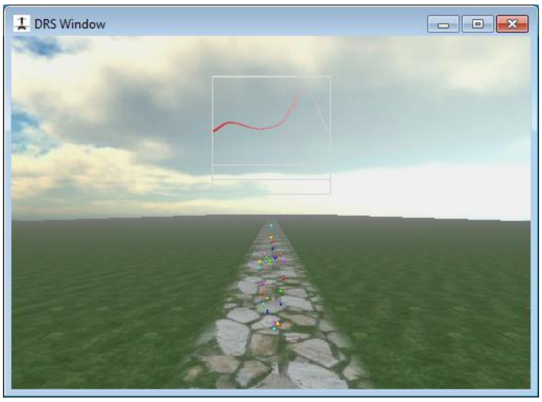
DRS Window

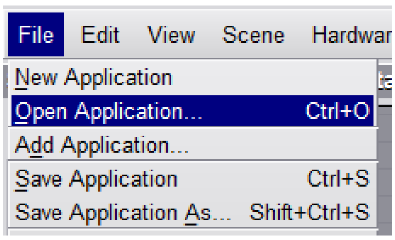
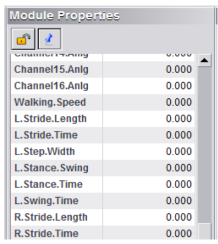
Procedure	Explanation	Illustration
We will start by loading the application that has been made in 'Gait Tutorial – Part 1: Creating an endless road'. Thereafter, we will set up the MoCap module for a Gait session.		
1. Go to 'File | Open Application' and open 'Endless Road.caren'	Using the Endless Road application allows us to visualize the subject walking on the road.	
2. Drag a MoCap module into the Data Flow Editor.		
3. Open the MoCap module.		
4. In the Marker tab: set the Mode to 'Labeled'.		
5. On the Markers tab: Set the number of markers to 26 and press Enter to confirm.		

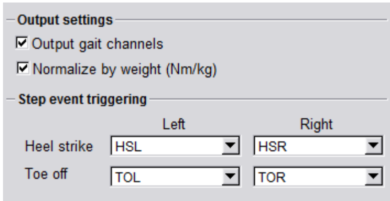
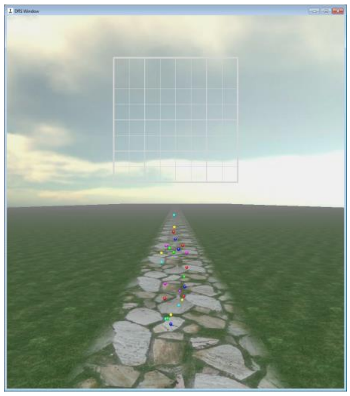
Procedure	Explanation	Illustration
6. On the Analog tab: enable ‘Force plate configuration’ and ‘Output analog channels’. Set the Force Threshold to ‘50’ N.	Enabling these checkboxes will allow you to collect force data. A threshold minimizes noise level during swing phase.	
7. On the HBM tab: check ‘Enable HBM’ and disable the option “Directly retrieve weight on Calibrate Subject’ and fill in a weight of 70 kg. Set Gender to Male.		
8. On the Display tab disable ‘Show HBM’.		
9. On the Gait tab: check ‘Enable Gait Processing’.		
The MoCap module is now set up for gait analysis. For now we will use a gait file that has been recorded previously to show the different options of the Gait functionality in D-Flow software. Using a file allows you to quickly test your application during development.		
10. Go to the File tab, at Playback: use the browse button to load the data file: ‘C:\CAREN Resources\Data\HBM2_normal walking.txt. Keep ‘Loop’ enabled.	Make sure that the demo file that is send with this tutorial is saved in the folder ‘C:\CAREN Resources\Data\’	
11. Set the source of the MoCap module to ‘File’.	When the MoCap module was first dragged in to the Data Flow Editor, there was no live source or file available therefore the module will go in to Simulation mode.	
12. Select the MoCap module and go to the Module Properties pane. Set the module action next to the event ‘Calibrate’ to ‘Calibrate HBM Subject’.	In order to output gait parameters, the HBM subject model needs to be initialized, i.e. calibrated.	

Procedure	Explanation	Illustration
13. Open the Runtime Console (F2). Set the treadmill speed to 1 m/s and hit 'Calibrate'. This is the small purple icon, next to the reset icon. Then hit 'Play'.	Normally the subject should stand in T-pose for that, but for now we will use the first frame of the file. The subject is walking at 1m/s in the file we are using. Therefore, the speed should be 1m/s in order to match the visual flow with the subject's movements.	
14. Note that the markers can be hidden by disabling 'Show markers' in the MoCap module.		
15. Go to 'File | Save application as' and save the application as 'Walking on an Endless Road'.	We will need this application in the following parts of the tutorial.	

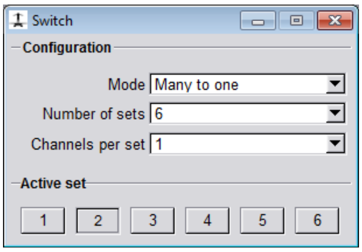
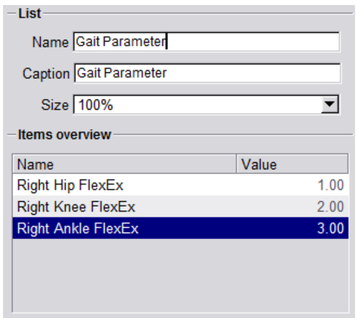
3.3 Gait Tutorial – Part 3: Plotting gait parameters

Introduction	
In this tutorial we will show you how to visualize gait parameters in real-time. We will add a graph with different gait parameters to our application that can be controlled from the Runtime Console.	
Preview of the result after completing this tutorial	
	
Data Flow Editor	**DRS Window**

Procedure	**Explanation**	**Illustration**
First we will load our latest Gait application and set up the MoCap module for outputting gait data and gait events.		
1. Go to 'File | Open Application' and open 'Walking on an Endless Road.caren'.		
2. Open the MoCap module and go the Gait tab.	Output settings and gait events can be set here.	
3. Enable 'Output gait channels'.	All gait parameters can now be used as input for other modules.	

Procedure	Explanation	Illustration
4. Make four new events in the Global Events section for heel strikes and toe offs. Name these HSL for heel strike left, HSR for heel strike right, TOL for toe off left and TOR for toe off right.	Heel strike and toe off events will now be broadcast. We will use heel strike events to refresh the graph later on. Toe off events can be marked in the graph.	
The MoCap module is now setup properly. The following steps will show you how to visualize gait parameters in a graph.		
5. Drag a Graph3D module into the Data Flow Editor.		
6. Open the Graph module.		
7. Enable 'Attach to camera '.		
8. Adjust the Position and Scaling until the graph is located above the road and has the right size. You may need to adjust the z-position to bring the graph into view.		
9. Make a connection between the MoCap module and the Graph3D module.		
10. Open the Connection Editor and delete the connections that were made automatically.		
11. Make a connection between a gait parameter – for example 'R.Knee.FlexEx' – and Channel 1 of the Graph module.		
12. Open the Runtime Console (F2) and consecutively hit 'Calibrate' and 'Play'.	Calibrate the HBM subject before starting the application.	

Procedure	Explanation	Illustration
13. Note that the gait parameter is not shown in the graph.	By default the vertical range of the graph is set to +/-1.	
14. Go to the Range tab and set the ‘Top Y value’ and ‘Bottom Y value’ to ‘90’ and ‘-90’ respectively and confirm with hitting enter.		
The following steps will guide you through setting up the Graph module. We will let the graph refresh on heel strike and mark toe off events.		
15. Select the Graph module and set the module action next to the ‘HSR’ event to ‘Cycle’.		
16. Open the Graph module and enable ‘Adjust cycle duration using ‘Cycle’ event’.	The graph now refreshes on every right heel strike.	
17. Select the Graph module and set the module action next to the ‘TOR’ event to ‘Mark’.	Right toe off is marked in the graph.	
Now we are going to add more parameters to the graph input and add the option to control the input of the Graph module from the Runtime Console. This way you’ll be able to check different gait parameters while your subject is walking.		
18. Drag a Switch module in to the Data Flow Editor.	This module will switch the input going to the Graph module.	
19. Open the Switch module and set the ‘Channels per set’ to ‘1’.		
20. Make a connection between the MoCap module and the Switch module.		
21. Open the Connection Editor and make the		

Procedure	Explanation	Illustration
following connections: R.Hip.FlexEx -> Set1. Channel1 R.Knee.FlexEx -> Set2.Channel1 R.Ankle.FlexEx -> Set3.Channel1		
22. Delete the connection between the MoCap module and the Graph module.		
23. Connect the Switch module to the Graph module.		
24. Open the Connection Editor and make a connection between 'Channel1' of the Switch module and 'Channel1' of the Graph module.		
25. Note that the input can now be altered through the Switch module. Activate set 1 to 3 to check the different input signals.		
26. Open the Parameter module and create a List.		
27. Name the list 'Gait Parameter' and create an item for each gait parameter. The value for each item should correspond to the channel number of the Switch module it is connected with.		
28. Make a connection between 'Gait Parameter' of the Parameter module and 'ActiveSet.Index' of the Switch module.	The input of the Graph module can now be controlled from the Runtime Console.	

Procedure	Explanation	Illustration
29. Open the Runtime Console (F2) and alter the 'Gait Parameter' by selecting a parameter from the list.		
30. Go to 'File | Save application as' and save the application.		
Tips to improve the application: - **Add a title to the graph that shows which parameter is visualized in the graph.** - **Add units to the X- and Y-axis.** - **Alter channel color to improve contrast between background and channel.** - **Add option in the Runtime Console to show and hide the graph.** - **Add option in the Runtime Console to show and hide 'the subject' walking on the road.**		

MMTRA-9027/ 20190429 / Printed: 20190429
Printed copies are uncontrolled. Consult DMS for up-to-date versions.

Form: MFL-05f011 / 20160808

3.4 Gait Tutorial – Part 4: Reaching a target

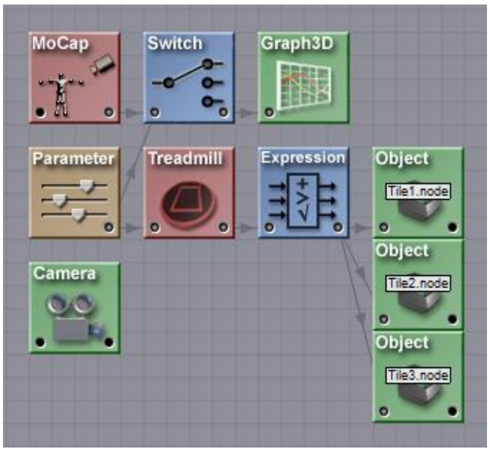
Introduction

This tutorial shows you how specific movements can be evoked using simple stock objects. The subject´s task is to adjust one gait parameter. When being successful a target will be reached and positive feedback will be provided. The magnitude of modifying the gait parameter can be set by the operator.

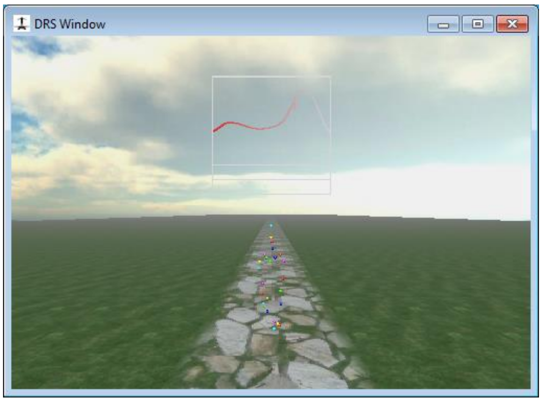
Files

- Walking on an Endless Road application (see Gait Tutorial – Part 2)
- HBM2_Normal walking.txt

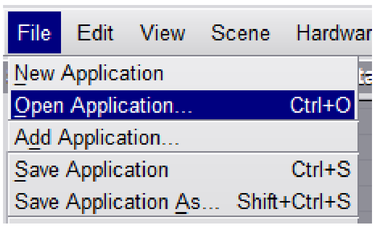
Preview of the result after completing this tutorial

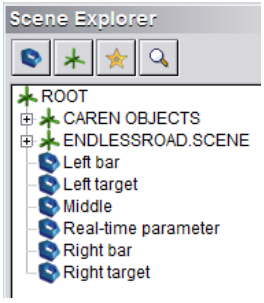
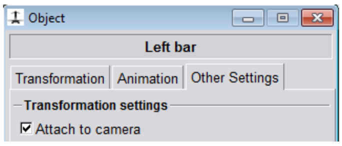


Data Flow Editor

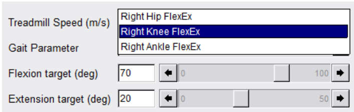
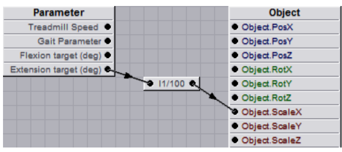


DRS Window

Procedure	Explanation	Illustration
The application that was made in Part 2 of the tutorial will be used as a basis for this application. First we will create the stock objects needed for feedback.		
1. Go to 'File | Open Application' and open 'Walking on an Endless Road.caren'		
2. Go to 'Scene | Add Object'. Add a cyan sphere, called 'Real-time'.	The real-time parameter object will visualize the real-time value of a chosen gait parameter.	

Procedure	Explanation	Illustration
3. Add three red cylinders, called ‘Left bar’, ‘Right bar’ and ‘Middle’.	The real-time parameter object will be moving along these bars. The middle bar will be used to mark the difference between flexion and extension.	
4. Add two blue spheres, called ‘Left target’ and ‘Right target’.	These objects will be placed at the end of the bars and will be used to evoke a certain movement.	
5. Drag the objects from the Scene Explorer in to the Data Flow Editor.		
The objects are now created and need to be positioned in the scenery. When doing so, we should keep in mind that the position of the targets (and thereby the length of the bars) is supposed to be variable and needs to be set in the Runtime Console.		
6. Open the Left bar and Right bar objects and attach the objects to the camera by enabling ‘Attach to camera’ in the Other Settings tab.		
7. Set the following offsets for both objects; Translation: Y=0.3, Z=-3 Scaling: Y=0.02, Z=0.02 (uncheck constrain axes) In addition, rotate the Left bar object 180° around the Z-axis.	The bars are positioned correctly. The scaling in X-direction (i.e. the length of the bars) will be set from the Runtime Console.	
8. Open the Left target and Right target objects and attach the objects to the camera by enabling ‘Attach to camera’ in the Other Settings tab.		
9. Set the following offsets for both objects:		

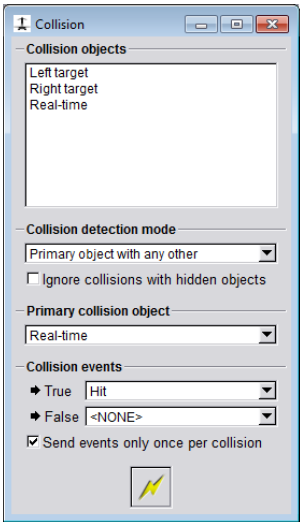
Procedure	Explanation	Illustration
Translation: Y=0.3, Z=-3 Scaling: 0.1 in all directions		
10. Create a connection between the Left bar and the Left target and vice versa for the right side.	.	
11. Open the Connection Editor, delete all connections and make a connection between the ‘Object.ScaleX’ of the bar and ‘Object.PosX’ of the target for both the left and right targets.	The length of the bars will determine the position of the target	
12. For the connector between the left objects: double-click on the connector, double-click on the math box and alter ‘I1’ to ‘-I1’.	The positive X values are at the right side, so for the left objects the values should be inverse (=negative).	
13. Open the Middle object and attach the object to the camera.		
14. Set the following offsets for the object: Translation: Y=0.25, Z=-3 Rotation: Z=90 Scaling: X=0.1, Y=0.01, Z=0.01		
15. Open the Real-time object and attach the object to the camera and position it in the middle of the bars, with the correct scaling.		
It’s now time to give the target objects the correct inputs. First we’ll need to add parameters to the Runtime Console and then we can determine the corresponding length of the bars.		
16. Create a list named ‘Gait Parameters’ containing right hip, knee and ankle flexion/extension. Check Part 3 (step 27, 28 and 29) of this tutorial.		

Procedure	Explanation	Illustration
17. Create a slider named 'Flexion target (deg)' and set the maximum value to '100', step to '1' and default to '70'.	This slider will determine the position of the right target sphere and can be used to impose a certain amount of flexion of the selected joint.	
18. Create a slider named 'Extension target (deg)' and set the maximum value to '50', step to '1' and default to '20'.	This slider will determine the position of the left target sphere and can be used to impose a certain amount of extension of the selected joint.	
19. Connect the Parameter module to the left and right bar Object modules,		
20. Make sure the 'Flexion target (deg)' is connected to the 'Object.ScaleX' of the right bar and the 'Extension target (deg)' to the 'Object.ScaleX' of the left bar.	The scaling of the bar is now set in the Runtime Console. The target spheres at the end of the bar will automatically be positioned because of the connection between the bar and the target.	
21. Note that the bars are extremely long.	The flexion and extension parameters in the Runtime Console are given in degrees. However, the scaling input of the object is in meters. When setting the 'Flexion target (deg)' to '70' will make the right bar 70 meters long.	
22. Open the Connection Editors of the connections between the Parameter module and the bar objects and divide the input by '100'.	An angle of 100 degrees will scale the bar to 1 meter.	

The last step to make the application function properly is providing the 'Real-time' sphere with input. The 'Real-time' sphere should be able to represent three parameters; hip flexion/extension, knee flexion/extension and ankle flexion/extension.

23. Provide the 'Real-time' object with the correct input. Tips: - Use a Switch module to switch input to the Object module (see Part 3; step		

Procedure	Explanation	Illustration
20-30). - The Object should move in the X-direction. - The MoCap module outputs joint kinematics in degrees, therefore the data must be scaled.		
The application is functioning properly. A nice extra feature would be to give the subject positive feedback. We will use a particle to show the subject (and the operator) that the target has been hit. The following steps will show you how to do this.		
24. Drag a Collision module in to the Data Flow Editor.	The Collision module will be used to determine a collision between the 'Real-time' object and the target object.	
25. In the Scene Explorer, drag the 'Left target', 'Right target' and 'Real-time' objects on to the Collision module. You may need to unfold 'CAREN OBJECTS', 'caren.viewer.node', 'caren.viewer.camera.node' to locate the objects.	These stock objects are attached to the camera and therefore are located under 'caren.viewer.camera.node'.	
26. Open the Collision module and set the 'Primary collision object' to 'Real-time'.	The Collision module detects collisions between the primary object and any other object.	
27. Enable 'Send events only once per collision'.	The module will not send another collision event until the current collision has ended.	
28. Create an event in the Global Events section and name it 'Hit'.		

Procedure	Explanation	Illustration
29. In the Collision module, next to Collision event 'True' select 'Hit'.	The module broadcasts 'Hit' every time the 'Real-time' object collides with the 'Left target' or 'Right target' object.	
30. Drag a Particle module in to the Data Flow Editor.		
31. Select the 'CAREN-Fireworks' particle in the dropdown list.		
32. In the Module Properties pane, set the action on 'Play' to None and the action on 'Hit' to Play.	The particle will be played when the 'Hit' is broadcast (i.e. when the 'Real-time' object collides with a target object), but will not be played when the application is started.	
31. Go to 'File | Save application as' and save the application.		

Tips to improve the application:

- **Add a checkbox in the Runtime Console to make the particle effect optional**
- **Add option in the Runtime Console to show and hide the feedback objects.**
- **Add option in the Runtime Console to show and hide 'the subject' walking on the road.**

Hogehilweg 18 - C | 1101 CD | Amsterdam | The Netherlands
T: +31(0)20 301 30 20 | F: +31(0)20 301 30 21 | www.motekforcelink.com