Manual
Heider Simmel Story Generator

developed by Robert Geirhos and Svenja Melbaum
under supervision of
Prof. Martin Butz
University of Tübingen

version 1.0
October 2016

Table of Contents

Introduction

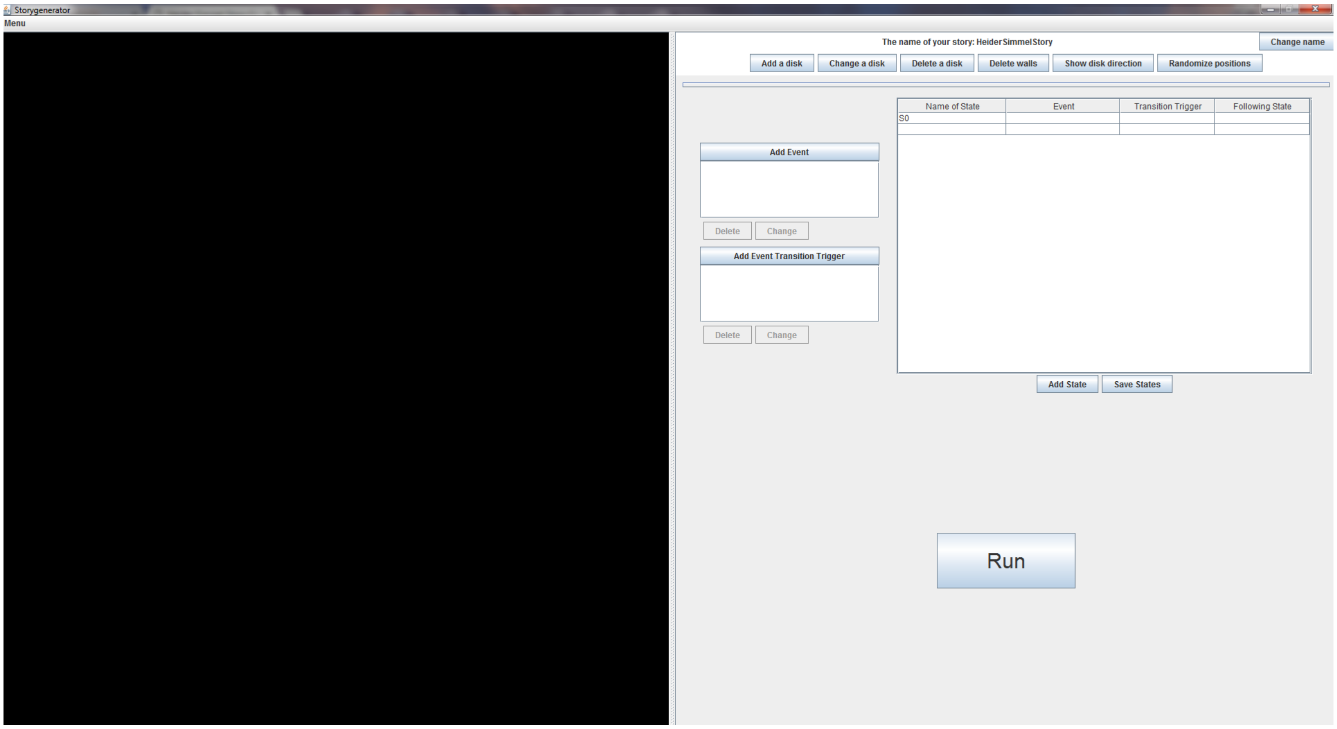
This is a GitHub project for an editor/simulator framework for generating little emotional stories in the style of the classical Heider Simmel experiments.
The idea is that interaction of simple geometric forms with a limited repertoire of movement patterns can lead to the attribution of complex cognitive interpretations.

This project has been initiated by the chair of cognitive modeling at the University of Tübingen.

The stories are composed of *events* that involve circular *disks*, *walls* and their interactions.

Overview

The initial view of the program looks like this:



The name of your story: HeiderSimmelStory

Name of State	Event	Transition Trigger	Following State
S0			

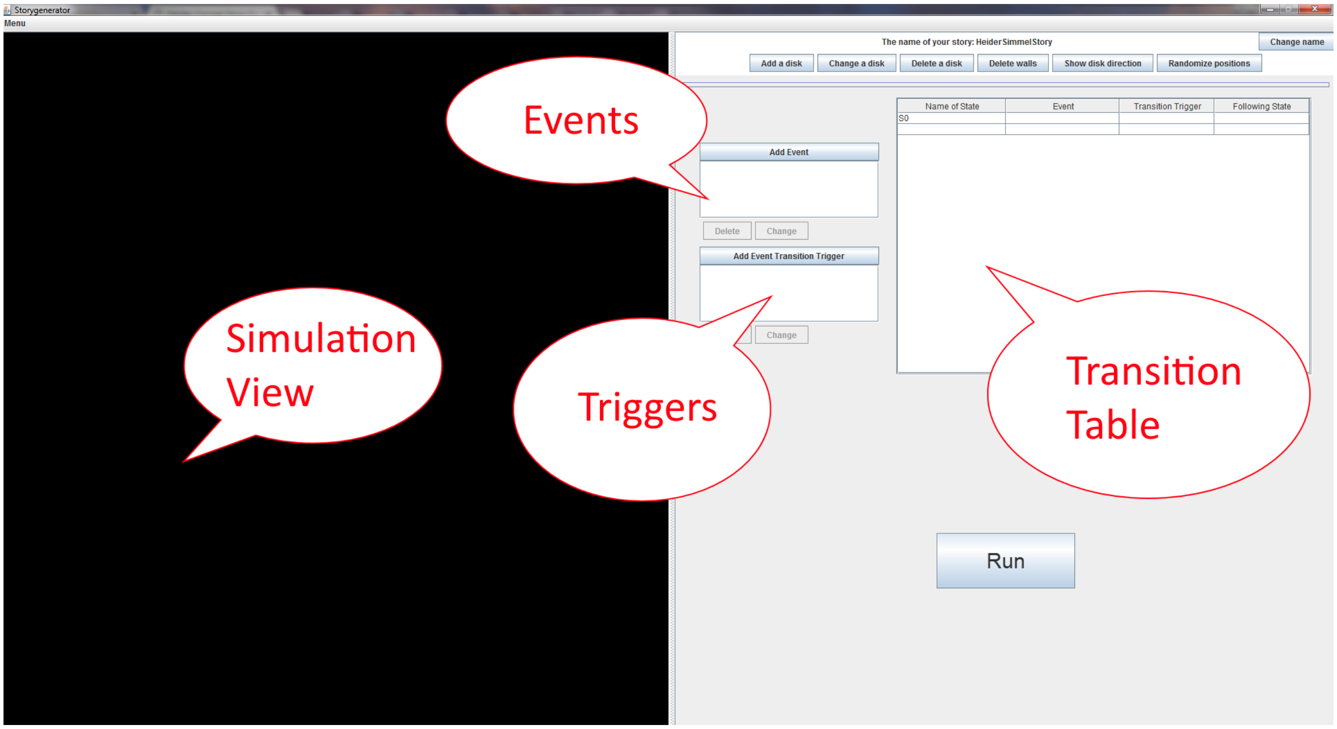
Menu Items

Note the following menu areas:

- The main menu to the top left: Reset and store/load stories
- Story buttons on the top right:
  - Add a new disk
  - Change properties of a disk
  - Remove an existing disk
  - Remove all existing walls
  - Toggle disk direction visualization on/off (only visible when a story is running)
  - Place all disks at random start positions
  - Modify the name of the story
- Run/Stop button on the lower right

Story editing areas

The process of editing a story mainly involves the following GUI elements:



The name of your story: HeiderSimmelStory

Name of State	Event	Transition Trigger	Following State
S0			

Simulation View:

During editing, this part of the window shows the initial positions of disks and walls. After pressing the "Run" button, the simulated story is being displayed here. This area can also be used to

- choose the position and size of new disks (click&drag)
- place walls (click&drag)
- choose path points (multiple clicks)

Event Behavior table:

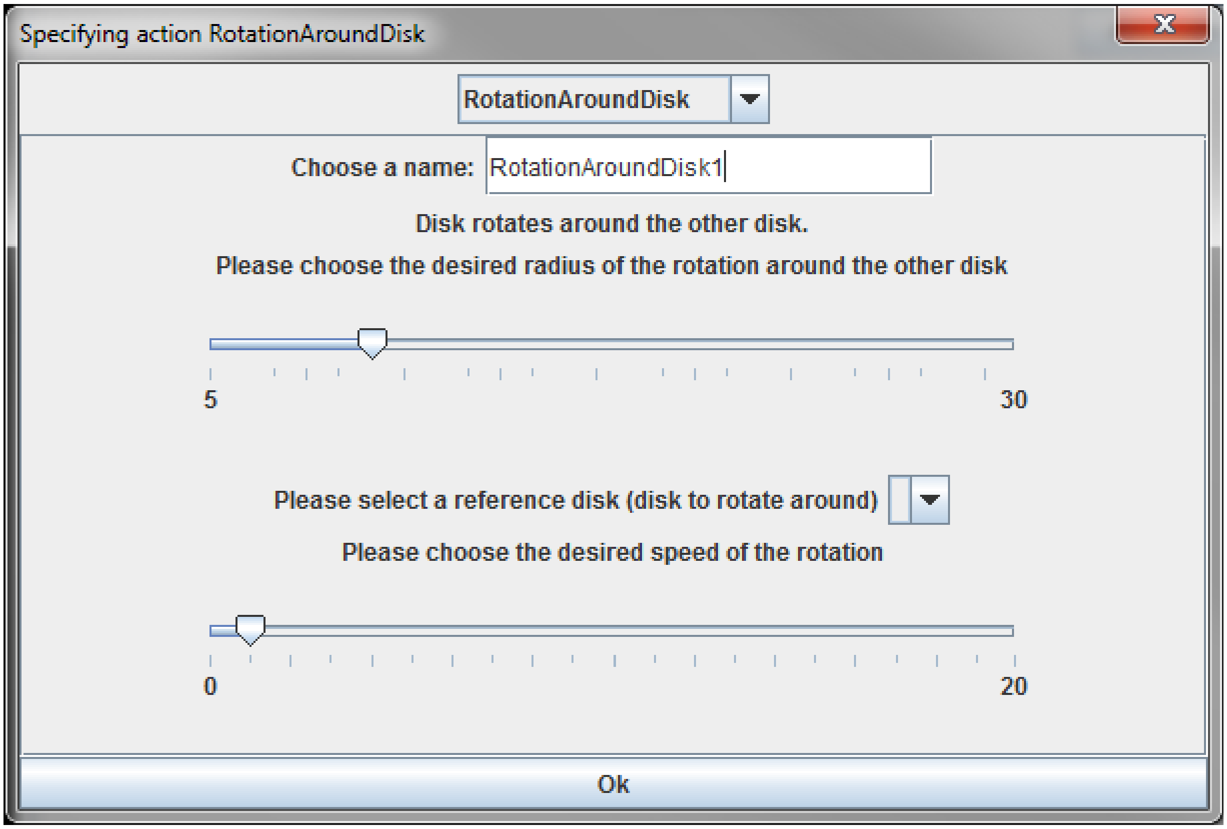
Here all the events of a story are listed. Each event determines the behavior of a disk. The following types of behaviors are currently defined:

- Attack
- Avoidance
- Avoidance and Rotation
- Chasing
- Intelligent Chasing
- Intelligent Trajectory

- Movement to Point
- Path Finder
- Random Movement
- Rotation
- Rotation around Disk
- Stopping
- Trajectory
- Turn To

After choosing the desired type of behavior, a pop-up window appears that allows the setting of parameters that further specify the type of behavior.

Example: Parameter Pop-up for “Rotation around disk”



Specifying action RotationAroundDisk

RotationAroundDisk

Choose a name: RotationAroundDisk1

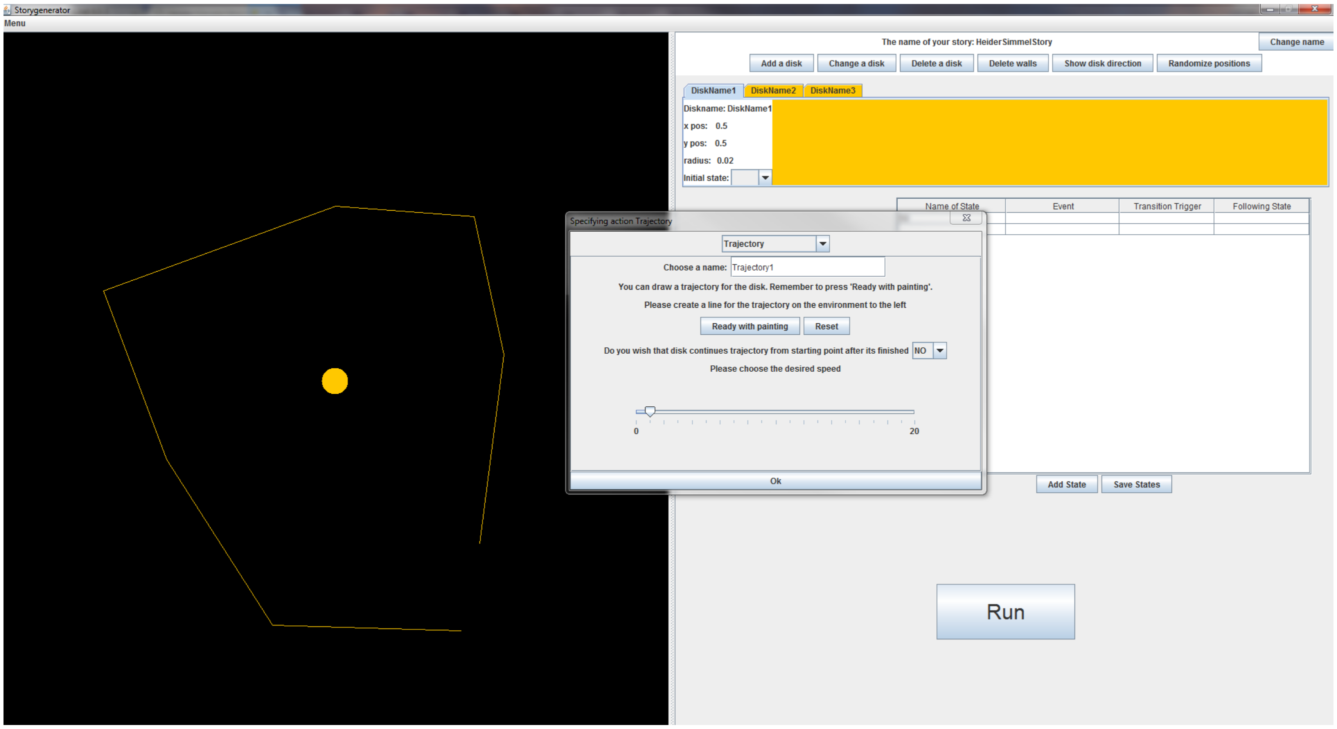
Disk rotates around the other disk.

Please choose the desired radius of the rotation around the other disk

Please select a reference disk (disk to rotate around)

Please choose the desired speed of the rotation

In case of a trajectory, the way points can be selected in the simulation view window:



The name of your story: HeiderSimmelStory

Diskname: DiskName1
x pos: 0.5
y pos: 0.5
radius: 0.02
Initial state:

Name of State	Event	Transition Trigger	Following State

Choose a name: Trajectory1

You can draw a trajectory for the disk. Remember to press 'Ready with painting'.

Please create a line for the trajectory on the environment to the left

Do you wish that disk continues trajectory from starting point after its finished NO

Please choose the desired speed

Event transition trigger table:

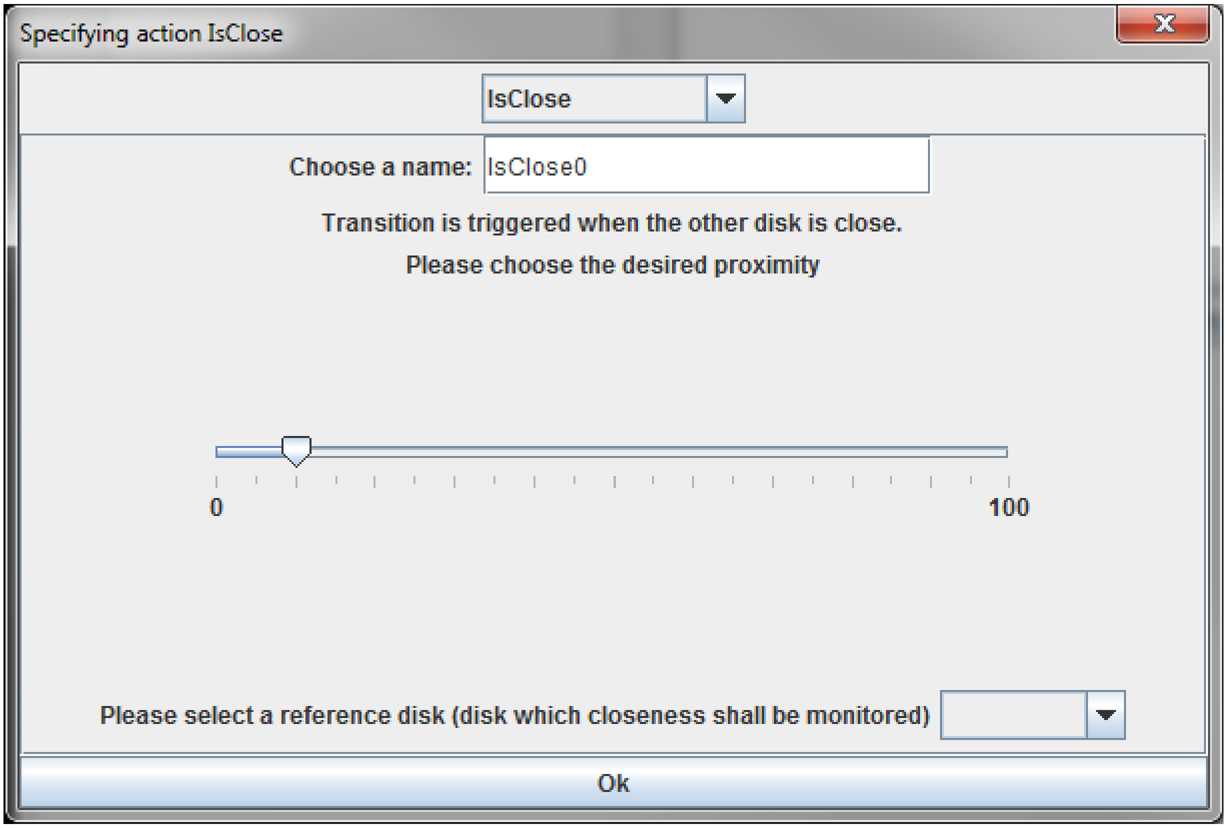
Here all possible conditions for transition between disk states are listed. Currently implemented are the following:

- Always
- BecameInvisible
- BecameVisible
- HasStopped
- IsClose
- IsNotClose
- OtherDiskStopped
- PeriodOfTime
- SameXPosition
- SameYPosition
- TimeSinceStart

- TouchDisk

After the type of trigger was chosen, a pop-up appears that allows specification of parameters:

Example: Parameters of the “IsClose” trigger:



Specifying action IsClose

IsClose

Choose a name: IsClose0

Transition is triggered when the other disk is close.

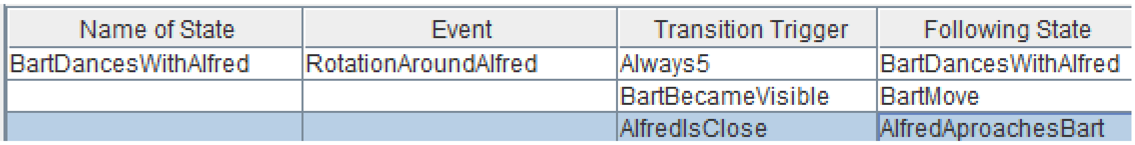
Please choose the desired proximity

Please select a reference disk (disk which closeness shall be monitored)

Transition table:

This is the main component for story creation. In this multiple tables (each having 4 columns), are created that encode the logic of your story. These tables control the simulation and determine in which way the story unfolds.

For each state of the story, there is a sub-table of the following form:

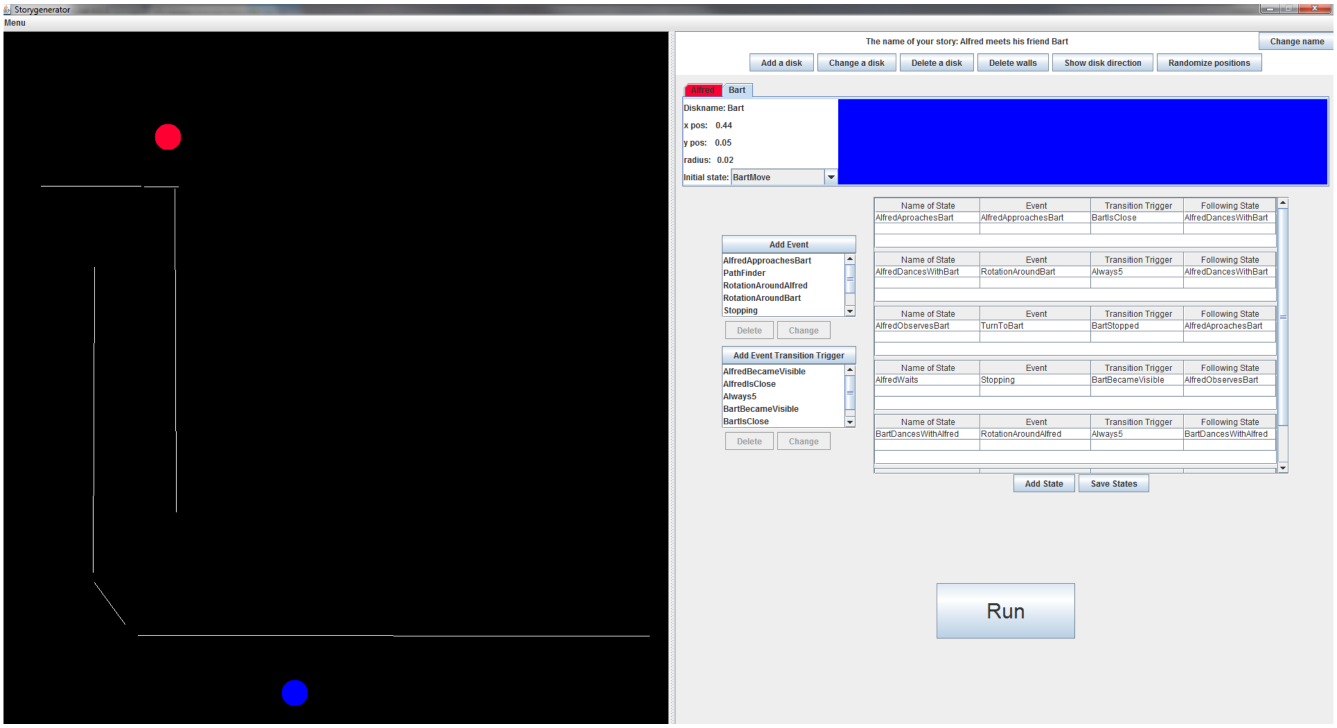


Name of State	Event	Transition Trigger	Following State
BartDancesWithAlfred	RotationAroundAlfred	Always5	BartDancesWithAlfred
		BartBecameVisible	BartMove
		AlfredIsClose	AlfredAproachesBart

For the first and second column of this table, only the first row holds some data: A user-definable name for the corresponding state and an event (behavior) that shall be assigned to this state (one of the behaviors listed in the “event behavior table”).

The third and forth columns may have multiple entries and specify a map from possible transition triggers (listed in the “event transition trigger tabe”) to states. This map determines if an event boundary occurs for a specific trigger condition and which the subsequent event will be.

Example: For the story “Alfred meets his friend Bart” (can be download from our GitHub page), the editor view looks as follows:



The name of your story: Alfred meets his friend Bart

Diskname: Bart
x pos: 0.44
y pos: 0.05
radius: 0.02
Initial state: BartMove

AlfredApproachesBart
PathFinder
RotationAroundAlfred
RotationAroundBart
Stopping

AlfredBecameVisible
AlfredIsClose
Always5
BartBecameVisible
BartIsClose

Name of State	Event	Transition Trigger	Following State
AlfredAproachesBart	AlfredApproachesBart	BartIsClose	AlfredDancesWithBart

Name of State	Event	Transition Trigger	Following State
AlfredDancesWithBart	RotationAroundBart	Always5	AlfredDancesWithBart

Name of State	Event	Transition Trigger	Following State
AlfredObservesBart	TurnToBart	BartStopped	AlfredAproachesBart

Name of State	Event	Transition Trigger	Following State
AlfredWaits	Stopping	BartBecameVisible	AlfredObservesBart

Name of State	Event	Transition Trigger	Following State
BartDancesWithAlfred	RotationAroundAlfred	Always5	BartDancesWithAlfred

In this story disk A (“Alfred”) finds his way through a little maze. When its exists the maze, it gets noticed by disk B (“Bart”), which follows its path curiously (make sure to press “Show disk direction” BEFORE you press the “Run” button, in order to see in which direction the disks are “looking”).

When A stops, B approaches carefully and after recognizing each other, both disks run merrily around on a circular track.