

# .raindrops

## A flowerful installation with body and sound interaction using camera tracking

Deborah Schmidt, Brian Eschrich, Martin Herrmann, Jan Wojdziak, Rainer Groh

**Abstract:** In the last few years new ways of communication with the computer have been invented. Unlike popular possibilities for using the hand as an input device like multitouch on mobile phones this project features whole-body use for interaction. A person tracked by a camera is able to interact with a virtual world projected on a screen in life size creating a direct connection between him and his digital counterpart. The aim is to water abstract plants with raindrops coming from the top of the projection. Furthermore the installation stays awake even without a person acting in some way. It uses random factors and also environment sounds to create a living world. Bystanders can interfere the performers action by making sounds which is also tracked and reflected by the application.

### 1 Motivation

Unconventional input devices for modern computers are well-liked nowadays. Especially within games or any fun based interaction developing the spirit of using experimental methods for interactive control is high. That could be based on the more simple target of games – you do not want to manage a certain task, you want to have fun in a way the game decides. Unexpected effects enhance the character of the game – it follows its own rules and the challenge is to cope with that.

A great way to increase the agility within interaction concepts is the possibility of using the whole human body as the corresponding medium between digital and analog systems. It provides the intuitive facility of direct influence on digital objects contrary to the usage of tools like the computer mouse or a joystick. In addition the actor is also not just able to control one single cursor point, his contour expresses an entirely different source for digital data manipulation. The body can for instance be tracked with a cam connected to the computer. Exemplary realizations exist in very complex systems who are able to identify different parts of the body or in a simpler way of just getting the outline of moving objects. One of the project aims is to find a suitable tracking algorithm in terms of its accuracy and the ability to represent a solid, coherent entity of a human body.

This project was also inspired by [[@THESYSTEMIS](#)], where the actors are playing with visual effects surrounding their bodies on a screen behind their backs and modifying them by making sound with varying intensity. Todd Vanderlin figures another precious source for creative ideas using camera tracking methods. He did a great effort adapting Box2d to openFrameworks (cf. chapter 3, [[@VANDERLIN](#)]).

### 2 Concept

The aim of the design concept follows two main aspects: It should be designed for exposing it on a big surface to involve various spectators and it should not freeze if no one is currently willing to interact with it. For this reason the concept follows metaphors from the nature outside like plants and raindrops, which are just fine without human intervention.

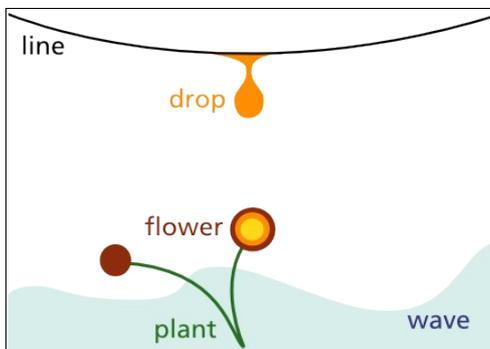


Figure 1: Basic construction of one level.

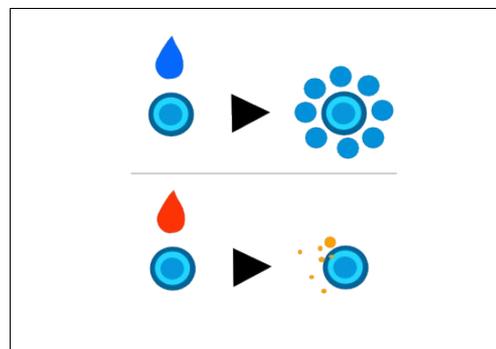


Figure 2: Concept of flourishing plants.

There had to be a goal to catch the performers attention and a plain way to reach it, because this should be an installation where you can walk by, have a look and try it out without the need of endless setting in or reading manuals. In this game concept you need no mouse or keyboard to interact. Through the image of a standard web cam the shade of every physical object moving in front of it is being tracked in order to manipulate everything on the screen. The world of Raindrops contains plants and raindrops which can be moved (fig. 1). Plants consists of various flowers. Raindrops are constantly dropping from a line hanging on the top of the screen. Different levels vary in the number of plants. The performer has to water each flower on the screen to win a level (fig. 2). This is reached by carrying a raindrop to every flower using the shade captured by the camera and visualized in the digital world as a shadow forming a barrier.

Some features provide a certain degree of difficulty. There are different color sets per level determining the coloring of the plants and the water dropping from the line. A flower can only flourish, if it is hit by a similar colored drop. In case of a mismatch the raindrop will crush. If the drop hits no plant until it reaches the bottom of the screen, a wave in the background will rise. One must prevent this wave from passing a critical height which would result in losing the level. Furthermore any bystander can interfere the performers tryings by making sounds like hand clapping or shouting. This will create wind swirling up the plants, visualized by tiny abstract butterflies crossing the image.

The level backgrounds have the design aspect to not complicate the view through the usage of more abstract forms and cozy colors fitting to the flower color sets as a feast for the eyes (fig. 3). As it will be described in chapter 4.3, it is easy to add background images yourself.

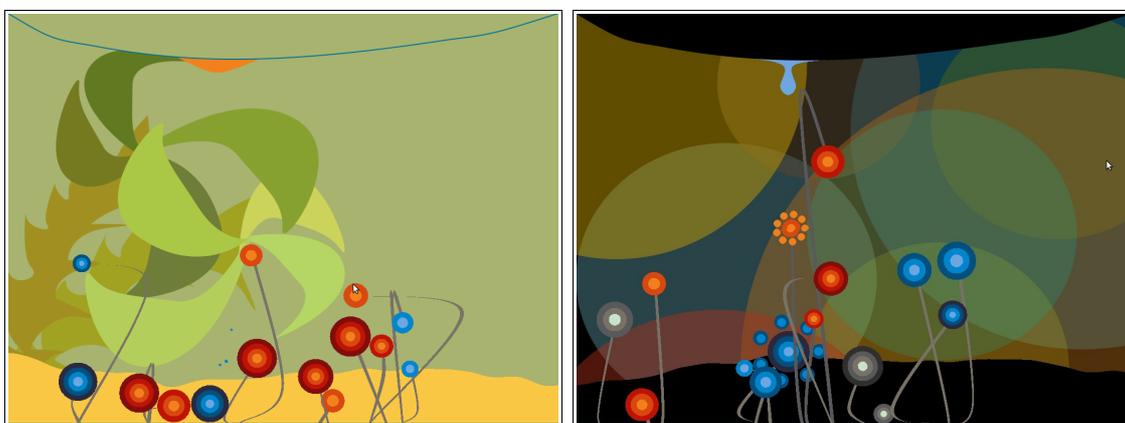


Figure 3: Different level sets.

### 3 Implementation

In practice this project requires the accessibility to a plugged in camera, image processing tools and an efficient physics engine with the opportunity to be free in questions of composition.

This project was simultaneously developed under Mac OS, Linux and Windows. Working with openframeworks, an open source C++ toolkit for creative coding [[@OPENFRAMEWORKS](#)], this caused some trouble finding a development environment able to use it in combination with plugins. After failed attempts with Eclipse, Visual Studio and Qt we finally were able to work with Code::Blocks [[@CODEBLOCKS](#)] under Linux and Windows. For Mac OS Xcode [[@APPLE](#)] is a working option.

The Box2d [[@BOX2D](#)] physics engine is used to simulate the interaction of and with the game elements. The engine provides a variety of different basic forms including circles, squares and polygons. These forms all have physical properties like friction or mass and interact with each other in a so-called "Physics World" which provides the real-world attributes like gravitation. One can connect the basic shapes with "joints". These Elements create an elastic until solid connection between the objects, depending on the parameters.

To provide a natural motion of the plants, they use itself the physics features. They are physical objects, attached to an elastic rope like joint, which is attached to a fixed object on the ceiling. So the plant can wave in the virtual wind, react in other ways with the environment and finally gets back in its original position. The raindrops consist of "Softbodies" developed by Todd Vanderlin [[@VANDERLIN](#)] being responsible for their semi-fluid appearance. These bodies are a composition of basic shapes and joints arranged in an outer and

inner circle with a single basic shape in the middle. The other rings are connected with solid joints and the center shape is linked with the inner ring by elastic joints. A force applied at a point causes a movement of the other points and so the shape seems like a soft form (fig. 4).

The "Polygon Man" is the representative of the human interactor in the game. In order to create this shadow-like shape the contours of the human body had to be extracted from the web cam picture. This is done by calculating the color difference of a static background picture taken without the player and the current camera picture with the player. After thresholding this difference picture the contour can easily be extracted of the resulting binary picture. This contour can now be added to the physic engine to enable the player to interact with the game objects. To improve the collision behavior of the raindrop "Softbodies" with the "Polygon Man" the shape is filled with polygons.

In order to provide a comfortable way to build levels yourself, the level structure is extracted in an XML file which setup will be explained in the next chapter. The very fast XML parser RapidXml [[@RAPIDXML](#)] provides interpreting the file content in C++.

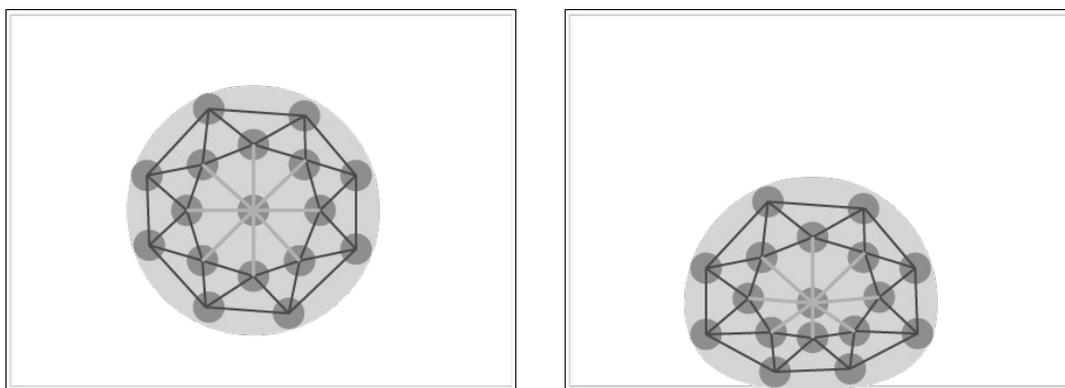


Figure 4: The mechanics of a "Softbody" (cf. [[@VANDERLIN](#)]).

## 4 Raindrops in action

This chapter contains important instructions to interact with the implementation of the described installation concept. It expands on the buildup of the installation, useful shortcuts to make it work as expected and possibilities to put own ideas into practice.

### 4.1 Construction

At first there has to be a camera connected with your computer. The best results for the detection of the human shape can be achieved if the player stands in front of a uniform background in a well lighted room. Also the automatic brightness adjustment of the camera should be disabled.

The output is separated into two parts. The left side is the actual game area which size can be customized, check therefor chapter 4.3. For maximal interaction fun it is recommended to project this on a large screen so that the shade displayed in the installation matches with its real size (fig. 5). A second monitor could be used to show the right side of the program output which contains the camera input image and the calculated shade of moving objects in front of the camera. That helps to detect possible error sources if the shade tracking is not working properly.

### 4.2 Usage & calibration

To run the application, run 'clickToLaunchApp.sh' in the root folder of Raindrops on Linux, run 'raindrops.exe' on Windows and double click the app folder on a Mac.

Taking every moving object out of the cameras sight and pressing the SPACE key is essential to initiate a level. Afterwards the shade of anyone stepping in front of the camera should be displayed correctly on the installation screen. There are lots of tuning options to compensate troubles with difficult backgrounds or lightning like for example morphological operations and additionally some cheating keys being assistant in level testing cases. A complete shortcut list can be found at the project website [[@RAINDROPS](#)].

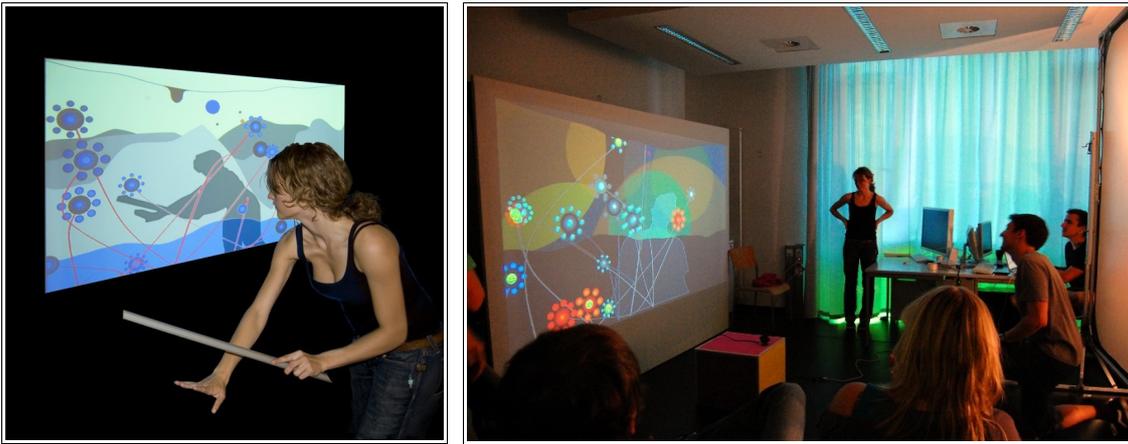


Figure 5: Raindrops in action [MOSCH].

### 4.3 Customizing

As mentioned before, the level structure is defined by an external XML file. Its composition will now be explained (fig. 6). This files location in the root folder of the application is

`data/level.xml`

and it can also be used to set the screen size, in this example in line 2. Adding a new level one must increment the level id (line 3). In line 4-7 the color sets of the plant branches are defined. It is possible to choose between 'blue', 'red', 'green', 'brown', 'grey' and 'colored'. The plants itself are placed in line 8-12, in this example there are three plants. For each plant you can set a maximal width and height. Additionally the parameter 'x' sets the horizontal position of the plant and 'branches' is defining the count of flowers growing out of this plant. Their colors depend on the raindrop color set, specified in line 13-15. One can also choose multiple color sets. To set the colors of the raindrop line, the background, the wave in the background and the wind animations, just edit their RGB values in line 16-19. To use a background image, add its source to the 'src' parameter in the background tag. The path is relative to the 'data' folder where the data.xml file is placed either.

```

1 <?xml version="1.0" ?>
2 <game width="1024" height="768">
3   <level id="1">
4     <branch>
5       <color>brown</color>
6       <color>red</color>
7     </branch>
8     <flower>
9       <item x="170" height="150" width="130" branches="1"/>
10      <item x="270" height="400" width="130" branches="2"/>
11      <item x="450" height="550" width="70" branches="3"/>
12    </flower>
13    <drop>
14      <color>blue</color>
15    </drop>
16    <line r="201" g="0" b="124"/>
17    <background r="171" g="171" b="171" src="img/grey.png"/>
18    <water r="94" g="94" b="94"/>
19    <fly r="255" g="139" b="210"/>
20  </level>
21 </game>

```

Illustration 6: Sample level.xml file.

## 5 Discussion and Future Work

This work describes the theoretical concept of the installation ".raindrops". The practical usage including the construction, calibration and customization is also mentioned. The concept provides a two dimensional world, consisting of abstract plants. The user in front of the screen is able to interact with this world by using his body symbolized through a shadow on the wall. To continue to the next levels the user has to water the plants. The system also reacts on sounds in order to create airflows moving the flowers.

In addition building own levels is already possible. This feature could be expanded by defining and style barriers, leading to more complex surfaces and higher creative quality. The shape tracking could be optimized as well. This process is still depending on the lighting conditions and the background environment. Other types of cameras, for example a depth-camera could be used for the tracking to receive better results. Another challenge is to map the installation to wider screens, like a complete wall, by using more than one camera and beamer. A realization in three dimensional space by using stereoscopic projection and tracking methods would provide a complete different impression.

Raindrops started as a one week project with still a lot of possibilities to improve. But yet at this stage of implementation one can recognize the appeal of this playable flower meadow. The fascination of having direct influence on digital objects with a human body guided this project to a pleasant application forcing a number of people to stop by and step in front of the camera making rather amusing gesticulations.

Please visit the project website at [[@RAINDROPS](#)] for further information.

## References

- [MOSCH] Pictures taken by Marc Mosch.
- [@APPLE] Apple. Xcode, distribution website. Online available at <http://developer.apple.com/technologies/tools/xcode.html>, reviewed at 2010-09-01.
- [@BOX2D] Erin Catto. BOX2D. Online available at <http://www.box2d.org/>, reviewed at 2010-08-31.
- [@CODEBLOCKS] Code::Blocks, project website. Online available at <http://www.codeblocks.org/>, reviewed at 2010-09-01.
- [@OPENFRAMEWORKS] openFrameworks, project website. Online available at <http://www.openframeworks.cc/>, reviewed at 2010-09-01.
- [@RAINDROPS] Deborah Schmidt, Brian Eschrich, Martin Herrmann. Raindrops, project website. Online available at <http://raindrops.at-random.de>, reviewed at 2010-09-02.
- [@RAPIDXML] Marcin Kalicinski. RapidXml, project website. Online available at <http://rapidxml.sourceforge.net/>, reviewed at 2010-09-01.
- [@THESYSTEMIS] Zachary Lieberman & collaborators. MESSA DI VOCE. Online available at <http://thesystemis.com/projects/mesa-di-voce/>, reviewed at 2010-08-30.
- [@VANDERLIN] Todd Vanderlin. Design & Technology. Online available at <http://toddvanderlin.com/>, reviewed at 2010-09-02.