Mini Project

Elizabeth Dwenger - 40141718

City University of Hong Kong

CS4187: Computer Vision for Interactivity

Kening Zhu

December 16, 2022

Introduction

The application created uses the contour of the image captured by the camera to create obstacles for objects which can be dropped by the user. The application explores how the physical world can interact with the digital world using edge detection. Augmented reality is when the digital is able to react to the physical world (Bonsor & Chandler, n.d.). I am very interested in augmented reality, and this project acts as a very basic example of this.

Description of Features

Within this application, there are a couple of features that add to the diversity of the program; these features are also implemented in such a way that for most users, it should be intuitive how to use the application. These features include multiple objects which can be dropped, a way to draw lines yourself, a method to reset the contour finder, and methods to toggle whether the objects are affected by the contours. The objects which can be dropped include a circle, a block, and a custom object, which is currently set as an image of a cat. Due to the use of ofxBox2d, and the methods it includes, each block has a different effect within the world, especially when it comes to the gravity and movement of the object. Furthermore, lines can be drawn by the user to add to the obstacles of the objects, as well as to add to the interactivity of the application.

In order to make these features more intuitive, in the upper right-hand corner where the image from the camera is, there is also a legend that describes the possible keys which can be pressed to make something occur. The keys which were chosen use the first letter of what is attempted to be completed in order to make this process a bit easier. For example, to drop a ball, the key that needs to be pressed is ‘b’, whereas to drop a square, the key is ‘s’, and ‘c’ drops a custom particle (in this case, a cat).

The first feature mentioned in the legend is the ability to draw a line. In order to extend the purpose of the application, I decided to add the ability for the user to draw on the screen by pressing the ‘l’ key. This allows the user to add theoretical obstacles to the environment to explore how this would change the interaction between the digital environment, and the physical environment of the contours. The second feature mentioned in the legend is a toggle for the mouse force. Essentially what this does is allows the user to attract all objects to the position of the mouse so that the objects can be redistributed. The next three features are varying objects which can be dropped, which as mentioned, are the ball, square, and custom particle. Each has slightly different parameters, and of course, shapes, which allows the user to experiment with different objects in the environment. The next item on the legend is that pressing the ‘space bar’ resets the background for contour detection. This is intended to allow the user to change the background and still have the objects placed in front of the camera to be detected. The final items on the legend are just the number of objects and frames per second; these are not really interactable, but still give valuable information to the user.

Along with what is mentioned in the legend, there are a couple of other features which are in the application. Pressing the screen resets the contour which is followed by the objects if the objects have been set as idle. If there is no movement after a couple of seconds, the objects are set as idle, which makes them not move. Clicking the screen resets this, and makes the objects ignore the contour until the screen is clicked again. These are all of the features in the application.

How to Use Application

The application can be used in multiple ways, so how the application is used depends on the desired outcome of the user. The application uses mouse-based interaction, keyboard-based interaction, and tracking-based interaction. The main intended use of the application is to use the contours/lines of the image captured by the user's camera to act as a barrier for the objects which can be dropped, which acts as tracking-based interaction. In order to achieve this, the user must first set the background by pressing the 'spacebar', and then entering the frame and clicking the screen to trigger that the contour is a barrier. Clicking the screen either triggers the contours to act as barriers or to no longer act as a barrier. In order to drop an object, the user can move their mouse to the desired location of the object, and then press either 'c' for a circle being dropped, 'b' for a block being dropped, or 'z' for the custom image (in this case, a cat) to be dropped. Pressing 'l' allows the user to draw on the screen, whereas pressing 'f' toggles the mouse force, which allows the user to move all objects on the screen to another destination.

Implementation and Demonstration

In order to facilitate the edge detection, ofxCvContourFinder was used as opposed to another source due to the fact that this method allows for the storing of the contour in a vector which makes interacting with the contour much simpler. The implementation of the contour method was assisted by an OpenFrameworks article that explained how to use ofxCvContourFinder (ofxCvContourFinder, n.d.). This method works by comparing the contrast between pixels and is made even more efficient with the use of foreground detection which reduces the amount of computation that needs to be done by comparing the current image to the stored version of the background. This edge detection was also used for the drawing aspect of the program, as the edges drawn by the user are detected to be contours, that can then be interacted with by the objects.

The objects are implemented using ofxBox2d. This is a wrapper for openFrameworks which uses box2d, an open-source 2-dimensional physics simulator, to simulate ways in which objects move in a realistic way (Catto, n.d.). There are many parameters which can be set using this library; the physics in the objects I used is a bit smoother and has less gravity than how objects interact on Earth. The reason I did this was that I preferred the way the objects moved this way. Using this library made the goal of dropping objects much simpler and resulted in a more refined-looking project.

Link to the demonstration: https://www.youtube.com/watch?v=yX6BN34gr2s

References

Bonsor, K., & Chandler, N. (n.d.). How Augmented Reality Works. *How Stuff Works*. https://computer.howstuffworks.com/augmented-reality.htm

Catto, E. (n.d.). Box2D Documentation. https://box2d.org/documentation/

ofxCvContourFinder (n.d.). *OpenFrameworks*. https://openframeworks.cc/documentation/ofxOpenCv/ofxCvContourFinder/

Perevalov, D. (2013). *Mastering openFrameworks: Creative Coding Demystified.* : Packt Publishing Ltd.