

S3doodle: Case Study for Stereoscopic Gui and user Content Creation

Diego González-Zúñiga, Universitat Autònoma de Barcelona, Spain

Jordi Carrabina, Universitat Autònoma de Barcelona, Spain

Abstract

This article describes the web application called S3Doodle. It is an initial example of the combination of a stereoscopic graphical user interface that produces stereoscopic content and is controlled by hand gestures. The web application uses core stereo 3D concepts and implements them with cutting edge technology to create a differentiated experience. We use a spatial input control as an interface that allows us to introduce depth into the created compositions. S3Doodle's creation is documented, and the experience we had while building it is explained, shedding light on which are the technical decisions behind. We also gather user's comments from the utilization of the web app. This application serves us as a case study for exploring gestural and stereo interactions, from which we get initial reactions upon which we can provide a better experience for future applications.

Keywords: Stereoscopy, 3D GUI, doodle, human factors, New Media Art, 3D application

Introduction

Stereoscopic 3D is an area that has been generally reserved for movies and some videogames. It works in these mediums because it is able to create a sense of immersion that adds emotional value to the stimuli the user is being exposed to. Even though many criticism for this technology is related to the lack of available quality content and other hardware related implementations, it still represents a growing market. In China, the number of 3D screens in 2014 is close to 23000 and is expected to almost double in just 4 years (International 3D & Advanced Imaging Society, 2015).

Television set size is increasing and the advances in imaging technologies like UHD (4K) and HDR capabilities set an interesting panorama for future stereoscopic content. But there is still an area that has seen little to none experimentation and development regarding stereoscopic features: that of applications. This is due to the unavailability on one hand, of hardware to display these apps and on the other, of development platforms that allow developers to create applications that take advantage of depth. While many challenges appear when including 3D depth in a graphical user interface, we wanted to develop a simple application that would not only use depth in its GUI, but also create stereo compositions. This article describes the resulting application.

Related Work

We research how stereoscopic depth affects a GUI. Similar research has been done where stereoscopic depth is applied in mobile applications like a phonebook to not only improve the UI design but also feature a utilitarian case, as shown in (Häkkinä, Posti, Koskenranta, & Ventä-Olkkonen, 2013). In the same vein, the study of emphasising content using depth done by Huhtala et al. (Huhtala, Karukka, Salmimaa, & Häkkinä, 2011) and using depth for GUIs in automotive dashboards (Broy, André, & Schmidt, Is stereoscopic 3D a better choice for information representation in the car?, 2012), reveal the validity and promising approaches of the stereo format.

Another reason why we are interested in this is because we know that we are not only able to change the gaze pattern of a specific stimulus related to a simple task like web search (Gonzalez-Zuniga, Chistyakov, & Carrabina, 2014). Also in areas like gaming, 3D stereo is perceived to be more enjoyable and immersive than 2D viewing for tasks that provide an advantage in 3D stereo (Kulshreshth, Schild, & LaViola, Evaluating User Performance in 3D Stereo and Motion Enabled Video Games, 2012). It can also alter the behaviour of a user playing a videogame, regarding decisions and actions they perform (Schild, LaViola, & Masuch, 2014). Finally, we took into account several proposed guidelines to S3D GUI creation found in (Broy, Schneegass, Alt, & Schmidt, 2014), (Schild, LaViola, & Masuch, 2014) and our own experience to build our application.

State of the Technology

When trying to develop a stereoscopic application, the most important thing to accomplish is the two different views that need to be fused (Puell, 2006). While most game and graphics engine platforms include support for 2.5D environments like Unity, DirectX and OpenGL, trying to recreate a dual view with different parallaxes is not straightforward, nor does their use come without a steep learning curve. Because of this, we looked into developing a tool that would allow us to create stereoscopic stimuli for our experiments. Our objective was to study the perception of depth in a graphical user interface. We developed a JavaScript library that allowed to draw using the '2d' context of the HTML5 Canvas. By using sets of pairs of

canvases (González-Zúñiga & Carrabina, 2015), we were able to create stereoscopic compositions in real time that could be interacted with. The tool has the advantage that is based in web technologies, so it has broad device compatibility and any JavaScript coder can easily incorporate it to his toolset.

We looked into available motion controllers to implement interaction. Our objective was to take advantage of depth by incorporating it not only in the GUI but also in the final composition created by the user. We found the LEAP Motion (LEAP Motion, 2015) to be good for our needs, since we could accurately track a user's hand's position and draw accordingly in the HTML5 canvas. With these to small pieces of software and hardware, we were able to create S3Doodle.

Application

After developing the stereoscopic drawing tool, we set to create an application that could serve as a case study for the inclusion of depth in a GUI. We looked for a task that would be simple, focused and isolated, because it would have the largest potential to increase user performance from the applied S3D (Kulshreshth, Schild, & LaViola, Evaluating User Performance in 3D Stereo and Motion Enabled Video Games, 2012). We choose to reproduce a simple painting application with 4 commands. In order to give depth a utilitarian function, we placed the menu closer to the user, creating a separation between the drawing area and the available tools. This way we wanted to mimic the opacity that some bitmap editors apply to create the separation from the drawing canvas and the tooling. It would also make the user pay immediate attention to the content that was out of the screen.

Graphical User Interface

To keep the GUI uncluttered, we just included a button for each available action, and a dark canvas to paint on. No menus or additional content was included. Each button was positioned centered on the bottom of the screen next to the other buttons, creating a toolbar that has an offset of 4 pixels and sits out of the screen. A monoscopic version of the menu can be seen in Figure 1.

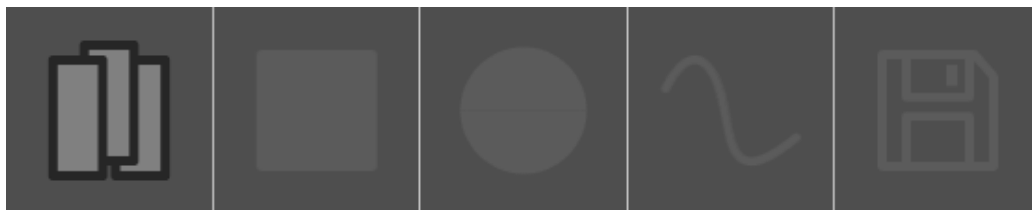


Figure 1: Menu buttons in the interface.

The available actions that we included in the application are: (i) colour selection, (ii) square drawing tool, (iii) circle drawing tool, (iv) free-hand drawing and (v) saving the composition as a PNG file. Since we are working with a graphical user interface that uses depth information to emphasise the menu, we decided to use depth also for the drawing. This represents a challenge since it requires a way of drawing in 3D, so we looked into a spatial input peripheral that would give us information to draw in all 3 axis of the Cartesian plain. We discarded the mouse and keyboard as the main hardware interfaces and opted for the LEAP Motion, because it provided the 3D positions for the hand, including palm and each separate finger. It also provided a JavaScript application programming interface that allowed us to easily incorporate it with the S3D canvas drawing toolkit. The API included several basic built-in gestures that we used to map actions.

We created a cursor based on the hand that would not only draw but also display contextual options to the user. Figure 2 shows the cursor we implemented, which uses the palm as the main cursor and each finger as a possible option when over a menu.



Figure 2: S3Doodle Cursor

The LEAP Motion device detects several hands, and whenever a new hand is detected, it is drawn on the screen, but only the ‘oldest’ detected hand gains control of the application to draw. Moreover, we mark the index finger on the hands with a stroke to indicate that it is the main inking ‘device’ on the cursor.

Interaction

In order to create a more “intuitive experience”, that is, an experience that we would find more natural to use, and taking advantage of the hand gestural recognition device we chose, we decided to assign the index finger as our main pointer for drawing. We wanted to call upon joint attention, the one we use when we are infants to learn (Mundy & Newell, 2007). We want to direct link this attention to the pointing gesture performed with our index finger, which is the one we enable for doodling in the web app.

Because of this, when using the application, the interaction is gestural. We use the LEAP peripheral to track the hand and then draw on the canvas. The cursor is located in the centre of the palm and fingers enable different actions depending on the selected tool. This way, if the palm is over the colour selection button, each finger is painted in a different colour, and ‘key tapping’ selects this colour for doodling. If we select the rectangle tool, then ‘key tapping’ defines the first point of the diagonal that defines a rectangle, and so on. The “KeyTap” gesture is defined as “recognized when the tip of a finger rotates down toward the palm and then springs back to approximately the original position, as if tapping. The tapping finger must pause briefly before beginning the tap”. This is a discrete gesture.

Since we set ourselves with creating an intuitive experience, we looked into free hand menu techniques for distant displays and decided that we would create a new cursor base in the “Marking menus” concept found in (Bailly, Walter, Müller, Ning, & Lecolinet, 2011) varying the layout of the options to accommodate them in each finger. We gave the cursor stereo depth according to the position relative to the motion sensor, so the hand seems to go into and out from the painting area, allowing the user to doodle “in space”. With this, both the cursor and the menu are taking advantage of depth as a utilitarian feature, to highlight the tools and to provide a visual cue of the Z axis position of the hand. Figure 3 shows the usage of the application. (Although the screen it is displaying in is not stereoscopic).

Generated Content

The generated content is a side by side composition of shapes. The free hand drawing consists of a frame by frame stamp of circular elements. This creates a line that complies with the stereo effect by applying each circle a different horizontal offset based on the position of the hand relative to the motion controller. The created composition has the same size as the drawing canvas, and the drawing canvas size is created dynamically based on the chrome of the parenting window/frame. The application lets the user save their compositions as a PNG file. Mouse and keyboard are required in this version to specify the filename and save location since it is managed by right clicking and then selecting “Save As” from the browser window. We have created content and viewed it in FullHD stereoscopic passive and active displays, and on 5” display phones like the Motorola G using Google Cardboard (Google, 2015).



Figure 3: Using the S3Doodle to draw a stereo 3D snail

Technical Implementation

The application is created using standard and recommended web technologies. The core of the application is HTML5 Canvas and its JavaScript ‘2d’ rendering context. Figure 4 shows the structure of the application. The HTML page hosts two pairs of canvases that correspond one the drawing surface, and the other pair to the menu drawing. It also includes a reference to the SXS3DCNV library (González-Zúñiga & Carrabina, 2015) that enables the stereoscopic rendering.

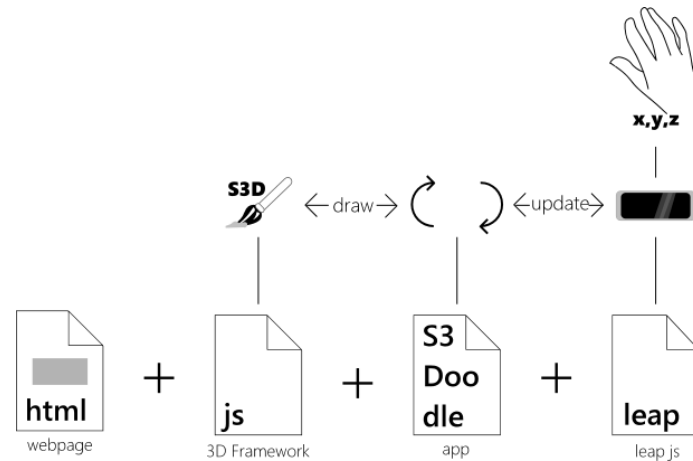


Figure 4: Application structure

The application works with a game loop structure (update/draw). This loop is implemented using a `RequestAnimationFrame` feature of modern browsers and polls the availability of hands and their position to the LEAP JavaScript API. The reason to do this is to only draw when the browser is ready to do so, and when the browser is active. This creates a smoother animation since it uses a consistent frame rate. It also makes the processor able to handle other tasks while rendering the animation, to the point where it is capable of determining a frame rate that works with other tasks that is currently working on, boosting performance. It uses this information to then draw the position of the cursor and ink in a stereoscopic format. All position values, on the X, Y and Z axis are converted to “screen coordinates” since they are relative to the LEAP device, which marks the origin at the centre of the infrared glass panel, as shown in Figure 5. We modify these coordinates with a multiplier to compensate for the different resolutions the application can be shown in (to make it adaptive and independent from screen resolution) and the half resolution loss from the selected side by side format.

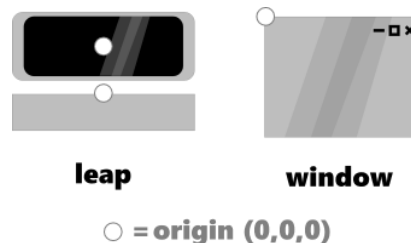


Figure 5: Comparison of origins for coordinate systems

When the user selects a command by key tapping, the app polls the cursor’s position and gives instructions to the user how to proceed, whether by indicating the mappings of menus in each finger, or by inking according to the selected tool.

The application is based in layered stereoscopic 3D. It does not feature perspective nor orthographic shapes. This is to better simulate a blank linen on where to draw, instead of an open space to model volumes.

Limitations

The app has limitations, as the technologies that it is built upon are novel and even the web standards that it leverages have different implementations across browser vendors. We identify technical and usability limitations in this example application. Technical limitations we identified are:

- Speed of processing spatial input data: having to poll position data and repaint every frame is processor consuming.
- Dual screen painting: Since we are using the '2d' context of the HTML5 canvas we must paint every single scene twice. This has an impact on performance.
- Infrared interference: Depending on the displaying method some active infrared glasses may cause interference with the LEAP's infrared. This might result in either losing the stereo capability or an inaccurate hand tracking.
- The two lower corners of the composition are unreachable. This is due to the motion sensing area that the LEAP cannot see.

Following we list the usability limitations. These have been detected by comments made by users.

- The position of the menu in the lower part of the UI makes it difficult to reach the buttons located in both edges of the menu. This is due to the motion tracking area of the LEAP motion.
- The "keytapping" gesture generally ends up selecting different options than the one intended by the user. This is due to several fingers moving alongside the one that represents the option.
- It is not unusual to lose the tracking area. Since the attention of the user is in the screen, many errors occurred from moving the hand away from the sensor's reach.
- It is hard to stop inking in certain parts of the screen. This comes as a result of the two previous logged limitations.

Conclusion

We developed the S3Doodle web app as a proof of concept of an application featuring a stereoscopic graphical user interface. It has gestural input and can create stereoscopic content compatible to with stereo displays. It adapts to different screen layouts and resolutions and Additionally, the most important finding, apart from the novel stereo/gesture implementations running in a browser, is that the users express that depth effect becomes much more powerful in the created compositions than in the GUI itself. This focus on the content is something that we will take into account for future development of applications and research experiments. Finally related to our interface, we need to implement a better consistency in the "key tap" gestures we used, since finger options to select and paint cause confusion in some users, as well as work on the spatial awareness of the interaction space of the used sensor.

Future Work and Discussion

We believe what is promising of this study case is the performance of the S3Doodle running on a browser handling live multiple hand positions and drawing (twice, since it is stereoscopic) on a web canvas. The current state of web technologies is powerful enough to compute values and "efficiently" run applications on the client. With this, for the application's future some enhancements to work on are in the shape of (i) adding size cues to the cursor to enhance the depth perception, (ii) introducing more advanced tools, (iii) adding features like video and image manipulation and (iv) fixing usability limitations noted above.

We have showcased the application in university fairs and in the IEEE 3DUI Symposium this year. We approach users, explain them the GUI and let them play around with it. While we haven't formally logged results due to the exploratory nature of the application, we have found many of users' comments important to get a preliminary list of UX limitations and ideas. Users have expressed that the task is captivating due to the depth that is applied to the doodle itself (rather than the depth applied to the GUI elements).

As stated before, this application serves as a test bed for stereo rendering in a web app. What interests us the most is the use of this type of format to get content to the user. We can create video players, games, doodling applications, sketching applications and infographics all of them being stereoscopic and interactive. Movies, pictures, videos, all can be apps on the web, taking advantage of the immersion that gestures and 3D provide, and enabling a better user experience in certain tasks.

Finally, a proposed experiment based on initial findings with this application regarding cursor implementations (aiding the position illusion with size) and perceived emotional perception is also recommended.

References

- Bailly, G., Walter, R., Müller, J., Ning, T., & Lecolinet, E. (2011). Comparing Free Hand Menu Techniques for Distant. *Human-Computer Interaction – INTERACT 2011 Lecture Notes in Computer Science Volume 6947*, (pp. 248-262).
- Broy, N., André, E., & Schmidt, A. (2012). Is stereoscopic 3D a better choice for information representation in the car? *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications - AutomotiveUI '12* (p. 93). New York: ACM Press.
- Broy, N., Schneegass, S., Alt, F., & Schmidt, A. (2014). FrameBox and MirrorBox. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (pp. 2037-2046). New York: ACM Press.
- González-Zúñiga, D., & Carrabina, J. (2015). Hacking HTML5 Canvas to Create a Stereo 3D Renderer. *Proceedings of the IEEE 10th Symposium on 3D User Interfaces 2015* (pp. 155-156). Arles: IEEE.
- Gonzalez-Zuniga, D., Chistyakov, A., & Carrabina, J. (2014). Re-defining a Pattern: Stereoscopic Web Perception. *Latin America Transactions, IEEE*, 12(3), 514-519.
- Häkkinä, J., Posti, M., Koskenranta, O., & Ventä-Olkkonen, L. (2013). Design and evaluation of mobile phonebook application with stereoscopic 3D user interface. *CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13*. ACM Press.
- Huhtala, J., Karukka, M., Salmimaa, M., & Häkkinä, J. (2011). Evaluating depth illusion as method of adding emphasis in autostereoscopic mobile displays. *3th International Conference on Human Computer Interaction with Mobile Devices and Services - MobileHCI '11* (p. 357). New York: ACM Press.
- International 3D & Advanced Imaging Society. (2015). *CES Las Vegas Report*. Las Vegas.
- Kulshreshth, A., Schild, J., & LaViola, J. (2012). Evaluating user performance in 3D stereo and motion enabled video games. *FDG '12 Proceedings of the International Conference on the Foundations of Digital Games* (pp. 33-40). ACM Press.
- LEAP Motion. (2015, 04 13). *Motion Controller for Games, Design and More*. Retrieved from LEAP Motion: www.leapmotion.com
- Mundy, P., & Newell, L. (2007). Attention, Joint Attention, and Social Cognition. *Current directions in psychological science*, 16(5), 269-274.
- Puell, M. (2006). *Optica Fisiologica: El sistema optico del ojo y la vision binocular*. Madrid: Universidad Complutense de Madrid.
- Schild, J., LaViola, J., & Masuch, M. (2014). Altering Gameplay Behavior using Stereoscopic 3D Vision-Based Video Game Design . *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (pp. 207-216). New York: ACM Press.