# History and map of best-selling video games
## - Project report

**Information Visualization -** DH2321, ivis12.

**2012-03-23**

*Rakiv Ahmed*
*William Lundin Forssén*

# Table of Contents

# 1. General introduction and visualization problem.

The project aims to visualize the difference in video-game sales and the popularity of different gaming consoles in a specific time span. To do this we will create a world map where continents represent gaming consoles and countries represent games. The land area of a country would represent the sales figures a game. For instance, the console *Nintendo Wii* would become a rather large continent compared to the *Sega Dreamcast* because of its popularity and the game *Wii Sports* would become a country on the Wii continent of significant size because the game sold so well.

To do this in a specific time span we would use a playback function where continents and countries pop up in regards to their release year (like Gapminder's playback).

## Why is this interesting?

We find that representing data as a world map is something that most people can relate to (since most people have seen a world map). A world map makes is possible to easily visualize the differences in the size of continents and countries while at the same time separating data which has different categories.

Most importantly, it also gives us room for creativity concerning the visual aspects of the world, continents and countries. The opportunity to work on an large scale image for a 4K screen comes rarely and we had to utilize this chance.

# 2. Background and related work

The web comic *XKCD* has made two world maps[1] [2]of social networks so far. This website bases its maps mainly from sources such as Google, Bing, Wikipedia, Alexa, Big-Board.com, StumbleUpon and several other statistical websites but also does a statistical estimate (guess work). These two maps have become our main inspiration for this project. Other inspirations are different billion dollar grams; such viewed on "Information is beautiful"[3] and talked about by David McCandless.

The origin of the billion dollar charts comes from the treemaps[4] done by Ben Shneiderman in 1990 and newsmaps[5] done by Marcos Weskamp in 2004. This is a clear way of presenting data in 2D format concerning different subject and their sizes. But our stance is that it would not be as aesthetic or explorative as a world map, even if some of the informational aspect might be lost. Our way to combat this is to use the preattentive property of colors[6] and the gestalt principle of proximity[7] as a visual mapping[8] to indicate different consoles and their respective games. Sizes are of course important in both tree and news maps and our world map, the difference being that we use irregular shapes instead of rectangles.

---

*1 XKCD: Online Communities 1 http://xkcd.com/256/ [Accessed 2012-02]*

*2 XKCD: Online Communities 2 http://xkcd.com/802/ [Accessed 2012-02]*

*3 The Billion Dollar Gram http://www.informationisbeautiful.net/2009/the-billion-dollar-gram/ [Accessed 2012-02]*

*4 Treemap. Mazza, Riccardo. "Introduction to Information Visualization". .2009, Springer. (p.83-85),*

*5 Newsmap. Mazza. (p.85-87).*

*6 Color. Mazza (p.35).*

*7 Proximity. Mazza (p.42).*

*8 Visual mapping. Mazza (p.20-23).*

World maps have been done previously but in much smaller cases or without any relationship to data. Maps for navigational use is a given, whether it is for the real world or some fantasy one. One example is the making of website maps as done by Andrea Rocci[9]. The map consisted of islands representing different topics in an online course. Similar to our map, there were some clickable content as well.

We realise that too much information at once can be overwhelming. So with the concept of *Information Overload* [10] in mind, we implemented the timeline to start at 1976 when there were no popular consoles in the world (the world is a blank empty space of water in this infant-like state). The user is allowed to generate the world at a pace they see fit so that they don't miss any information or can choose to ignore information they deem irrelevant (by allowing the user to fast-forward to a year of their own interest).

We also chose to limit the amount of games on the consoles to only the top 10 sellers in order to only visualize the most relevant aspects of the world, had we not done so the amount of countries would have been overwhelming with some consoles having well over a 1000 games.

It would sometimes also be relevant for users to see detailed information about each console and their games (so they can further relate to the data that is reflected through the size of the continents and countries). Since this information is not relevant to everyone it's hidden until requested (by clicking on a continents name to receive a window with this information).

# 3. How have you collected the data you need.

We processed the data from Wikipedia[11] and put them into a Google Spreadsheet. We used Mr.Data Converter to convert it to the JSON JavaScript format, similar to what we did in Lab 1 with Daniel Lapidus from Gapminder. This took us approximately 2-3 hours per group member (total 4-6 hours). The names of game and console had to be cleaned by removing special characters (non-alphanumerical) so they could be used in the file names for images. Spaces were replaced with "-" for example. "_" were used as a separator for special cases, such as continents and title text.

# 4. What kind of tools, libraries and frameworks have you used to set up your prototype?

HTML5, CSS3 and JavaScript were used to produce the website. In JavaScript we used jQuery (mainly for the fading effects) and jQueryUI (for the information boxes). The reasoning behind our choice of libraries is related to the fact that jQuery allows us to create advanced visual effects while being easy to use and also that one group member had used the library before. jQueryUI was used for the information boxes simply because it allows HTML code to be embedded into the information messages while enhancing the visual look as the JavaScript function alert() does not allow HTML code inside the message nor does it allow for any form of visual animation.

---

*9 Website maps. Mazza (fig 6.3 p.94)*
*10 The Problem of Information Overload. Mazza (p. 105-106).*

*11 Wikipedia: List of best-selling video games.*
*http://en.wikipedia.org/wiki/List_of_best-selling_video_games [Accessed 2012-02].*

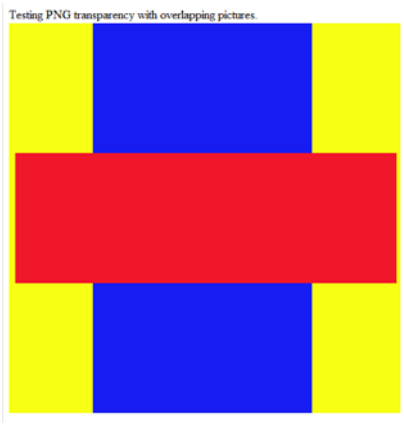
The images were created with Photoshop, and since there were approximately 240 different images, Photoshop scripting was incorporated into this process to batch a large amount of images at once (minimizing the amount of work hours). Photoshop Scripting is a powerful tool and is based on JavaScript, ActionScript or Adobe's own adaption. Without the script we found to easily export grouped layers and trim them down to a smaller size, this project would not have been feasible due to performance issues (mentioned later). This script enabled us to make changes to our images and quickly have them in the appropriate format and size.

# 5. Description of prototype

## Web programing work-flow

In total we made 15 different version of our prototype, each subsequent one incorporating more functions and/or visual effects than the previous. The following steps will indicate the work flow for the prototype:

**1.** This version tested the starting-assumption we had that PNG images with partial transparency would be able to overlap each other without effectively "covering" *entire* areas of other images that were opaque (only partially). This assumption proved to be correct.

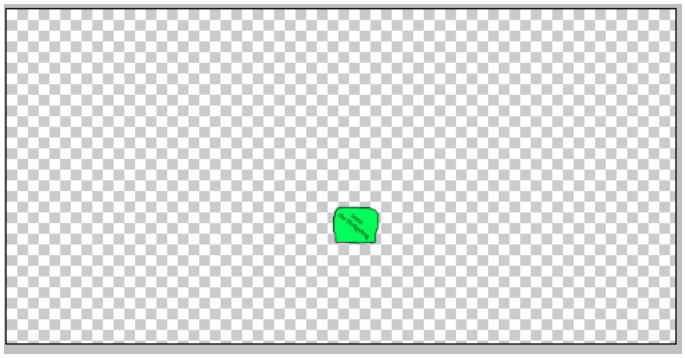



All images have the same size and position.

**2.** We started testing the HTML5 Canvas in this version. It could display the images just fine. But redrawing became a problem since you couldn't simply change the source of the image to remove it, but you needed to redraw the entire Canvas which was bad for us performance wise. We had also added a "year-slider" by using the HTML element <input type="range"> (Google Chrome is currently the only browser that supports this).


Conceptual continent with time slider.

**3.** The testing of large images (4000x2000 pixels) began. With just 3 or 4 images visible the performance decreased severely (even though 95% of the pixels in the images were transparent). This would not work for our purposes as we had about 240 images in total.



The grey and white squares are transparent surfaces.

**4.** HTML5 Canvas was replaced by the more commonly used HTML <img> tag. This boosted our performance quite a bit due to the fact that we didn't need to redraw the entire canvas when an image changes state. A header, footer and a floating menu was added to the site, but the CSS for the text and background was not included in this version.


The slider is above (z-index wise) all other elements.

**5.** Implementation of the jQuery library. Fade-in and fade-out functions added. CSS for the header, footer, background and surrounding areas added. The gradients were made using only CSS (Chrome webkit).

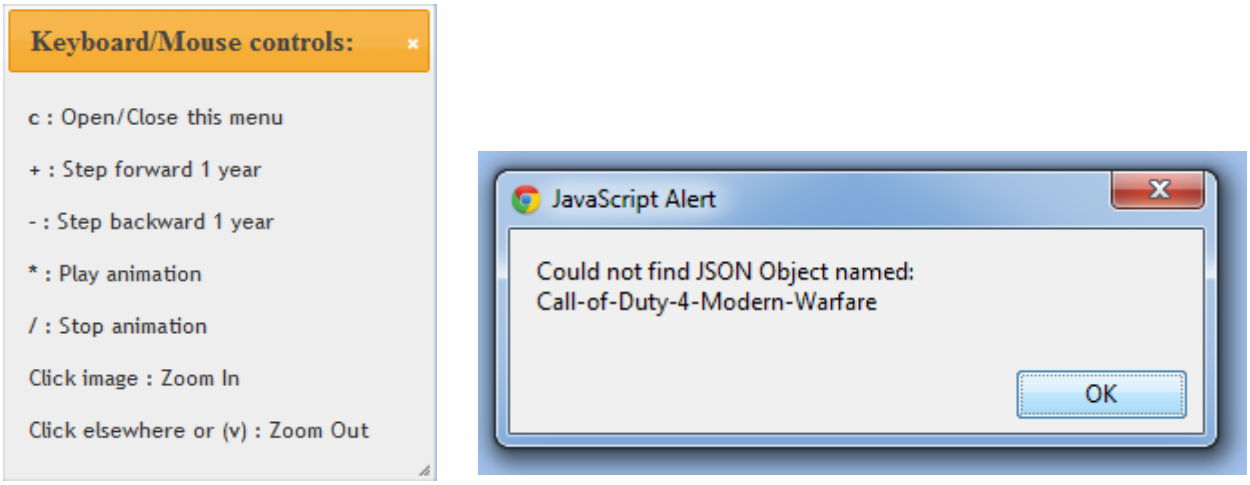

New improved header fixed with CSS.

**6.** No visible differences implemented but several bugs were taken care of. The reading of JSON data was implemented. A function for giving images a unique jQuery safe id was implemented (jQuery is very sensitive about certain characters for id tags and does *not* escape ( \ ) them automatically).

**7.** Implemented a system for detecting if images were loaded correctly or not, giving the user alerts if anything went wrong, most of these problems appeared when a file name wasn't matching the JSON data. jQueryUI was also implemented for the information messages.

Another library called Zoomooz was implemented as a way of controlling the zooming functions of the website. Zoomooz worked well for zooming in as the library allows for zooming on any object on

the site, however zooming *out* became an issue and the entire idea was later abandoned when it was realized that browser zooming would be sufficient.




[Left] jQueryUI Message Dialog.
[Right] Image error checking function in action.

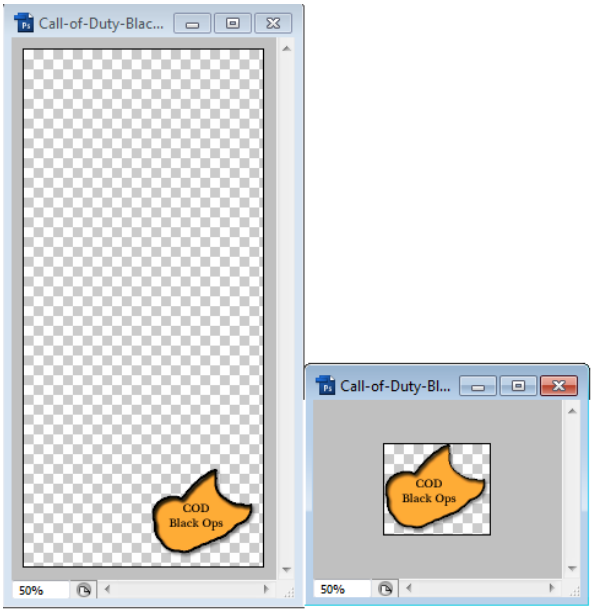
**8.** At this time we believed that the size of a country image could be the same size as the continent of that country. With this belief in mind we only had to implement X and Y positions for the continents and then match the countries of that continent to the same coordinates (later as the number of images grew, performance became an issue once again). Also a new look for the year-slider was implemented.


The new and improved year slider.

**9.** In this version a search for a color-overlay library was initiated. However, all of the libraries and functions found did not care for transparency in the images thus coloring not just the opaque areas but the transparent ones as well. If we had used HTML5 Canvas this would not be an issue as the canvas can manipulate on a per-pixel basis. A decision was made to have correct coloring from the beginning (in Photoshop) and not to manipulate it by overlays.

**10.** A click-image function was implemented and added to all images. The function displays a jQueryUI dialog with detailed continent information (sales, release year etc.). Overlapping was not a problem as each image was as big as the continent it belonged to. But as we later switched to smaller images the small overlapping between different images belonging to different continents made this function obsolete.

**11.** Instead of images having the same size as their respective continents, we trimmed them down to be as small as possible. This increased performance tremendously. We also learned that Google Chrome supports hardware graphic acceleration (Chrome flags) which also increased performance.

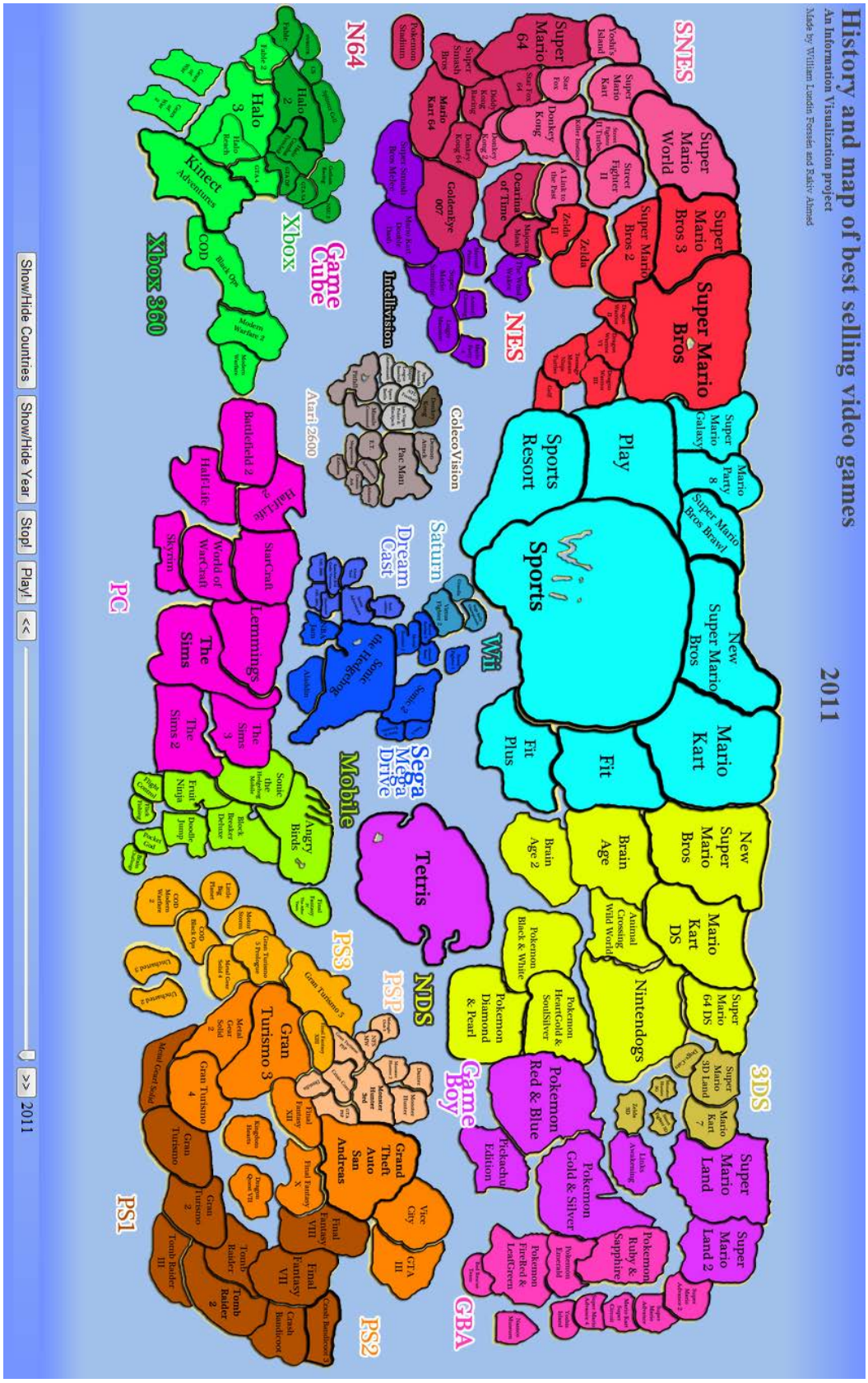


[Left] Before trimming. [Right] After trimming.

**12.** We added some text next to continents to indicate which console it was representing. The click-function mentioned in version 10 above also came into use once again as we put the texts z-index to be above everything else, making image overlapping a non-issue.

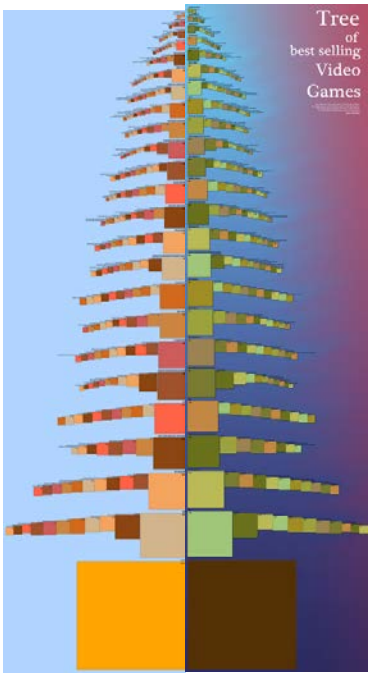
A click on a continent text brings up a jQueryUI dialog showing detailed information.

**13-15.** Added a grand 10% opacity background year indicator to aid viewers/users as to what year was displaying, a button to turn this feature on and off was also added. A button for only showing the continents (without any countries) were also added. Other changes in these versions were related to fixing small bugs and general tweaking.




Final version of the website with the slider set at the year 2011.

## Drawing the areas with JavaScript




[Left] JavaScript produced image. [Right] Photoshop processed image.
These images could be viewed as a tree.

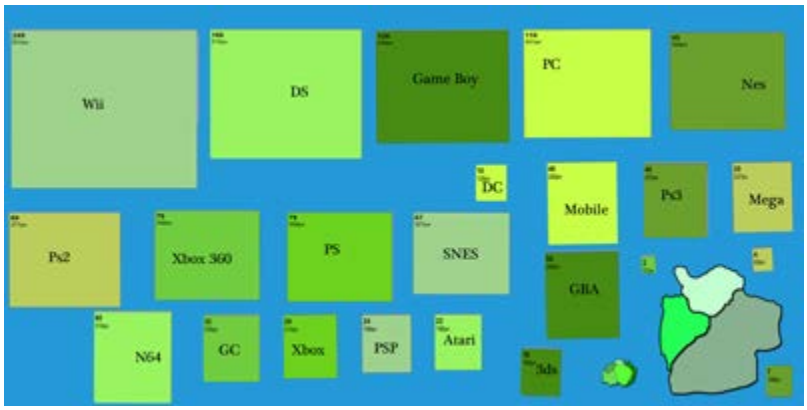
We drew a simple image in JavaScript (left image) using the data gathered previously and cleaned after getting processed by Mr. Data Converter. We also processed the image in Photoshop (right image) to get a nicer visual finish. The results can be viewed on http://www.nada.kth.se/~rakiv/ for greater details.

The JavaScript simply read the data, got the relevant information (name, console and sales) and drew them up in a special pattern. Small continents got draw first at the top while larger ones getting added further down, with their respective games being added horizontally to form a branch. Lastly a square representing all the sales we had got drawn.

The images and areas later helped us sizing up the world map. We could have developed this script further, making other shapes or even a map, but it would not be as cohesive or creative as doing it manually. It would have been a real algorithmic mapping challenge, which we felt we could not solve in the time frame of this project.

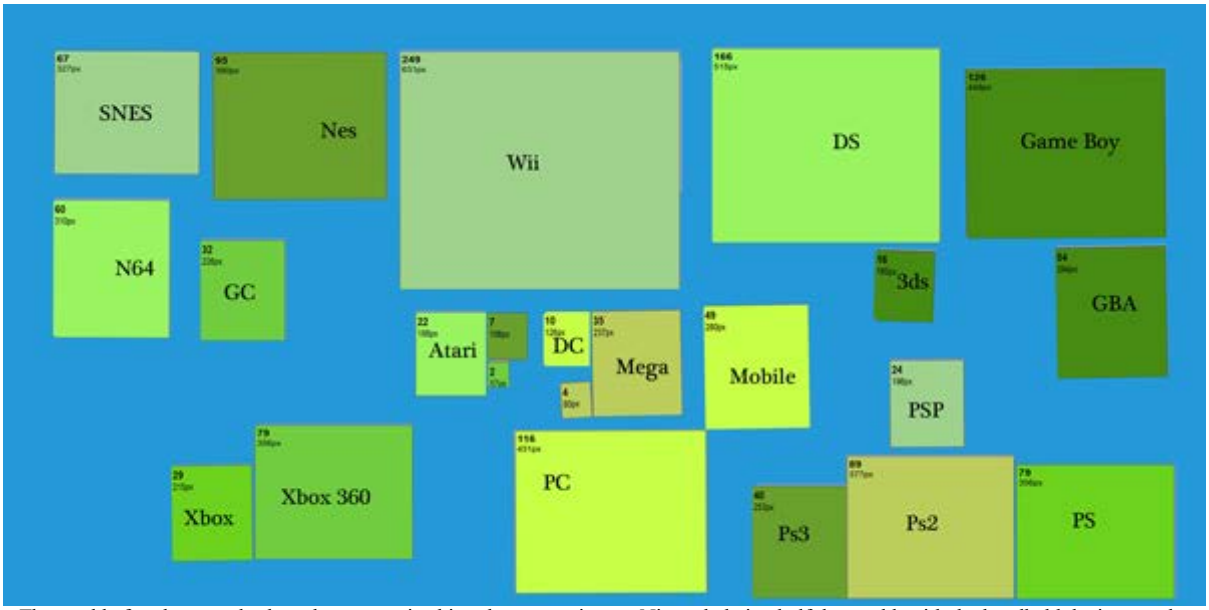
## Photoshop work-flow

First of all we imported the tree image into a 4000 by 2000 pixel image and started to separate the console rectangles to get a general over view of the world. The rectangles got resized to fit the new image size better.



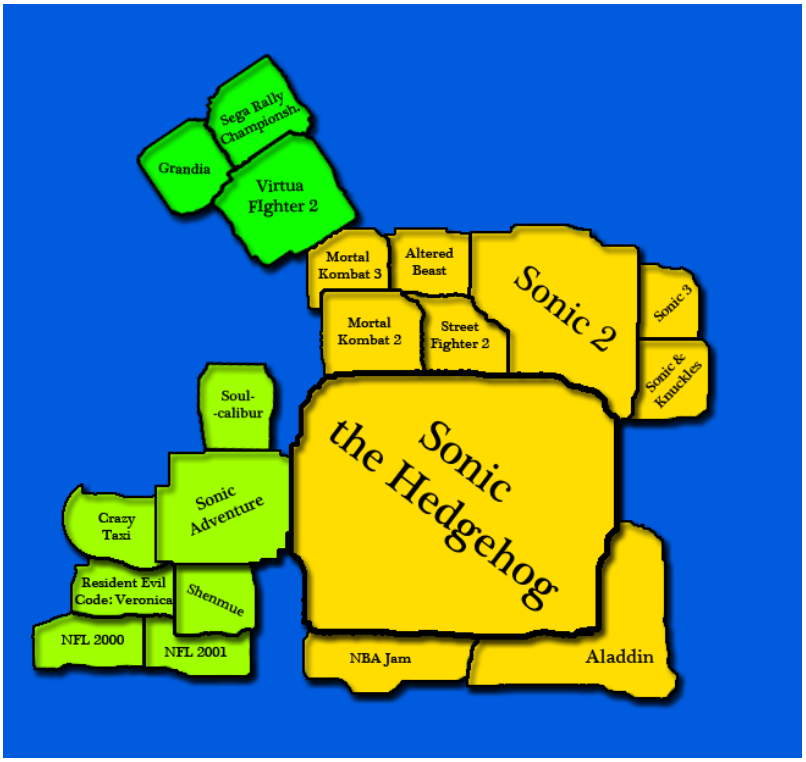

The very first draft of the world using scaled rectangles. An example continent is shown on the lower right corner.

After that we started to group the console by their company or technology relevance to get a bigger continent. For example we have a Playstation continent and the earliest consoles (Atari 2600, Colecovision and Intellivision) making their own continent. PC and mobile games also joined together, as they are an open gaming platform not tied to a specific company.




The world after the consoles have been organized into larger continents. Nintendo being half the world, with the handheld devices on the right.

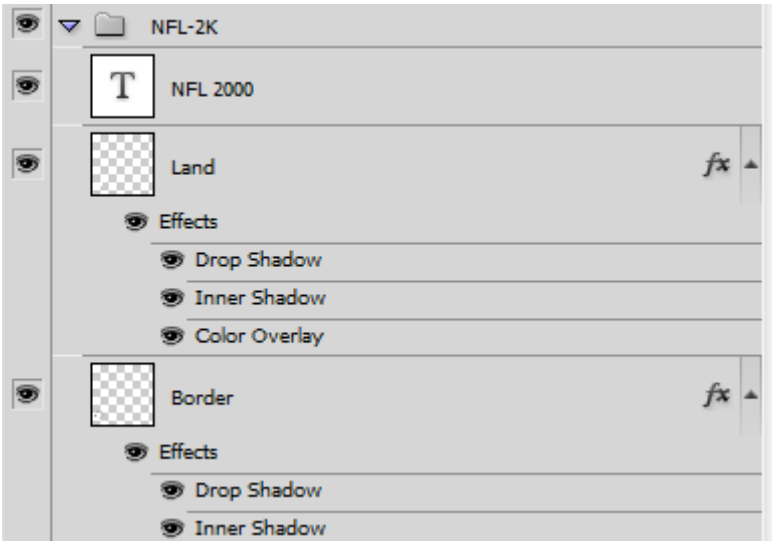
The first continent to be drawn was the Sega one, since it was in the center, not too big and quite easy too map without too many game series. One idea was to try to keep games of the same series or genre close together.



The first continent drawn. This was later redrawn to fit the better map style.

The result was not what we hoped for with the countries being to rectangular but we had a preliminary work flow in place now. The work flow continued to develop during the project. We will only describe how the work flow in the end was like. It took us many hours to develop the appropriate work flow which would work with Photoshop scripting.

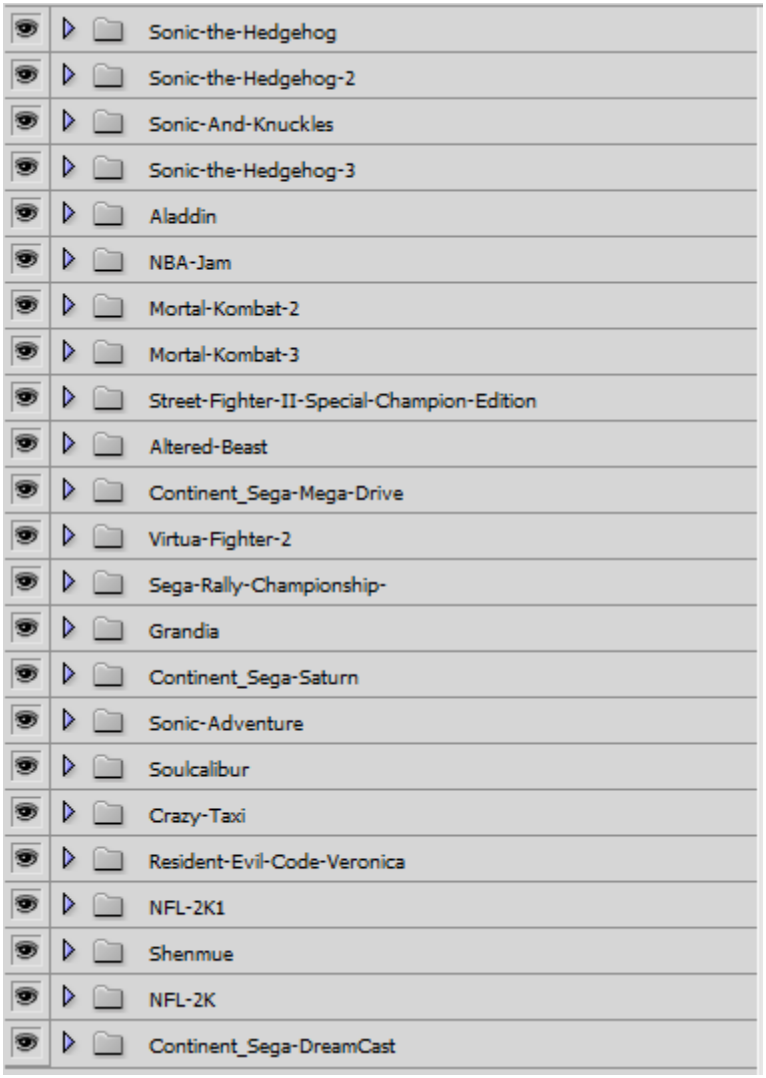
The final work flow consisted of using 1 group and 3 layers for each country. The group name being the game's name, as it was required by the Photoshop script to easily export all grouped layers. The first layer made for a country consisted of the actual land area which was drawn by hand. The next layer for the border, which was made by selecting the land area and then stroking (Photoshop function) it with black color. These two layers got some shadow effects added to them. Lastly the text layer was added with the game's name.



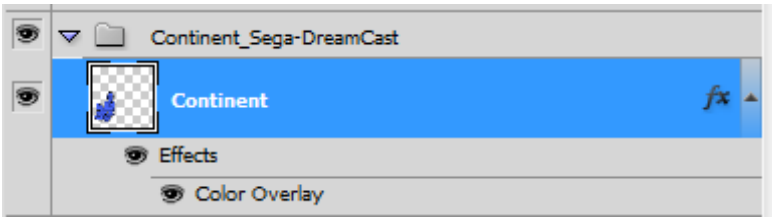
The 3 layers creating an country game group, with the responding group name.

Each console and continent also got colored according to their logo or color of the actual console (where applicable). 4 to 6 hours were spent on drawing each continent. About 1 hour was spent thinking on the mapping of the countries and how to place them optimally for each continent. The Nintendo part of the world took a while longer since there were so many games, especially the Super Mario games. The solution was to let all the Mario games touch one after another, creating a long chain of Mario games at the top of the world. Big series such as Zelda and Pokémon got their own islands.

To create the console backgrounds all the game groups belonging to a continent were ordered together, copied, merged and got a color overlay effect to get the desired color. Once a continent was done, we simply used the Photoshop script we found to easily export and trim all the grouped layers.




Ordering all the games above their respective console.




The result of merging the countries together.

# 6. Evaluation of prototype

We have not been able to get any real user feedback since our prototype is very specialized for a 4K screen. The user feedbacks we have gotten have been done by letting people check out the map on a regular screen. Users we have been testing on are gaming interested and are aware of some the titles and knows which ones are usually big sellers. This is our target audience, since people not interested in gaming would probably not understand or appreciate the map.

The users often gets the basic idea of the map quickly and might ask the question “Does the country size represent sales?” to get confirmation. They also understand that the games come in chronological order when the slide show starts playing. Although it can be overwhelming on the later years, the slide show functions were appreciated. Otherwise they used manual navigation.

Since we had two trial demonstrations in the VIC room, we had the opportunity to get expert feedback from Björn Thuresson, who is both interested in information visualization and gaming. He though it was good that we chose to make a map instead of a regular billion dollar chart or similar, as those tend to be boring. He had some issues with the Wii Sports country having Wii spelled out in a inland lake form, as Wii and Wii Sports already dominated the map and viewer’s attention. We decided to shrink this lake after that. He also made other graphical remarks, which for the most part we adjusted to.

# 7. Discussion

## Problems and important findings

Large images (width, height (amount of pixels)) were problematic in relation to performance. This lead to the implementation of a per-image position implementation. In practice this meant that each image was given an X and a Y position which was implemented as CSS margin-top and margin-left.

## Interesting points

The continent sizes are not a representation of the console sales, as one might think, but is a sum of all the sales of those continents top games. If we had made the continents to match their own sales it would have generated great areas of desert landscape on each of the continents as we only had time to study the top 10 games of each console. This would probably have added further confusion about the map itself rather than aid in the visualization that we were trying to accomplish.

## Future work

Implement support for different browsers and screen resolutions. It was optimized for Google Chrome and the Visualization Studio (VIC) screen of a 4096 by 2400 pixel resolution.

Implement a function so that countries start small and then expand to their correct size as time passes (since sales increase over time and doesn't really come in big chunks). This can be difficult as the data for the number of sales per year is difficult and time consuming to find, but could be achieved by either linear growth or other fitting function.

Learn more about Photoshop scripts so we might be able to do the positioning via scripts instead of manually. This would save us many hours, especially when content is added or adjusted spatially.

Photoshop scripts might also help us map the individual countries and make them clickable as well, as it would require a specific image or rectangle area for the clickable surface. At the moment we have rectangular console title text areas that are clickable.

## Conclusion

We felt that this information visualization was an out-of-the-box idea with both a creative and a technological challenge. It was very interesting to learn about the best-selling games throughout history since we are both interested in gaming. A future note is to learn more about the scripting and have a more scientific and computational approach.

# Link to prototype:

www.nada.kth.se/~rakiv/best_selling_video_games.html
Click on the "History and map of best-selling video games" link.
Follow the instructions to be able to view the interactive slide show.
You can view a static image as well.