

Kandinsky and Pollock in 3D

Lucas Simes
Math 198 - Francis
Project Proposal
Written in Glorious L^AT_EX

December 13, 2016

Abstract

The development of the abstract style in art has allowed for the integration of geometry and precision into an age old form; the intention of this project is to recreate the 2 dimensional style of certain celebrated abstract artists in a 3D space. The final iteration is presented to the viewer in such a way that they have full control over the re-generation of the painting, as well as full control over the 3D rotation of the scene projected onto a 2D canvas. In order to achieve this, I have spent quite a bit of time finding art which would prove interesting to recreate within the canvas space - the artists I have chosen, Wassily Kandinsky and Jackson Pollock, have rendered styles which can be recreated through the development of specific shapes, lines, and materials that can be randomized within a set of limits. The entire program is written in Javascript utilizing the THREE.js library. The final project features smooth camera controls, and a design intuitive even to users who are not familiar with my process.

1 Introduction and Background

Throughout human history, art has been used to a great degree in the pursuit of the recreation of real images; future generations would have to rely on the descriptions of a scene or a king, for example, if an artist was not consulted to create an accurate landscape or portrait. Some artists, like Johannes Vermeer, developed techniques within the market and field of painting that proved integral in the field of hyper-accurate representation of a scene or object, but it was not until the development of photography in the mid 19th century that art was truly forced between a rock and a hard place; artists had to make the decision to compete in a services market against a growing technology, or to develop a new and unseen method of art. The direct result of this difficult decision is the creation of the concept of an *art movement*. From this concept came many incredibly important movements, which can be recognized in world famous expressionist, pointillist, cubist, and modernist artists. This particular project will be used as a homage and tool to facilitate the appreciation of **abstraction**.

The early 20th century Russian artist known as *Wassily Kandinsky* is considered by many to be the grandfather of what we now know as abstract art,

and is most well known for his role in the maturation and development of geometric abstraction. His work significantly features purely abstract images - he transitioned from painting almost entirely impressionist works to focusing on turning images that highlight geometric patterns and shapes, arranged with the aid of a particular color palette, into full abstract compositions. His work was not entirely accepted within its early days - in the 1930s, Nazis took his first compositions from the Bauhaus School of Art and displayed them in a state art exhibit titled "Degenerate Art", before ultimately destroying them. Kandinsky believed that painting was similar to composing music, and that ultimately, abstract art is a physical representation of the way that music makes listener feel. Many of his compositions can be found on display at the Art Institute of Chicago.

The later 20th century American artist known as *Jackson Pollock* was a significant member of the art movement of abstract impressionism. He was known most famously for his unique style of art, drip painting. Pollock composed his paintings in his own style as well, which he attributed to his appreciation of Native American art; he found it important to work on and appreciate his artwork from many sides at once, and often worked by walking around all four sides of his prone canvas. From the early 1940s to the early 1950s, Pollock's work was featured in museums nationally, until his untimely death in 1956 as a result of drunk driving. Nearly a dozen of his compositions are also on display or kept in archive at the Art Institute of Chicago.

The intention of this project is to recreate the styles of these two celebrated abstract artists in a medium that was unavailable to them at the time: a 3 dimensional canvas space. As all but the works and legacy of these two artists remain, the project will revolve around the random generation of geometric objects that can simulate the style of their paintings. This will be achieved in **Javascript** utilizing the library offered by **THREE.js**. The project will be generated in three dimensions, but presented for viewing in two dimensions.

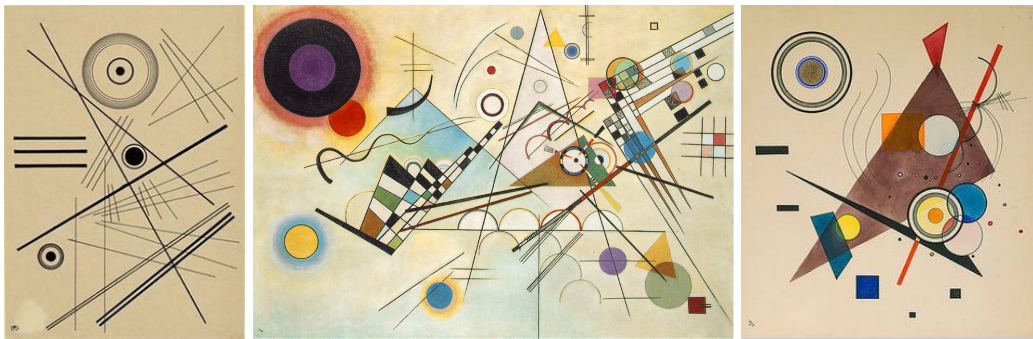
2 Method and Process

I began the process of researching both Kandinsky and Pollock in the early fall, once my ideas were solidified. My most important concern was in choosing work from a section of their career that I believed best described

their visions, while also ensuring that the specific works I had chosen would prove itself possible for timely semester-long completion. After a great deal of narrowing, I decided to focus my attention on recreating their styles through studying the following images:

2.1 Kandinsky Pt 1

For Kandinsky, I chose 3 exemplary works.



These works are all three a part of Kandinsky's *Compositions*. Works in this series begin resembling a geometric-abstract style that I believe is typical of Kandinsky, and provides an excellent base in the pursuit of the recreation of his work. This series is, by and large, composed of simple geometry, manifested in seemingly random locations throughout each respective canvas.

In order to complete a Kandinsky in THREE.js, I first made a list of a number of common objects and qualities present in his works; most compositions are composed of a great deal of straight lines with no particular relation to one another in length, width, or position. After lines, Kandinsky included simple, straightedge geometric shapes (*squares, triangles, planes, etc.*) and curved shapes (*circles, Beziars, etc.*). I quickly developed a rudimentary random-value system in my program to represent points within the cube, corresponding to a triangle, a plane, and a few lines. I created 3 separate variables for every single point needed; the result was inefficient and gross, but worked at the time:

```

var o = Math.floor(Math.random() * 10);
var oo = Math.floor(Math.random() * 5);
//triangle variables:
var x = Math.floor(Math.random() * 20)-10;
var y = Math.floor(Math.random() * 20)-10;
var z = Math.floor(Math.random() * 20)-10;
var a = Math.floor(Math.random() * 20)-10;
var b = Math.floor(Math.random() * 20)-10;
var c = Math.floor(Math.random() * 20)-10;
var l = Math.floor(Math.random() * 20)-10;
var m = Math.floor(Math.random() * 20)-10;
var n = Math.floor(Math.random() * 20)-10;
//plane variables:
var xx = Math.floor(Math.random() * 5)-oo;
var yy = Math.floor(Math.random() * 5)-oo;
var zz = Math.floor(Math.random() * 5)-oo;
var aa = xx + Math.floor(Math.random() * 5)-oo;
var bb = yy + Math.floor(Math.random() * 5)-oo;
var cc = zz + Math.floor(Math.random() * 5)-oo;
var dd = xx + Math.floor(Math.random() * 5)-oo;
var ee = yy + Math.floor(Math.random() * 5)-oo;
var ff = zz + Math.floor(Math.random() * 5)-oo;
var gg = xx - aa;
var hh = yy - bb;
var ii = zz - cc;
var jj = xx - dd;
var kk = yy - ee;
var ll = zz - ff;

```

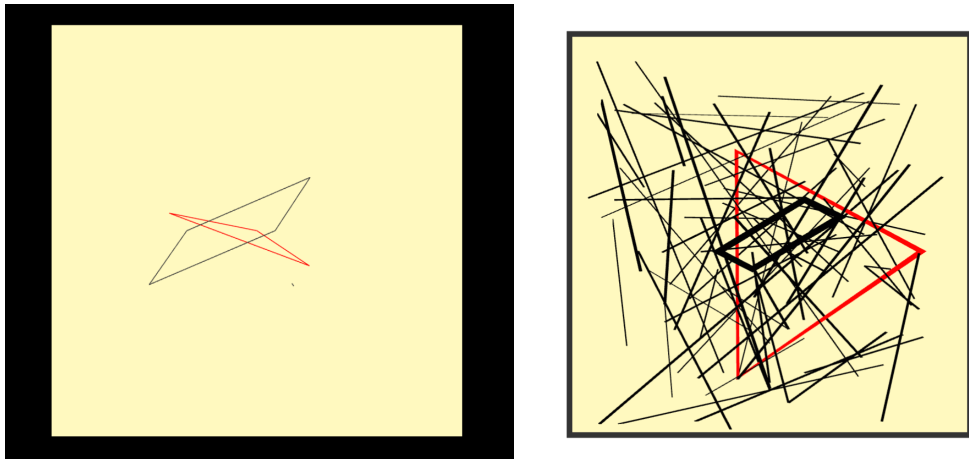
Figure 1: A gross an inefficient temporary solution.

Satisfied with my work, I moved on.

2.2 Camera and UI

I spent a great deal of time perfecting this - it was my opinion that this was one of the most important aspects of the program. My rationale was that every user of this program should not have to worry about proper usage and technique - instead, their power should be focused on the generation of aesthetically pleasing images. In order to complete this task, I first implemented a system to rotate all objects in the scene in the direction of each arrow key - this proved useful in determining my next step, which was improving performance and image. I quickly scrapped, mentor Nathan's request, my code which commanded objects to rotate, in favor of a system which allowed the camera to orbit around the objects of the scene - a much more intuitive solution in the event that I need to add or remove an object from the scene. Next, I altered the color of the scene, inserted a frame around the canvas, and added an anti-aliasing effect; these choices proved integral in creating a proper environment to mimic a gallery canvas. I achieved the cel-shading appearance of the frame by inserting a larger cube around the existing cube, and removing rendering on its camera-facing sides.

Finally, and most importantly, I discovered the real library I needed to ensure that the program worked across all operating systems and web browsers: THREE.MeshLine. This library allows lines created through THREE.js to carry a weight larger than 1 - a change that I had vowed to make (and falsely diagnosed) in my seminar. A diagram below visualizes the changes:



I moved on after the completion of this UI to the beginning of starting my Pollock program.

2.3 Pollock

Pollock's works are similar throughout his artistry career, so I did not choose a specific set of works to base my program off of. Instead, I decided to create a program reminiscent of his splatter painting technique in general.

For this program, I utilized the Bezier curve option available within THREE.js. This program was much simpler to create than I had previously anticipated; I had started out along the same inefficient path as I had began in Kandinsky, in which I write a brand new series of lines and variables unique to every object. On mentor analysis, I began to realize that there was another entire strategy I had missed while writing the initial code:

if I run a *for loop*, I can generate all of the lines I need without creating unique variables for every line rendered.

I quickly implemented the changes to my program, which allowed it to render hundreds of lines in any shape or color I desired - for Pollock, I devised a system in which the first and last point of each Bezier curve is constrained to one side of the cube, while the remaining points are free to choose any part of the canvas' space to distort the curve. I applied this idea to each side of the cube, and the result looked very much like a Pollock.

2.4 Kandinsky Pt 2

I adapted the efficient changes made in Pollock within Kandinsky, and began to work on adding a greater library of objects to the canvas. After its addition, I considered both Pollock and Kandinsky to be complete.

3 Inspiration and Concluding Thoughts

After the completion of both programs, I came to a decision to understand what I could include that I had not promised before in my previous documentation. In order to understand what I could be missing from each project, I decided to paint my own cubes in each style.

I attempted to achieve an effect similar to that found in my program on both cubes. For Kandinsky, I painted geometry common to his compositions - for Pollock, I spread all three sides onto my floor and splattered paint as though I was Jackson himself. What I had found for Kandinsky is that there is a certain amount of feeling, captured in Kandinsky as well as other abstractionists like Miro, in un-opaque lines. As a result, I worked to remove opacity from the lines of my Kandinsky program.

In my study on Pollock, I found that one aspect I had entirely glossed over was his use of paint splotches - I decided to add, in response, splotches

which fit into the program's decided aesthetic. I constructed one full cube of two halves to aid me in this process.

4 Concluding Thoughts

I believe that I have successfully reached the criteria I have set out to achieve over the course of the semester. My regret is that I had only one semester to fully develop the ideas I had presented - if I had more time, I would have loved to replicate Miro or Mondrian in 3D. I believe that MATH 198 was an excellent method of launching me into the world of coding, as an entire outsider. In addition, I am very grateful to the expertise of the mentor team - I could not have completed this project without Nathan's bountiful advice.

I am very proud of my first completed programs in Javascript, and I hope that I keep the motivation not to drop my practice and proficiency.