DG290: Creative Programming – Reflection

In the first half of this assignment I was introduced to programming. Before this course I had only written simple lines of code for my website, which consists of HTML, CSS and a very limited JavaScript code for a popup, with very little understanding of the actual language. I had tried to get into coding, but I had some difficulties actually following through, because a lot of the coding seemed uninteresting, and it was hard to keep track of my progress.

This assignment was a very hands-on course. Whenever we learnt something new, we instantly had to use it in a sketch. Processing is the perfect tool for this, since it is very oriented towards visual output, instead of simple conditional actions which you would see in 'regular' coding languages. By learning the basic syntax I learned to solve the weekly assignments we got, such as making circles revolve around each other and having a cube bump in a fixed window. I altered some of these project myself to create different things, such as the projectile motion sketch[3]. Doing this has helped me think about the use of things, they don't necessarily need to be used for their original goal, placing things outside of their usual context can lead to fun discoveries.

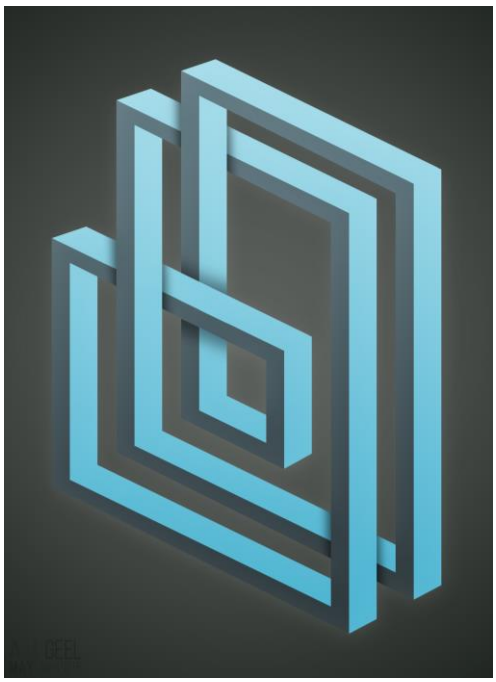

Image 1: the poster for challenge one.

For the first challenge of this assignment we were instructed to make a poster featuring static visual arts that demonstrates beauty and complexity, made in Processing. For this poster I wanted to make something that looks good aesthetically, and something that has been thought out. I did not want to make it have a randomised feel, so before I started coding I carefully thought about the design, and realised I wanted something simple, yet complex at the same time. I wanted to achieve this by studying art such as M. C. Escher's, and found that artists use impossible perspective[1].

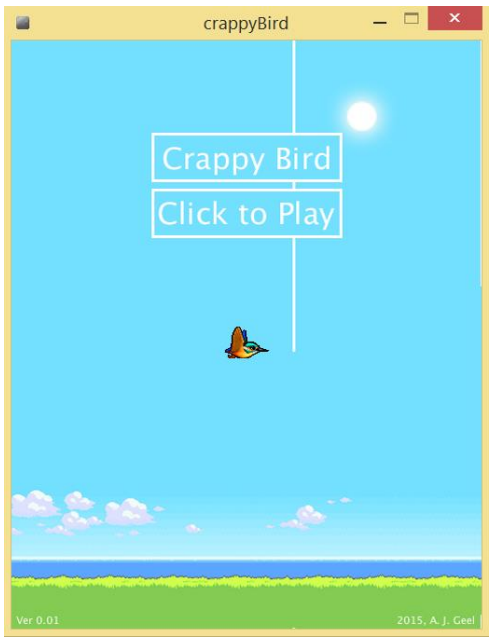

Image 2: the game for challenge two.

The project is built up from smaller vertices, all placed in coordinates on the map. I sketched and modelled the object prior to coding it, so I had to translate all of the coordinates to processing. While doing this I quickly learnt that in programming you need to be as precise and systematic as possible. The smallest mistakes in your code can break the entire script, and finding these mistakes can be a very time-consuming job.

The second challenge revolved around creating an interactive application. When I heard about this assignment I instantly thought about creating my own game. The thing with games is that they can be very diverse, they can be used for different applications and you can really follow an art style in them. For my game[2] I chose to pursue a retro-inspired graphical style. The game is inspired by the incredibly famous Flappy Bird game, which I tried to mimic to have a clear goal of what I actually want.

Even though the game is very easy to play and seems quite simple, it was hard to figure out how it should be written in code. I chose Flappy Bird, because it seemed quite easy to make, but in the end some things were quite complicated. The bird is stationary in the x-axis, and can only move in the y-axis. The bird accelerates towards the ground with a force of 1/3 pixels/frame^2, it can be boosted upwards (by performing a flapping motion) which adds some upward speed to the ySpeed.

The hardest thing in this game was making it detect whether the player was still allowed to play. For this I wrote two conditions: If the bird reached the absolute bottom of the screen it would be over, and if the bird hit the gates the game is also over. The way I solved this problem is by checking the x-position and y-position of both the bird and the gate in every frame, and if they overlapped the game would be over.

Even though my game may be deemed as simple by some, I still am proud of it. I solved a problem using pure logic, and I actually had fun while doing it. Learning the basics of code has opened a new world for me, since code can be used for a wide variety of things. In the future I would like to apply this knowledge in both new sketches/games as well as in real things, such as Arduino's.

A. J. Geel - s149144

Appendix

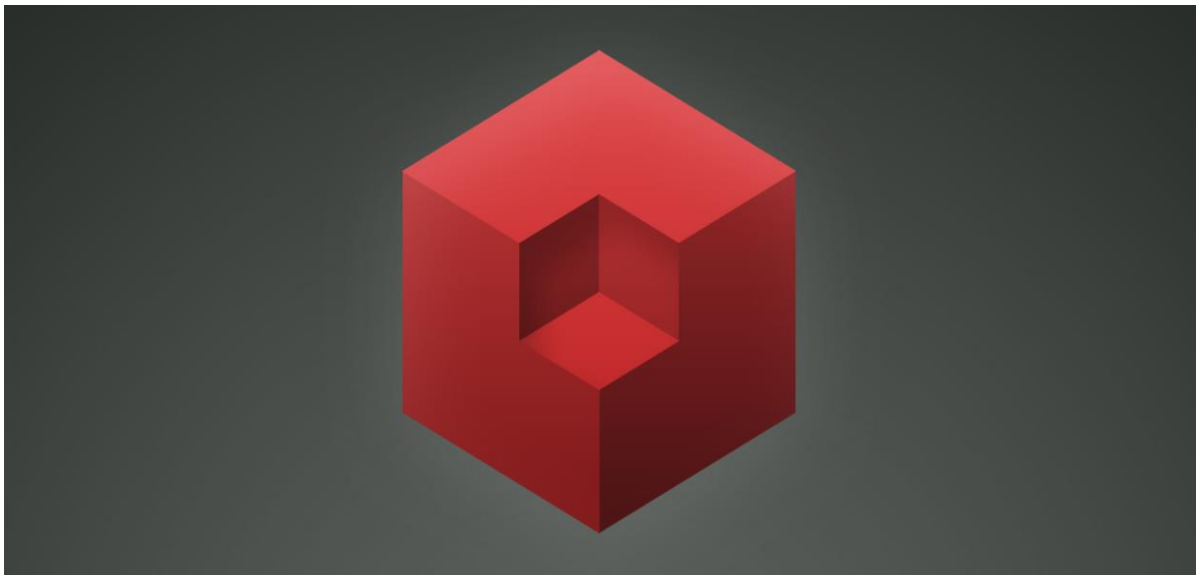

Image 3: various iterations of the ‘projectile motion’ script.

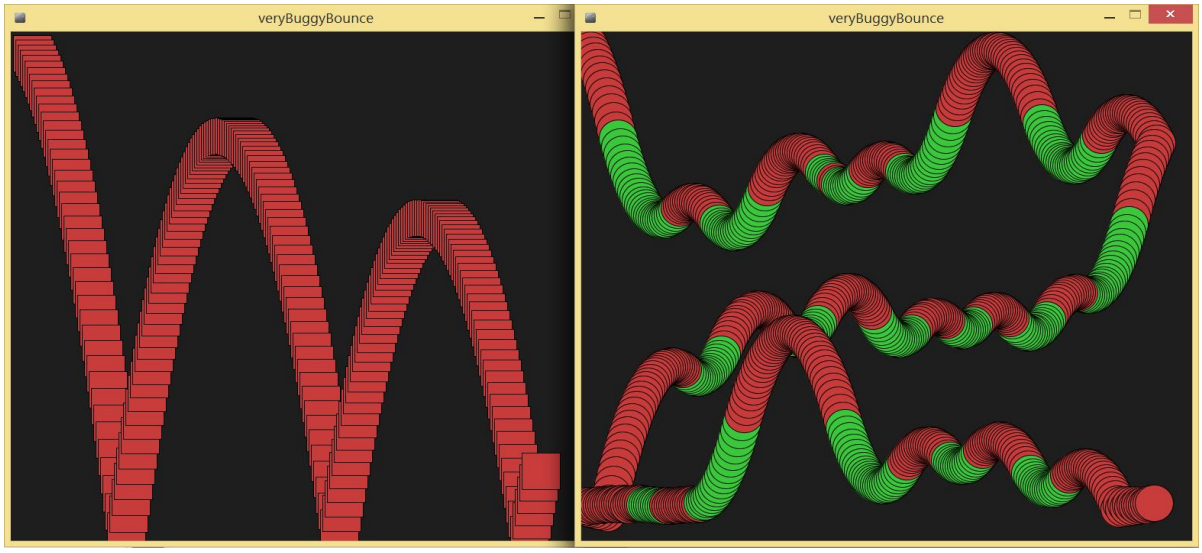

Image 4: exploratory render of impossible perspective. (1/2)

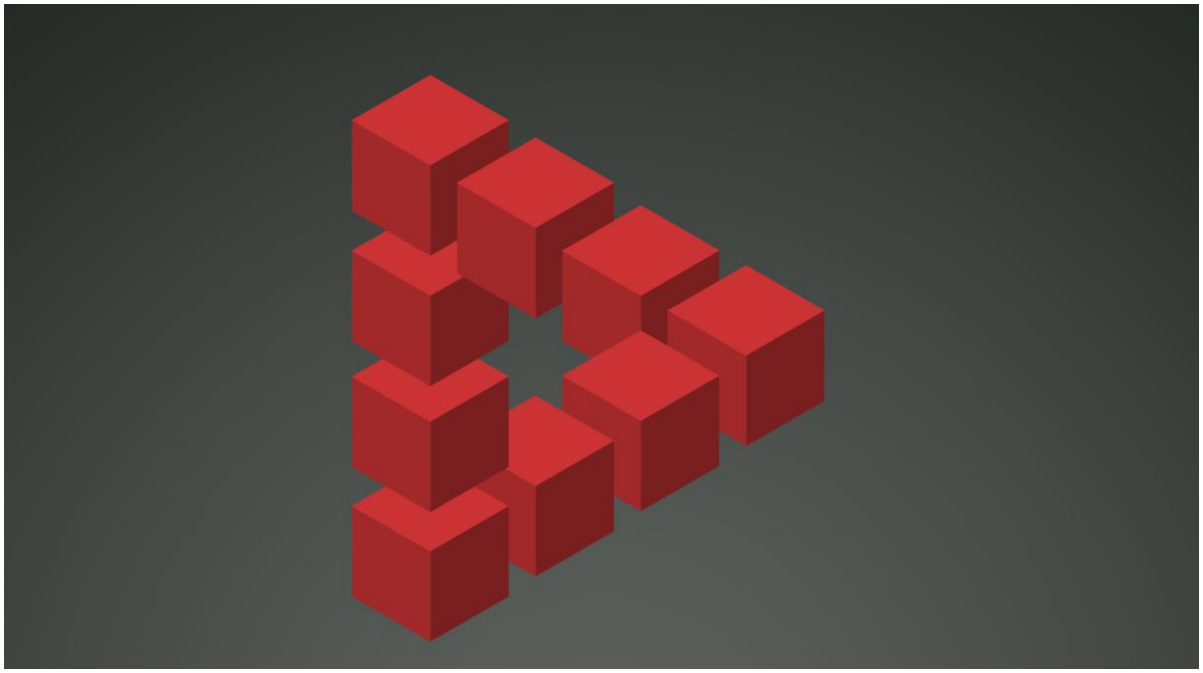

Image 5: exploratory render of impossible perspective. (2/2)

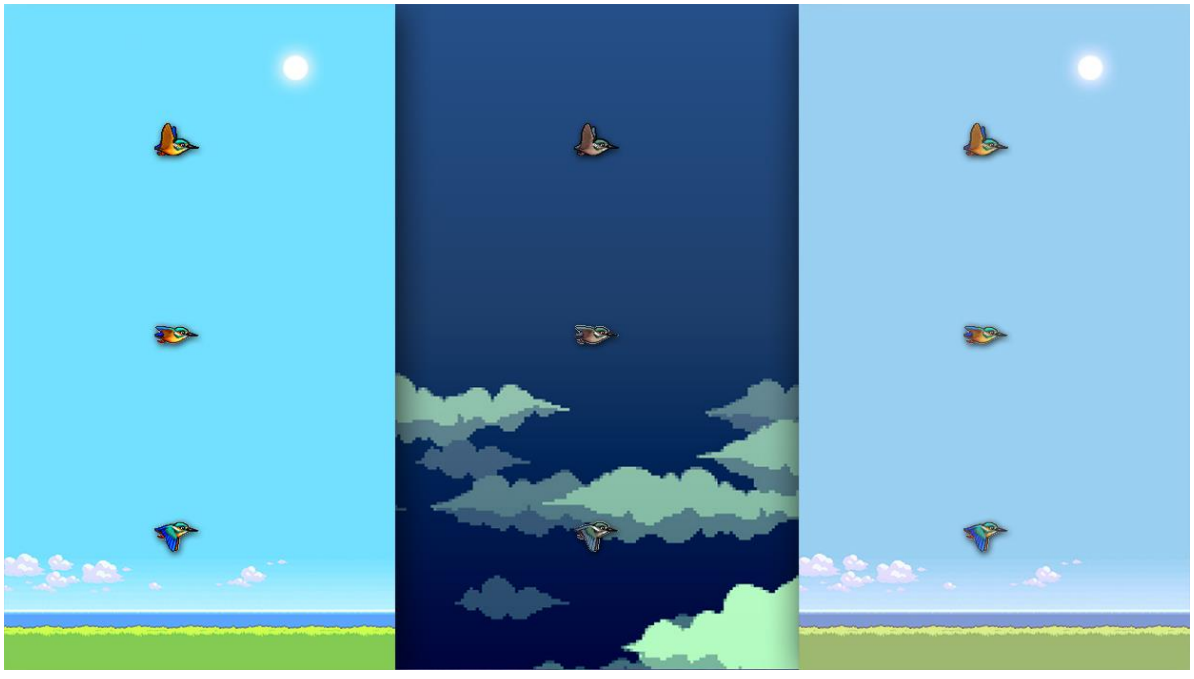

Image 6: customized graphics, created for my game.