# Final Project Report

叶茂 2200017852

## 1 关于选题

选题的灵感实际上来自于楚梦渝老师讲到 Text Visualization 时所展现的 Wordle 图片。依稀记得老师提及到词块之间有着弹簧模型连接，可以通过鼠标拖拽实现动态效果，彼时的我惊讶于所学物理仿真在文本可视化中的神奇运用，因此留下了深刻的印象。


## 2 实现思路

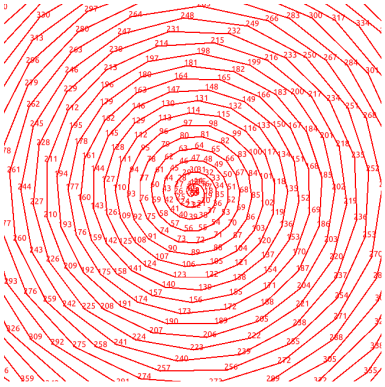
### 2.1 整体框架

WordCloud 要求将所有给定单词绘制在一个有意义的图形内，并且出现频率高的单词更大。在大体的实现思路上，我参考了这篇文章 。单词的绘制分为两部分: 位置的选择和碰撞检测。首先要为单词选择一个放置的位置，直观上权重大、频率高的单词应该居于画布中央，视觉效果最明显，因此位置函数应该从中心开始并逐步向四周扩展，阿基米德螺线便是符合这一要求的函数之一。

$$x = (a + b * \theta) * cos\theta$$

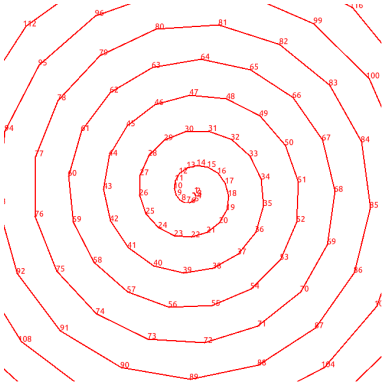
$$y = (a + b * \theta) * sin\theta$$

其次，选定位置后需要检测在该位置绘制单词是否会和已绘制的单词发生重叠。观察词云可以发现，单词之间的碰撞并不是简单的矩形碰撞，而是精细到画布上的像素是否被占用，因此我参考 Lab 5 中提供的`PrintText`函数，在绘制时进行逐像素的检查。



(a) Distance = 0.005

(b) Distance = 0.010



(c) Distance = 0.015

## 2.2 UI 设计

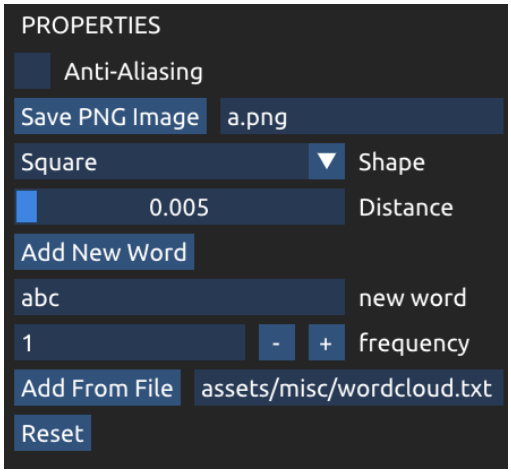
`Shape`: 采用下拉菜单，可选择方形、圆形或心形词云。

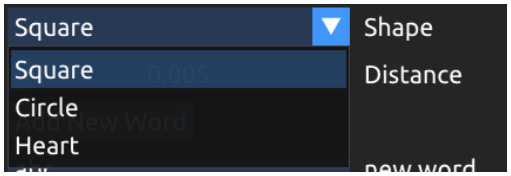
`Distance`: 该滑动条控制阿基米德螺线之间的间距，从左到右逐渐变大。间距越大，绘制的词云中，词块的分布越稀疏。

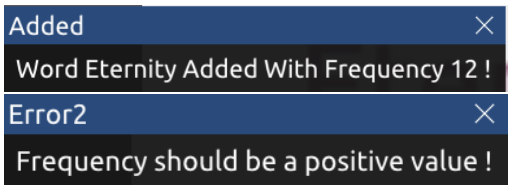
`Add New Word`: 该按钮以及下方的两个输入框用于向词云中添加新词语。添加过程实施字符检查和频率检查，当出现非法字符或频率值非正时将会出现报错，添加失败。

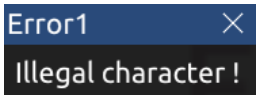
`Add From File`: 该按钮用于从文件中读取数据绘制词云。文件内容应为: 第一行为词语个数 n，随后 n 行分别为用空格分隔开的词语和频率值。若文件打开失败将出现弹窗报错提醒 (存在 bug，未实现)。

`Reset`: 此按钮用于重置词云，使画布变为空白状态。




### 2.3 功能实现

为了实现词语与频率的一一对应，以及应对单词权重变化的情况，我使用`unordered_map`数据结构储存词云数据。绘制时将其转化为`vector`按频率降序排序绘制。

词语的颜色我使用随机颜色，词语大小与其频率大小成正比，最大者为固定值。

对于图形的实现，我目前只实现了圆形和心形，在碰撞检测时判断该像素是否落在图形轮廓线内部，若在外则寻找下一个放置点。理论上，对于给定的方程确定的图形都能通过此方法使词语落在图形内。

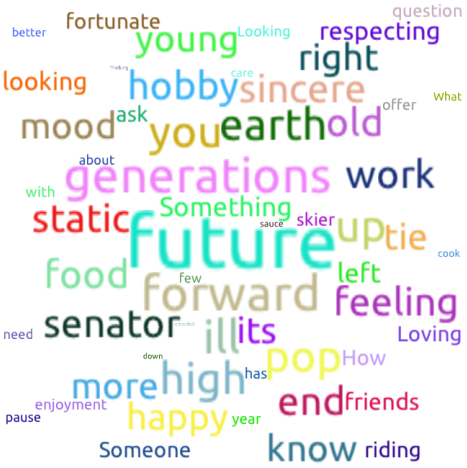
若出现某词语遍历了所有放置点仍无法放置的情况，则减小其大小，重新遍历放置点；大小下限为`lineHeigh = 0.01f;`

## 3 实现结果

### 3.1 成果展示

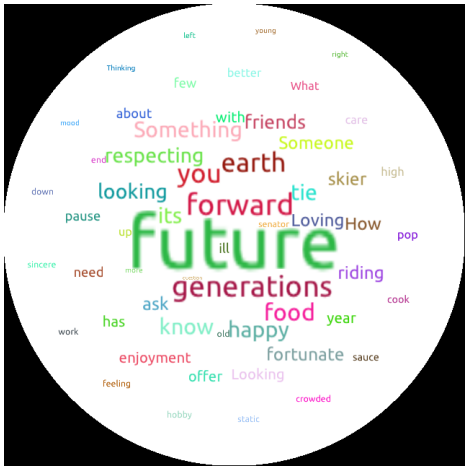
由于未能实现背景的透明化，因此为了突出形状，圆形词云图中圆形外的背景被设置为黑色 (当然也可全白)。


(a) Distance = 0.005



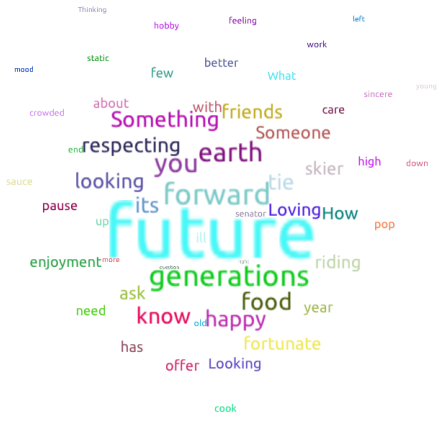

(b) Distance = 0.015


(a) Distance = 0.005




(b) Distance = 0.015


(a) heart




(b) heart

## 3.2 不足之处

除了上述提到了一些不足之外，我认为该项目仍有许多不足之处。一是没有对PrintText进行改进，在文字尺寸较大时会出现走样的现象，造成文字模糊；二是交互功能的不便性，文件读入和手动添加局限性过大，没能实现对现有词语的信息展示、动态增删、修改等功能；此外，词云的初衷应是对一段文本进行分析，提取其关键词再制作词云，由于 C++ 实现难度大且文本分析脱离了可视化的目的便没有纳入考虑；其次，课上所提及的使用弹簧模型实现拖拽词块的动态碰撞效果由于难度过大也没有纳入考虑。