20210625 CA Final Project Report

Team 16

0716207 呂思函

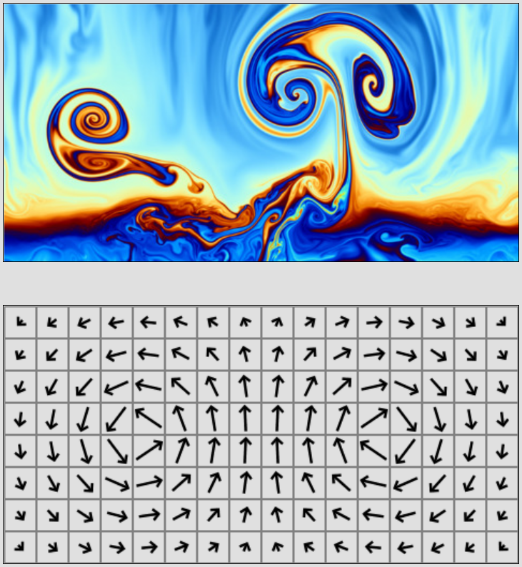
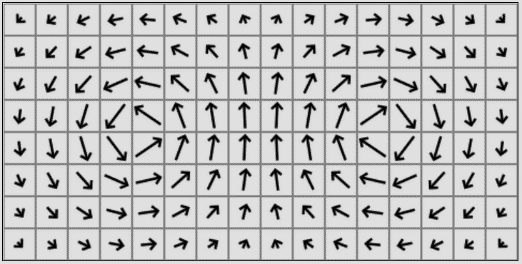
* Final\_Project\_16資料夾包含
  + Report.pdf
  + final presentation slides.pptx
  + demo video (compressed)
  + CA\_final資料夾 : 影片中間的Unity素材，直接用Unity開啟舊專案就可以執行
  + CA\_Scene.unitypackage : 片頭片尾的Unity素材，使用方式 :
    - 開啟空檔案 : Unity > HDRP
    - 匯入package : Assets > Import Package > Custom Package > CA\_Scene.unitypackage
* Title : Time Step with Sound
* Goal/Motivation

由於在oral proposal的時候，想要做的內容過於豐富，所以老師給了建議說可以選擇在其中一個領域把它做好，因此最後選擇專注操作流體模擬。

在調查資料時，偶然看到2D Stable Fluid的藝術展演([Flow exhibit (karlsims.com)](https://www.karlsims.com/flow.html))，它是藉由人與畫面的互動，就像是滑過液體那樣推動顏料，而關於「流動性」，我延伸發想到的是時間和聲音，因此，最後的demo video會包含時間、聲音、流動性這幾個元素。

* Related Work/Background
  + 2D Fluid Flow([Fluid Flow Tutorial (karlsims.com)](https://www.karlsims.com/fluid-flow.html)) : from Job Stam “Stable Fluid”.

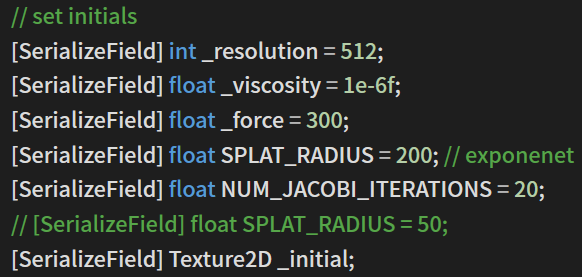
It is about fundamental of flow field, fluid momentum, divergence, boundary condition, and external forces.



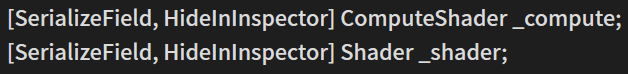
* + Flow Exhibit([Flow exhibit (karlsims.com)](https://www.karlsims.com/flow.html))
    - This exhibit presents various fluid flow, particle systems, and image processing simulations that react to visitors as they move in front of a display.
    - A Kinect depth sensor detects the shapes of people in front of the display, and allows them to manipulate the simulations with their gestures and motions.



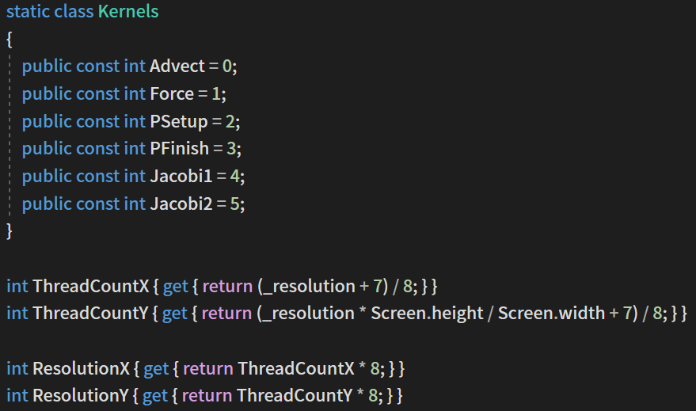
* Demo video架構
  + Demo video頭尾的動畫 : 添加時間、聲音元素(抽象的流動)
  + 片頭 : 畫面從一個寂靜無聲的留聲機開始，當相機慢慢運鏡接近留聲機時，出現了腳步聲(時鐘指針的音效)，之後燈光亮起，留聲機旋轉軸和黑膠唱片緩緩轉動，聲音也隨之播放，進入具體的流動
  + 中場 : 體的流動 : 2D Fluid Simulation & Ripples
  + 片尾 : 留聲機旋轉軸和黑膠唱片的旋轉漸慢停下，聲音也跟著停止播放，之後燈光暗去，感官可以感受的流動性皆停滯下來
* Approach/Tools
  + 2D Fluid Simulation : 流動性元素
    - Script
      * 使用者可以輸入參數



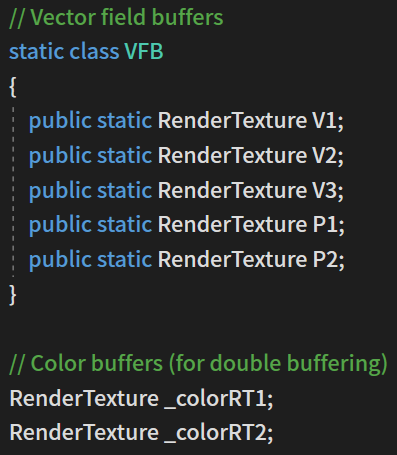
* + - * Pass parameter to shader and compute



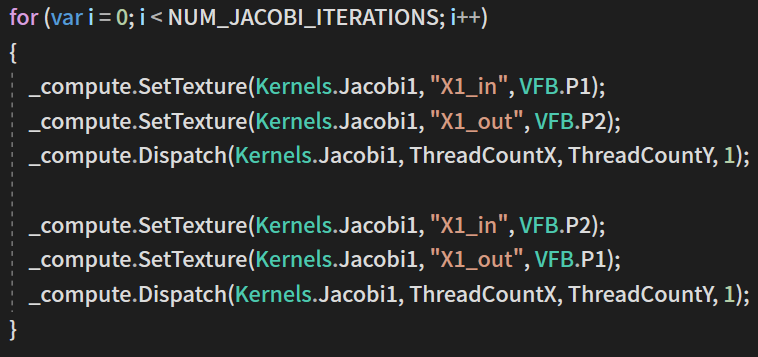
* + - * Because I use function “ComputeShader.Dispatch”to pass and compute shader, it will launch the indicated number of compute shader thread groups in the X and Y dimensions. Within each work group, a number of shader invocations ("threads") are made.

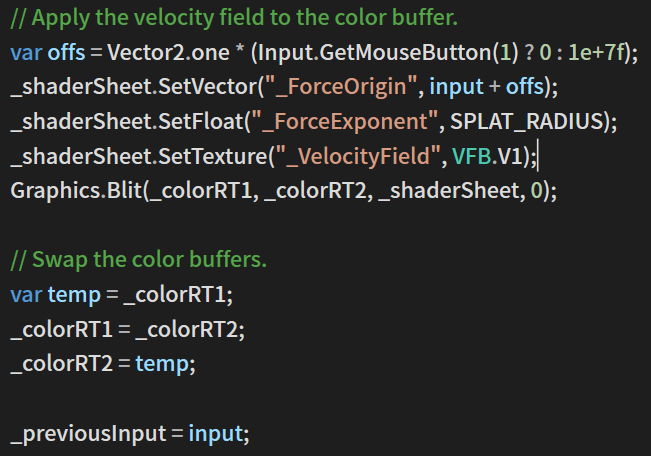


* Set buffers : Vector field buffers are for texture (picture on screen) and Color buffers.

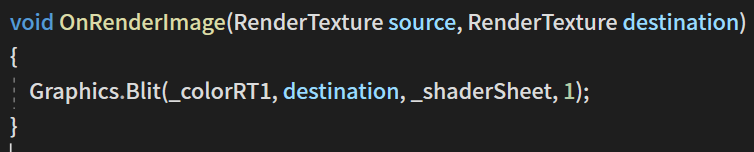


* Set values to \_compute to pass parameter to shader in function Update() by SetVector, SetFloat, SetTexture, or Dispatch :
  + input point
  + time parameters
  + advection
  + diffuse : count viscosity
  + Jacobian Iterations
  + projection
  + velocity : get velocity field, force exponent, force, and origin and compute
* 先設置初始狀況的Vector field buffer，再使用Jacobian Iterations和Projection去更新「下筆」之後的color buffer改變情況，並且記錄這一時間點的情況

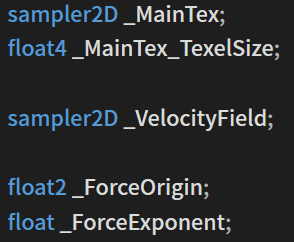




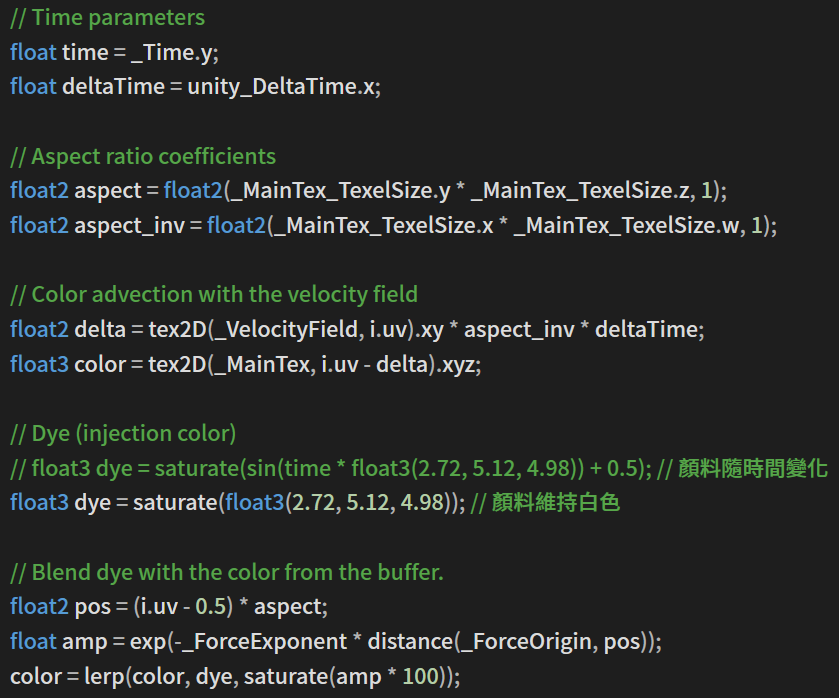
* Copies source texture into destination render texture with a shader



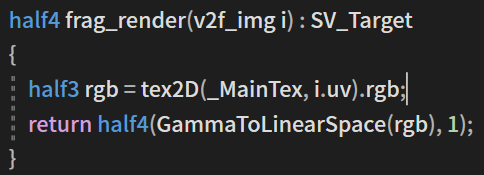
* + - Shader
      * 取得從script傳進來的參數



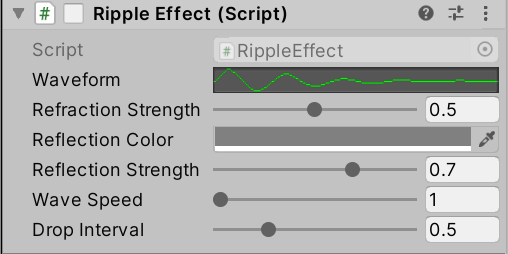
* 參考各種github上的寫法，計算各項參數



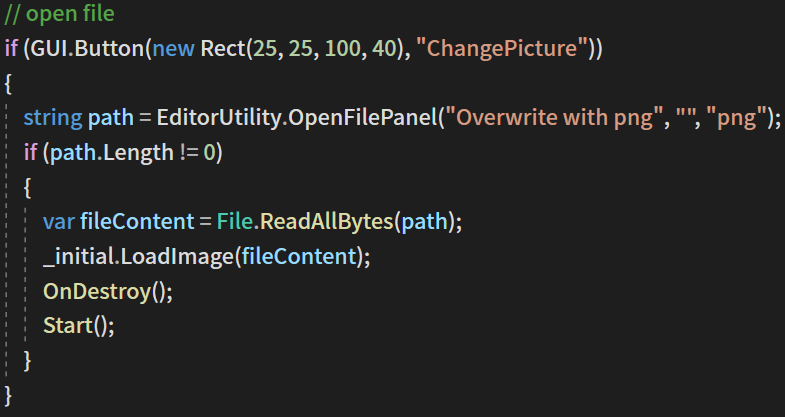
* 由fragment shader傳出變化的色彩

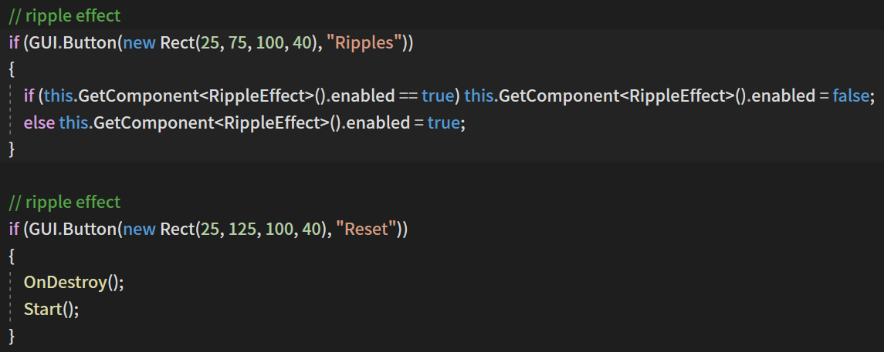


* + Ripples : 直接使用github上類似濾鏡的套件，然後再自行調整參數，只是動畫效果都差不多。

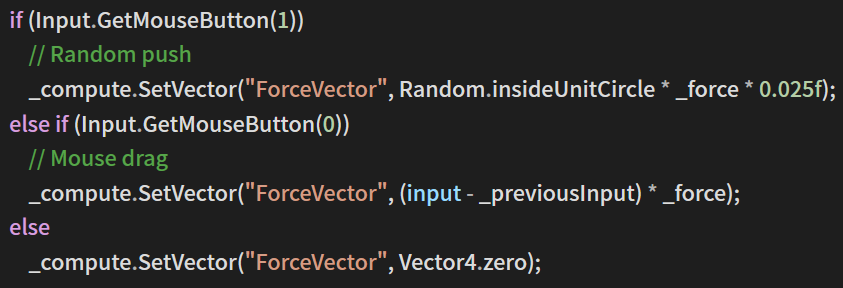


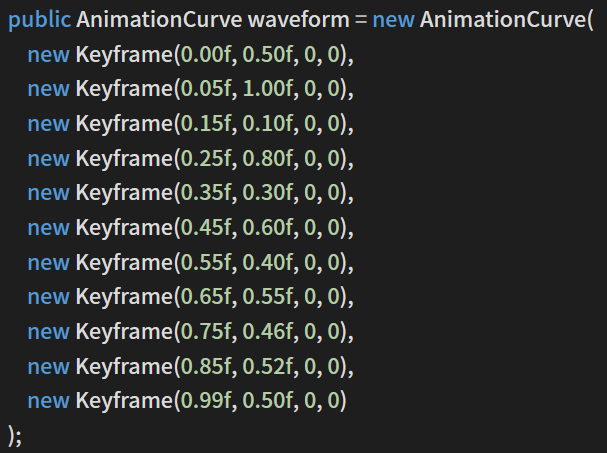
* + GUI : 按鈕功能 : 隨意選擇背景圖、漣漪、重置



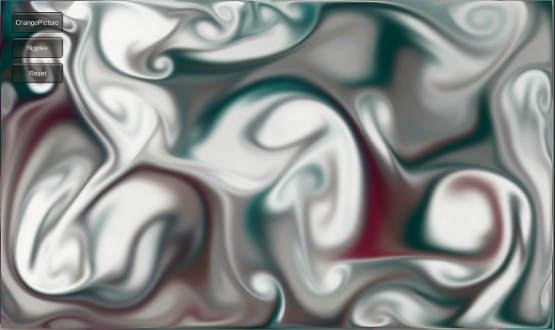
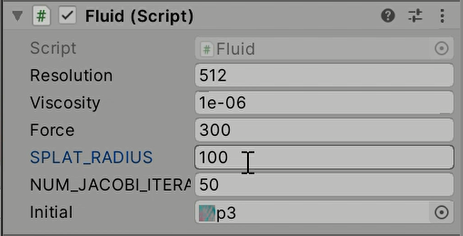


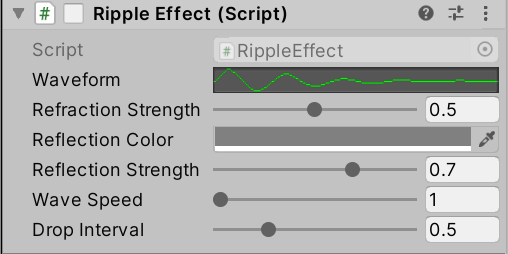
* Problems
  + 原本是要連接外掛的程式Ableton Live來控制筆畫和漣漪，但是連接MIDI控制的部分搞不定，因此後來改為 : 筆畫連接滑鼠、漣漪自動產生動畫





* + 套用的漣漪效果是一個類似濾鏡的物件，原本以為這個ripples可以推動模擬出的流體，但似乎只有視覺上的效果，嘗試改寫code卻仍然無法實現漣漪推動fluid。
* Results
* 畫了兩張成果，底下是script可以讓使用者調整參數的panel
* 資料夾中的影片是壓縮過的，未壓縮的影片連結 : <https://youtu.be/JsvRslYLc9g>





* References
  + 圖片來源 : Unity架設場景、網路
  + [ns.pdf (toronto.edu)](https://www.dgp.toronto.edu/public_user/stam/reality/Research/pdf/ns.pdf)
  + <https://www.karlsims.com/fluid-flow.html>
  + [Flow exhibit (karlsims.com)](https://www.karlsims.com/flow.html)
  + <https://github.com/arsh-khokhar/GPU-Fluid-Simulator>
  + <https://github.com/keijiro/StableFluids>
  + <https://github.com/keijiro/RippleEffect>
  + <https://github.com/omgware/fluid-simulator>