Particle System

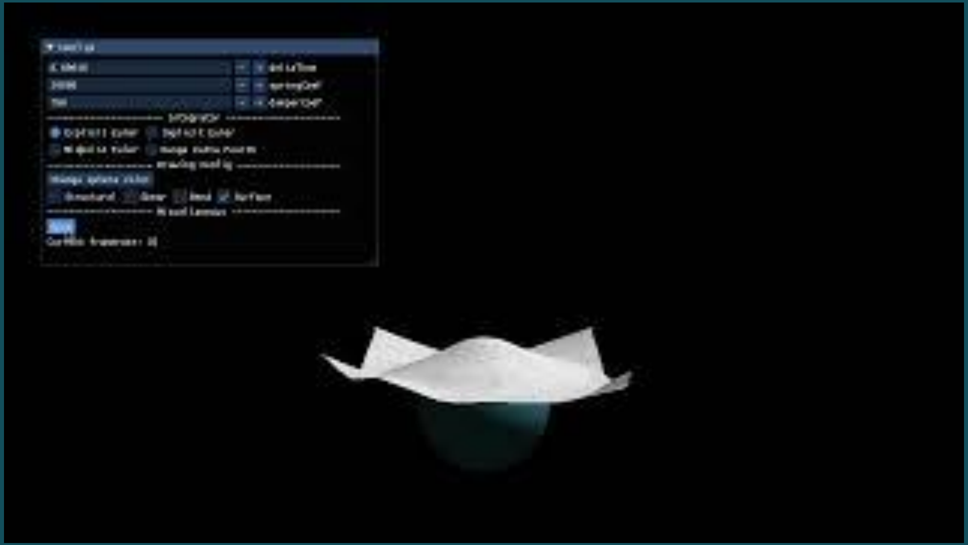
HW1
2024 Computer Animation and Special Effects

Outline

- Overview
- Environment Setup
- Objective
- Report
- Scoring
- Submission
- Note

Overview (cont.)

- Use arrow keys to move, space bar to stop the sphere
- 1 ~ 4 to pin/unpin the corners



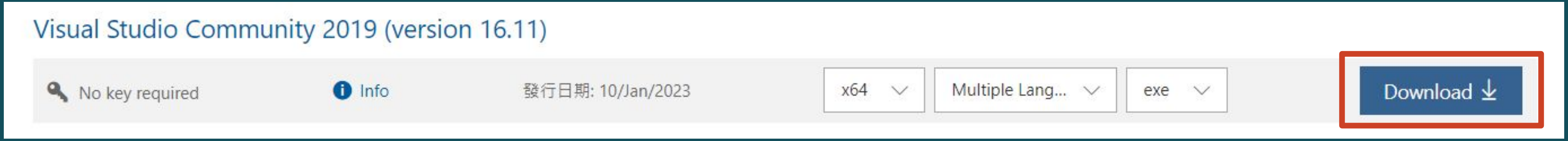
Control
[illegible] deltaTime
[illegible] springCoef
[illegible] damperCoef
Integrator
Explicit Euler · Implicit Euler
Midpoint Euler · Runge Kutta Fourth
Drawing config
Change sphere color
Structural · Shear · Bend · [✓ Surface]
[illegible]
[illegible]
Current framerate: [illegible]

Overview (cont.)

- IDE: Visual studio 2019 / Visual studio 2022
- Graphics API: OpenGL
- Dependencies
  - Eigen
  - glfw
  - glad
  - Dear ImGui

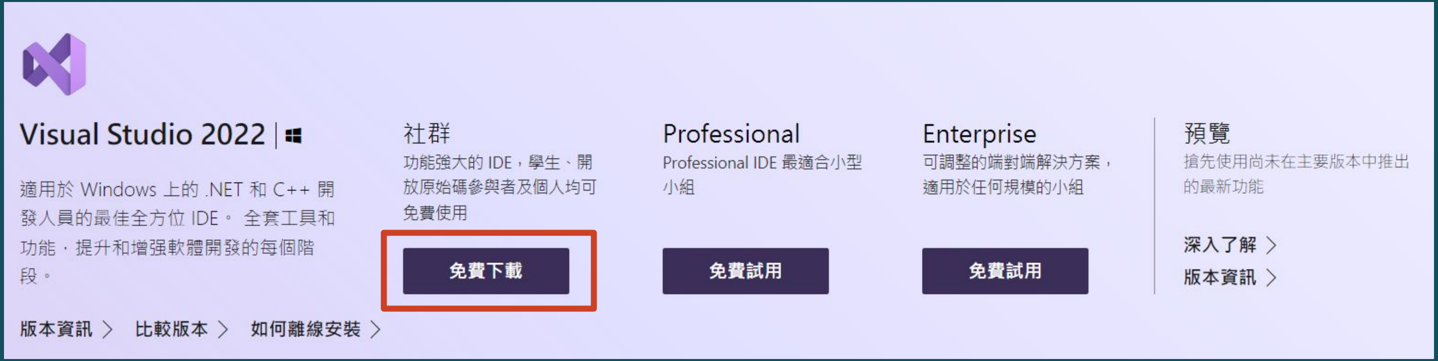
Environment Setup

- Download Visual Studio 2019 – Community or Visual Studio 2022 - Community



Visual Studio Community 2019 (version 16.11)

No key required | Info | 發行日期: 10/Jan/2023 | x64 | Multiple Lang... | exe | Download



Visual Studio 2022

適用於 Windows 上的 .NET 和 C++ 開發人員的最佳全方位 IDE。 全套工具和功能，提升和增強軟體開發的每個階段。

版本資訊 > 比較版本 > 如何離線安裝 >

社群
功能強大的 IDE，學生、開放原始碼參與者及個人均可免費使用
免費下載

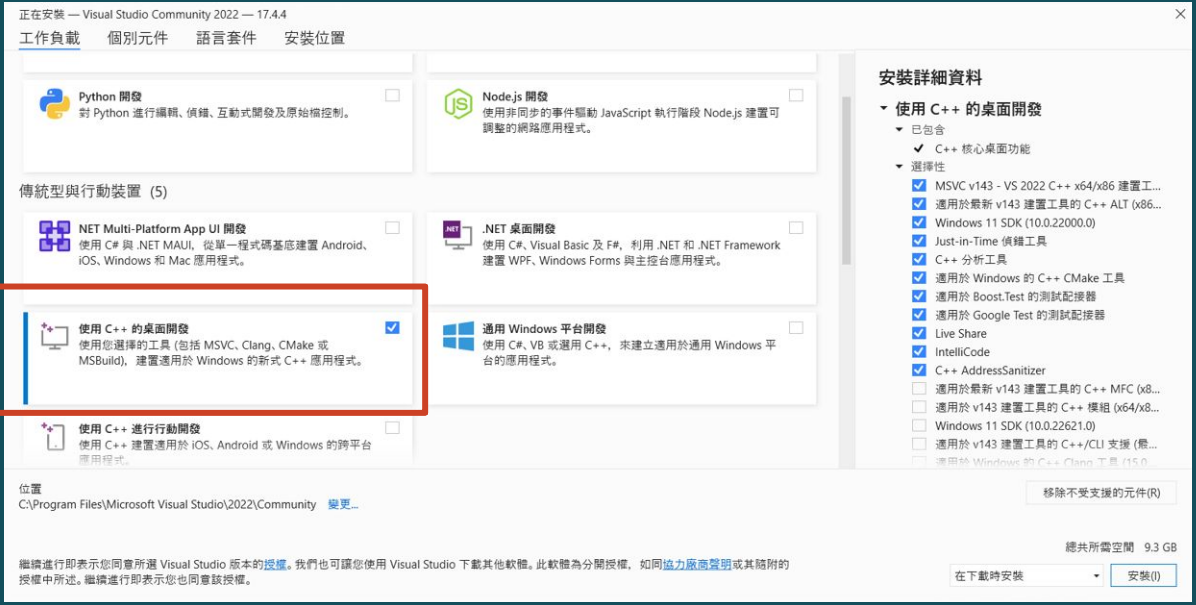
Professional
Professional IDE 最適合小型小組
免費試用

Enterprise
可調整的端對端解決方案，適用於任何規模的小組
免費試用

預覽
搶先使用尚未在主要版本中推出的最新功能
深入了解 >
版本資訊 >

Environment Setup (cont.)

- Launch Visual Studio Installer



正在安裝 — Visual Studio Community 2022 — 17.4.4

工作負載 個別元件 語言套件 安裝位置

Python 開發 [ ]
對 Python 進行編輯、偵錯、互動式開發及原始檔控制。

Node.js 開發 [ ]
使用非同步的事件驅動 JavaScript 執行階段 Node.js 建置可調整的網路應用程式。

傳統型與行動裝置 (5)

NET Multi-Platform App UI 開發 [ ]
使用 C# 與 .NET MAUI，從單一程式碼基底建置 Android、iOS、Windows 和 Mac 應用程式。

.NET 桌面開發 [ ]
使用 C#、Visual Basic 及 F#，利用 .NET 和 .NET Framework 建置 WPF、Windows Forms 與主控台應用程式。

使用 C++ 的桌面開發 [✓]
使用您選擇的工具 (包括 MSVC、Clang、CMake 或 MSBuild)，建置適用於 Windows 的新式 C++ 應用程式。

通用 Windows 平台開發 [ ]
使用 C#、VB 或選用 C++，來建立適用於通用 Windows 平台的應用程式。

使用 C++ 進行行動開發 [ ]
使用 C++ 建置適用於 iOS、Android 或 Windows 的跨平台應用程式。

安裝詳細資料

使用 C++ 的桌面開發
- 已包含
  - ✓ C++ 核心桌面功能
- 選擇性
  - [✓] MSVC v143 - VS 2022 C++ x64/x86 建置工...
  - [✓] 適用於最新 v143 建置工具的 C++ ALT (x86...
  - [✓] Windows 11 SDK (10.0.22000.0)
  - [✓] Just-in-Time 偵錯工具
  - [✓] C++ 分析工具
  - [✓] 適用於 Windows 的 C++ CMake 工具
  - [✓] 適用於 Boost.Test 的測試配接器
  - [✓] 適用於 Google Test 的測試配接器
  - [✓] Live Share
  - [✓] IntelliCode
  - [✓] C++ AddressSanitizer
  - [ ] 適用於最新 v143 建置工具的 C++ MFC (x8...
  - [ ] 適用於 v143 建置工具的 C++ 模組 (x64/x8...
  - [ ] Windows 11 SDK (10.0.22621.0)
  - [ ] 適用於 v143 建置工具的 C++/CLI 支援 (最...
  - [ ] 適用於 Windows 的 C++ Clang 工具 (15.0...

位置
C:\Program Files\Microsoft Visual Studio\2022\Community 變更...

移除不受支援的元件(R)

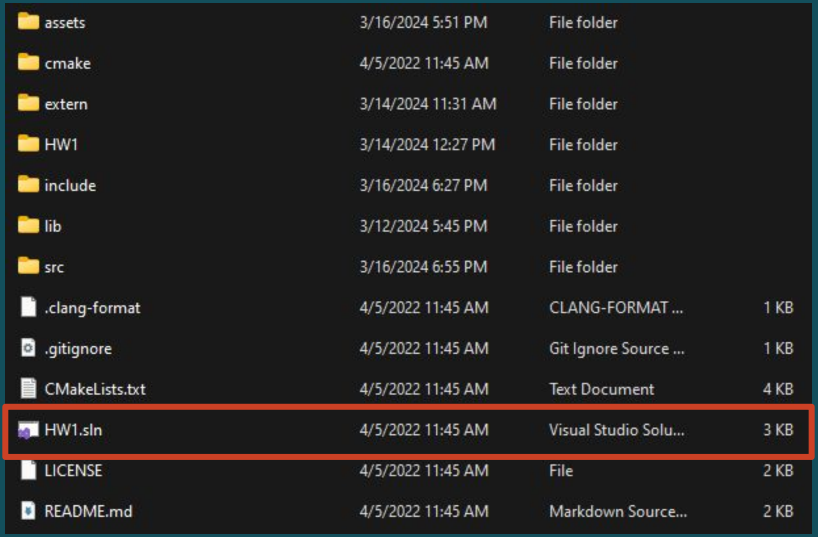
總共所需空間 9.3 GB

繼續進行即表示您同意所選 Visual Studio 版本的授權。我們也可讓您使用 Visual Studio 下載其他軟體。此軟體為分開授權，如同協力廠商聲明或其隨附的授權中所述。繼續進行即表示您也同意該授權。

在下載時安裝 安裝(I)

Environment Setup (cont.)

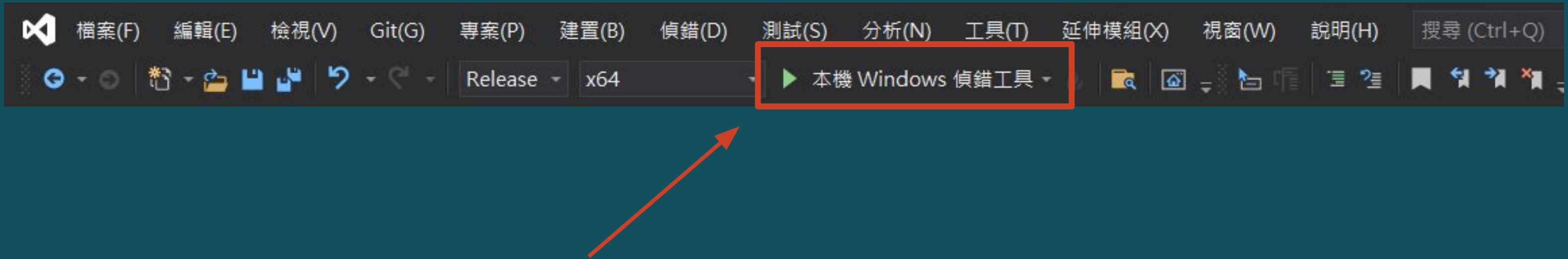
- Download HW1.zip and unzip
- Open HW1.sln



assets	3/16/2024 5:51 PM	File folder	
cmake	4/5/2022 11:45 AM	File folder	
extern	3/14/2024 11:31 AM	File folder	
HW1	3/14/2024 12:27 PM	File folder	
include	3/16/2024 6:27 PM	File folder	
lib	3/12/2024 5:45 PM	File folder	
src	3/16/2024 6:55 PM	File folder	
.clang-format	4/5/2022 11:45 AM	CLANG-FORMAT ...	1 KB
.gitignore	4/5/2022 11:45 AM	Git Ignore Source ...	1 KB
CMakeLists.txt	4/5/2022 11:45 AM	Text Document	4 KB
HW1.sln	4/5/2022 11:45 AM	Visual Studio Solu...	3 KB
LICENSE	4/5/2022 11:45 AM	File	2 KB
README.md	4/5/2022 11:45 AM	Markdown Source...	2 KB

Environment Setup (cont.)

- Run the project



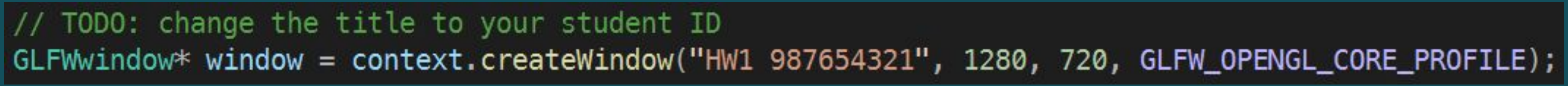
- Select config then build (CTRL+SHIFT+B)
- Use F5 to debug or CTRL+F5 to run
    - It will spend a lot of time to debug so release is recommended unless you need debugger

Objective

- src
    - main.cpp
    - cloth.cpp
        - void Cloth::initializeSpring()
        - void Cloth::computeSpringForce()
    - sphere.cpp
        - void Spheres::collide(Cloth* cloth)
    - integrator.cpp
        - void ExplicitEuler::integrate(...)
        - void ImplicitEuler::integrate(...)
        - void MidpointEuler::integrate(...)
        - void RungeKuttaFourth::integrate(...)

Objective (cont.)

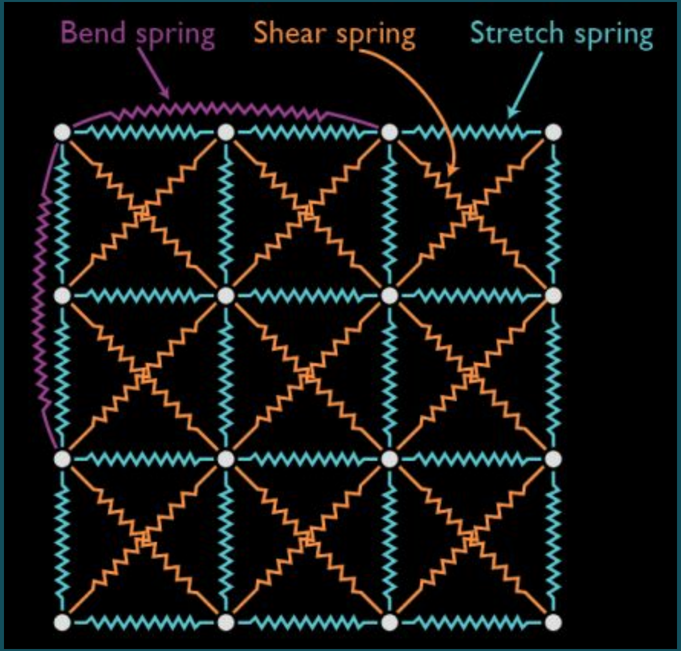
- int main()
  - Change the title to your student ID



```
// TODO: change the title to your student ID
GLFWwindow* window = context.createWindow("HW1 987654321", 1280, 720, GLFW_OPENGL_CORE_PROFILE);
```

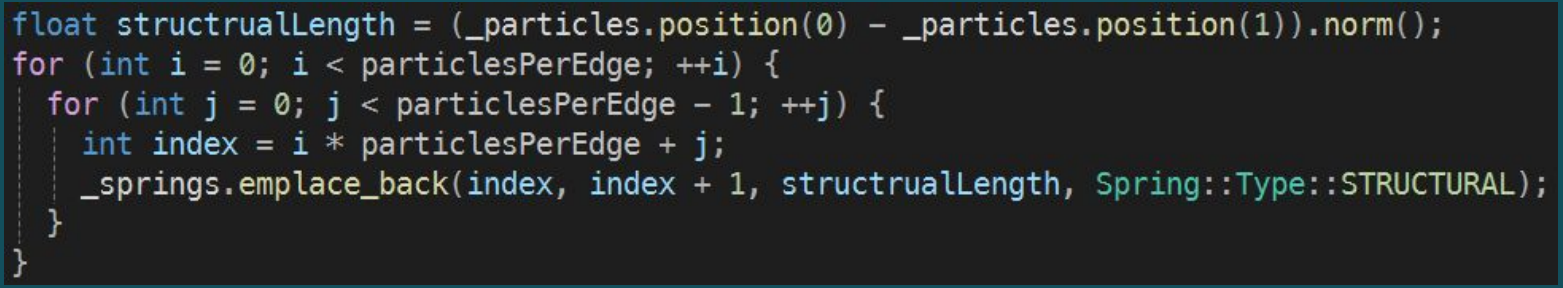
Objective (cont.)

- void Cloth::initializeSpring()
  - Goal
    - Construct the connection of springs
  - Hint
    - You should initialize three types of spring struct, shear and bending



Objective (cont.)

- Put all springs in std::vector<Spring> _springs, which is a member in class Cloth
- Here is a sample code of connecting struct springs in the horizontal direciton



```cpp
float structrualLength = (_particles.position(0) - _particles.position(1)).norm();
for (int i = 0; i < particlesPerEdge; ++i) {
  for (int j = 0; j < particlesPerEdge - 1; ++j) {
    int index = i * particlesPerEdge + j;
    _springs.emplace_back(index, index + 1, structrualLength, Spring::Type::STRUCTURAL);
  }
}
```

Objective (cont.)

- void Cloth::computeSpringForce()
  - Goal
    - Compute spring and damper forces
  - Hint
    - Review “particles.pptx” from p.9 - p.13
    - Trace every spring and apply the force accordingly
    - Set acceleration to apply the force
    - You can use float Particles::inverseMass(int i) to get the inverse of mass
    - The parameter springCoef and damperCoef are defined in config.h

Objective (cont.)

- void Spheres::collide(Cloth* cloth)
  - Goal
    - Handle collision between spheres and cloth
  - Hint
    - Review “particles.pptx” from p.14 - p.19
    - You can use their radius and distance to determine whether they are collided
    - The radius of particles of cloth can be regarded as 0

Objective (cont.)

- To compute the velocity after collision
  - You only need to worry about the component of the velocity that is in the direction of the collision
  - You can assume that the sphere is stationary
  - You can refer to this website for detailed information

$$v_a = \frac{m_a u_a + m_b u_b + m_b C_R (u_b - u_a)}{m_a + m_b}$$

and

$$v_b = \frac{m_a u_a + m_b u_b + m_a C_R (u_a - u_b)}{m_a + m_b}$$

C_R is the coefficient of restitution

Objective (cont.)

- Integrator
  - Update particles' position and velocity
  - void ExplicitEulerIntegrator::integrate(...)
    - Hint: review “ODE_basics.pptx” from p.15 - p.16
  - void ImplicitEulerIntegrator::integrate(...)
    - Hint: review "ODE_implicit.pptx" from p.18 - p.19
  - void MidpointEulerIntegrator::integrate(...)
    - Hint: review “ODE_basics.pptx” from p.18 - p.20 and “pbm.pdf” from B.5 - B.6
  - void RungeKuttaFourthIntegrator::integrate(...)
    - Hint: review “ODE_basics.pptx” p.21 and “pbm.pdf” from B.5 - B.6

Objective (cont.)

- Bonus – constrained particles (optional)
  - Review “constrainedParticles.pptx”, “clothAnimation.pptx” from p.48 - p.52
  - For example
    - A flag (0 dof along an edge)
    - A curtain (1 dof along an edge)
    - …
  - Don’t break original requirements (if it does, make an toggle for switching between requirement parts and bonus parts)
  - Mention it in your report

Report

- Suggested outline
  - Introduction/Motivation
  - Fundamentals
  - Implementation
  - Result and Discussion
    - The difference between integrators
    - Effect of parameters (springCoef, damperCoef, etc.)
  - Bonus (Optional)
  - Conclusion

Scoring

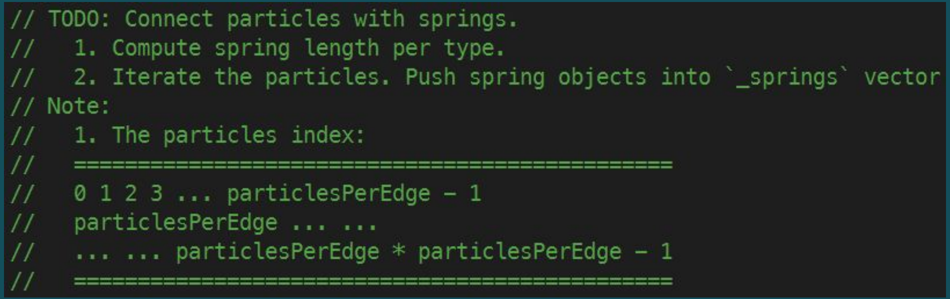
- Change window title to “HW1 **STUDENT_ID**” (0%)
    - -10% if title is wrong
- Construct the connection of springs - 10%
- Compute internal force - 25%
- Handle Collision - 20%
- Integrator - 25%
    - Explicit Euler - 5%
    - Implicit and Midpoint Euler - 5%
    - Runge-Kutta 4th - 15%
- Report - 20%
- Bonus - up to 15%

Submission

- Please upload hw1_<your student ID>.zip and report_< your student ID>.pdf respectively
- hw1_<your student ID>.zip (root)
    - src
    - include
- Late policies
    - Penalty of 10 points on each day after deadline
- Cheating policies
    - 0 points for any cheating on assignments
- Deadline
    - Monday, 2024/04/01, 23:59

Note

- Read TODOs in the template and follow TODOs' order



```
// TODO: Connect particles with springs.
//   1. Compute spring length per type.
//   2. Iterate the particles. Push spring objects into `_springs` vector
// Note:
//   1. The particles index:
//   ===============================================
//   0 1 2 3 ... particlesPerEdge - 1
//   particlesPerEdge ... ...
//   ... ... particlesPerEdge * particlesPerEdge - 1
//   ===============================================
```

- How to contact TAs?
  - Please ask your questions on new E3 forum or send email to ALL TAs via new E3
  - If you need to ask questions face-to-face, please send an email for appointment