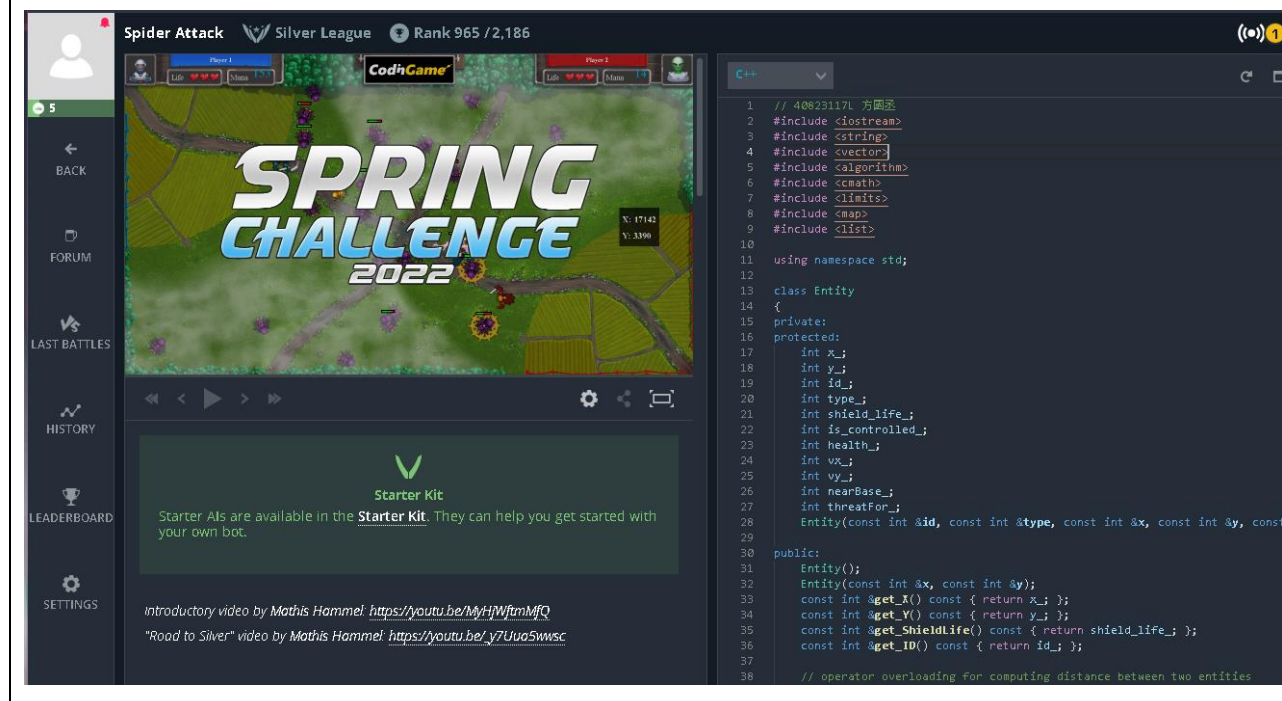


基本資料

| | | | |
|-------|------------------|----------|------------|
| 使用者 | 方國丞 | 選擇使用的類別庫 | Team 4 |
| 使用前成績 | Brone 500 ~ 1000 | 使用後成績 | Silver 965 |

使用後成績名次截圖



量化指標

| 大項指標 | 細項指標 | 同意打 √ | 總評 (1-5) |
|------|----------------------------|----------|-------------|
| 易讀性 | 類別拆解方式（有哪些類別）符合直覺。 | √ | 4 |
| | 類別、變數與函式名稱清楚，符合直覺。 | √ | |
| | 單個函式內的程式碼量適中、功能單一明確。 | √ | |
| | 避免使用難推測其意義的字面常數。 | | |
| | 避免使用特別炫技且難以理解的程式碼。 | √ | |
| | 程式碼使用一致的風格與模式。 | √ | |
| | 程式碼結構儘量扁平。 | √ | |
| 使用性 | 在有需要的地方提供適量清楚的註解。 | | 5 |
| | 提供足量、適切且容易選取的（成員）函式以達任務需求。 | √ | |
| | 函式參數個數適中、易於分辨其意義。 | √ | |
| | 容易除錯：避免使用全域變數。 | √ | |
| 擴充修改 | 容易除錯：善用 const/constexpr。 | √ | 4 |
| | 適時使用迴圈與陣列，擴充資料量時不需修改太多程式碼。 | √ | |
| | 善用具名常數，易於調整數值。 | | |

| | | | |
|--|----------------------------------|---|--|
| | (成員)函式：實作與介面切割清楚，修改時不會牽動過多其它程式碼。 | √ | |
| | (成員)函式：功能單一明確，修改時不會牽動過多其它程式碼。 | √ | |

質性意見

這個函式庫的文件很清楚，都在 github 上，變數與函式的命名方式也符合直覺，易於理解。函式內的程式碼量不會太多，閱讀起來也算容易，功能單一明確，程式碼風格一致

但在一些地方，如下圖，程式碼中直接使用數值，例如 1280、2200 等。這些數值應該用具名常數替代，才方便做調整。

```
bool canWind(const Hero &a, const Entity &b)
{
    return ((a - b) < 1280 && my_mana > 10 && b.get_ShieldLife() == 0);
}
bool canControl(const Hero &a, const Entity &b)
{
    return ((a - b) < 2200 && my_mana > 10 && b.get_ShieldLife() == 0);
}
bool canShield(const Hero &a, const Entity &b)
{
    return ((a - b) < 2200 && my_mana > 10 && b.get_ShieldLife() == 0);
}
```

還有註解可以再多一些