

# CPPSDL & SGE2

Az CPPSDL & SGE2 egy SDL2 wrapper könyvtár, ami az SDL2 és az SDL kiegészítő könyvtárainak fő funkcionalitását C++ osztályokba rendezi (CPPSDL). Ezen felül egy objektum + komponens alapú 'keretrendszer' biztosít egyszerű játékok elkészítéséhez.

A CPPSDL által lefedett funkcionalitás:

- SDL inicializálás
- Window + Renderer létrehozása
- Textúrák betöltése (SDL\_Image könyvtár)
- Szöveg textúrává alakítása (SDL\_TTF könyvtár)
- Hangok betöltése és lejátszása (SDL\_Mixer könyvtár)

Az SGE2 ezen funkcionalitásokat használja fel. Fontos megemlíteni, hogy a felhasználó ebben az esetben egy programozó, így az elvárt bemenet C++ kód, tetszőleges formában, amelyet jelen esetben a könyvtárral együtt fordítunk. A könyvtár alaposztályokat ad komponensekhez (*class Component*) és *gameobject*ekhez (*class GameObject*), amelyeket a felhasználó bővíthet(öröklődés).

Minden gameobject-et a Game (manager osztály) birtokol (unique\_ptr vector). Az objektumokról létrehozáskor kapható referencia, valamint később id alapján is. A komponenseket hasonló módon az egyes gameobjectek birtokolják.

Minden komponens konstruktora átvesz egy GameObject referenciát és egy std::string id(ez lehetőleg unique legyen a többi komponensre nézve)-t (előbbi a root objektum referenciájához, a második a komponens azonosításához szükséges). Az előbbit a komponens már objektumhoz adáskor automatikusan megkapja.

Minden komponens rendelkezik egy Startup és egy Update metódussal. Előbbi az komponens létrejöttkor egyszer (A komponens létrehozása után), utóbbi minden framen lefut. Ezen metódusok felülírásával tud a felhasználó egyéni funkcionalitást adni a programjához.

A könyvtár biztosít néhány előre elkészített komponenst, például egy *Renderer* komponenst, amely egy textúrát renderel a *gameobject* pozíciójára.

Minden *gameobject* komponensekből és egy transformból (pozíció + méret + forgatás) áll.

A *gameobject*ekből lezármazott classok a „prefab” -ek. Céljuk, egy *gameobject* blueprint felépítése (pl. ne kelljen minden ellenséghez egyesével hozzáadni ugyanazokat a komponenseket).

A főprogram egy *init.hpp* filet includeol, ebben a fileban deklarált *init()* függvényt hívja a játék indítása előtt. Az *init()* függvényt az *init.cpp* fileban kell definiálni. Ebben a függvényben történik a *gameobject*ek és komponensek létrehozása és játékhoz adása. A felhasználó ezen felül nincs megkötve a fájljainak rendezésében.

A könyvtár tartalmaz egy egyszerű példát, amely bemutatja egy project felépítését:

- *example.h*: egy példa prefab és egy példa komponens.
- *init.cpp*: példa a *gameobject*ek játékhoz adása és komponensek *gameobject*hez adására.

A könyvtár példaprogramja konzolos formában egy unit testet futtat, míg ablakos formában egy példát mutat a renderelés és irányítás működésére (*example.hpp* + *example.cpp*).