An interface for the Epitech Arcade project

This interface was designed over the course of a week and a half in order to get the best starting point possible for the arcade project.

Design

This library was designed in a way such as to attempt to make it as easy as possible to implement games that work fine in both text and graphics mode.

This is why there is such concepts as cells in the library. These are made such that the text mode display libraries can work properly (e.g. in ncurses, one cell is one character on the screen).

Positions calculated in cells or pixels should be easy enough to convert between each other, such as in this example code:

```cpp
// This function converts a raw position (i.e. a position as measured in pixels) into a cell position (i.e. a position as measured in cells)
inline ICore::Vector2u rawPixelPositionToCellPosition(ICore::Vector2u rawPosition, std::uint32_t pixelsPerCell)
{
    return {rawPosition.x / pixelsPerCell, rawPosition.y / pixelsPerCell};
}

// This function converts a cell position to a raw position
inline ICore::Vector2u cellPositionToRawPixelPosition(ICore::Vector2u cellPosition, std::uint32_t pixelsPerCell)
{
    return {cellPosition.x * pixelsPerCell, cellPosition.y * pixelsPerCell};
}
```

```
ICore::Vector2u getSquareCenter(ICore::Vector2u squareTopLeftPos, ICore::Vector2u squareSize

{

   return {squareTopLeftPos.x + ((squareTopLeftPos.x - squareSize.x) / 2), squareTopLeftPos.y + ((squareTopLeftPos.y - squareSize.y) / 2)};

}
```

We also made the decision of having the game modules only ever be able to interact with the `ICore` interface, such as to prevent any misuse of the `IDisplayModule` interface. The `ICore` interface exposes much of the same interface as `IDisplayModule` (with a few other additions), save for a few functions which cannot be exposed safely (or are redundant). If you want to understand why this decision was made, see such functions as `IDisplayModule::loadTexture`, which was the motivating example for this decision.

How to implement this interface

The core application (i.e. the program that loads dynamic libraries and is always present) needs to implement the `ICore` interface.

Each graphics shared library needs have an implementation of the `IDisplayModule` interface, and then implement the `gEpitechArcadeGetDisplayModuleHandle` function, which returns a unique_ptr to such an interface.

Each game shared library needs have an implementation of the `IGameModule` interface, and then implement the `gEpitechArcadeGetGameModuleHandle` function, which returns an `std::unique_ptr` to such an interface.

Contributing

Randomly talking in PMs on Discord does work in some ways, but using the Github Issues or Pull Requests system would be nice if you have concerns about some part of the interface. This allows these conversations to be public and to see whenever someone does something on the repo by using the GitHub Watch functionality (see the top right of the GitHub page).