Ubisoft Toronto NEXT: Programming Challenge

We will make the API available to download on Friday, November 21*st*, 2025, and release the challenge theme & brief on Friday, January 16*th*, 2026.

Your submission must be a Windows application **written entirely in C++**. Regardless of your operating system, you will need to download CMake. This is a build system generator which allows us to build the project for multiple operating systems.

For Windows users:

You will need to download Microsoft Visual Studio 2022 to successfully complete this challenge. We recommend Visual Studio Community 2022.

To build the project, run the ***generate-windows.bat*** script by double clicking on the file (this will launch a Command Prompt window while the script is running). Once the script finishes, you will see a new folder called ***build*** was created.

Under build/win64 you will find the .sln file (named UbiTorContestAPI) which you can double click to launch Visual Studio with the project. From this point on all work on the submission should be done in Visual Studio and Visual Studio will be used to compile the code and build the game.

For Mac users:

You will need to download Homebrew which is a package manager for the Mac operating system. Once this has been downloaded, open a Terminal application window to use Homebrew to download the freeglut and SDL3 libraries. To do this type the following into the Terminal window one at a time (i.e. wait for the first installation to finish before starting the second one) (1) ***brew install freeglut*** and (2) ***brew install SDL3***. You will also need the XCode command line tools. These should already be installed on your computer, but if not in the same Terminal window run the following command to install them: ***xcode-select –install***.

To build the project run the generate-macos.sh script. To do this open a Terminal window (or use the one you had open before to install all the libraries) and navigate to the location of the API folder. From there if you type ***./generate-macos.sh*** the script will start to run. Once the script finishes, you will see a new folder called ***build*** was created.

Whenever you make any changes (and when you first generate the build), you will need to navigate to the ***build/macos*** folder and run ***make all*** from it. This will compile all of the changes that were made.

To launch the game, from the same ***build/macos*** folder run the ***make run*** command. This will start the game window. To close the window, you will need to use ***Cmd+Q***.

For development we recommend you download and use Visual Studio Code. This application can have a Terminal window open as part of your application's window (to do this navigate to View >

Terminal) so you can easily run the ***make all*** and ***make run*** functions from the same application that you are writing your code in.

Adding files/game assets:

Whenever you want to add a new code file to the project, it will need to be placed inside the ***src/Game*** folder.

Whenever you want to add any new game assets to the project, they will need to be placed in the ***data*** folder (feel free to use the ***TestData*** folder or create your own).