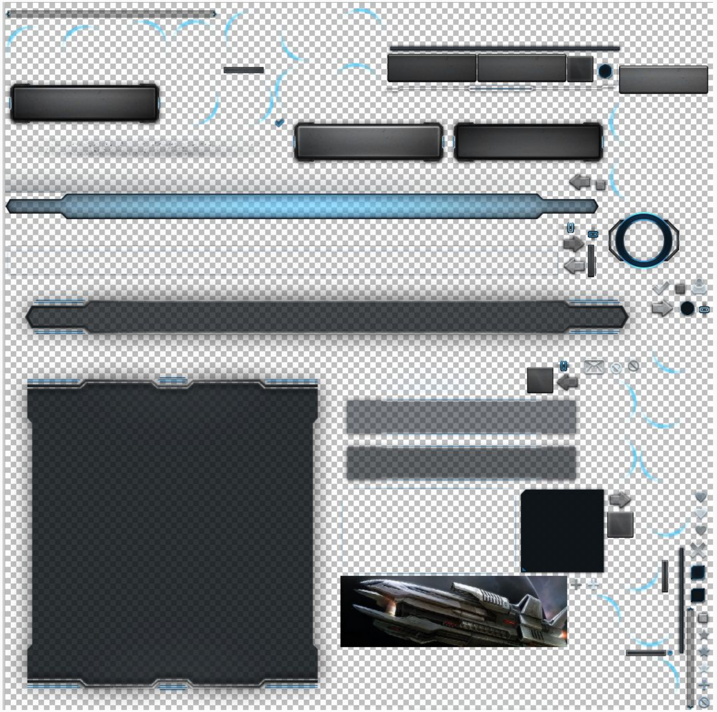
Game Development

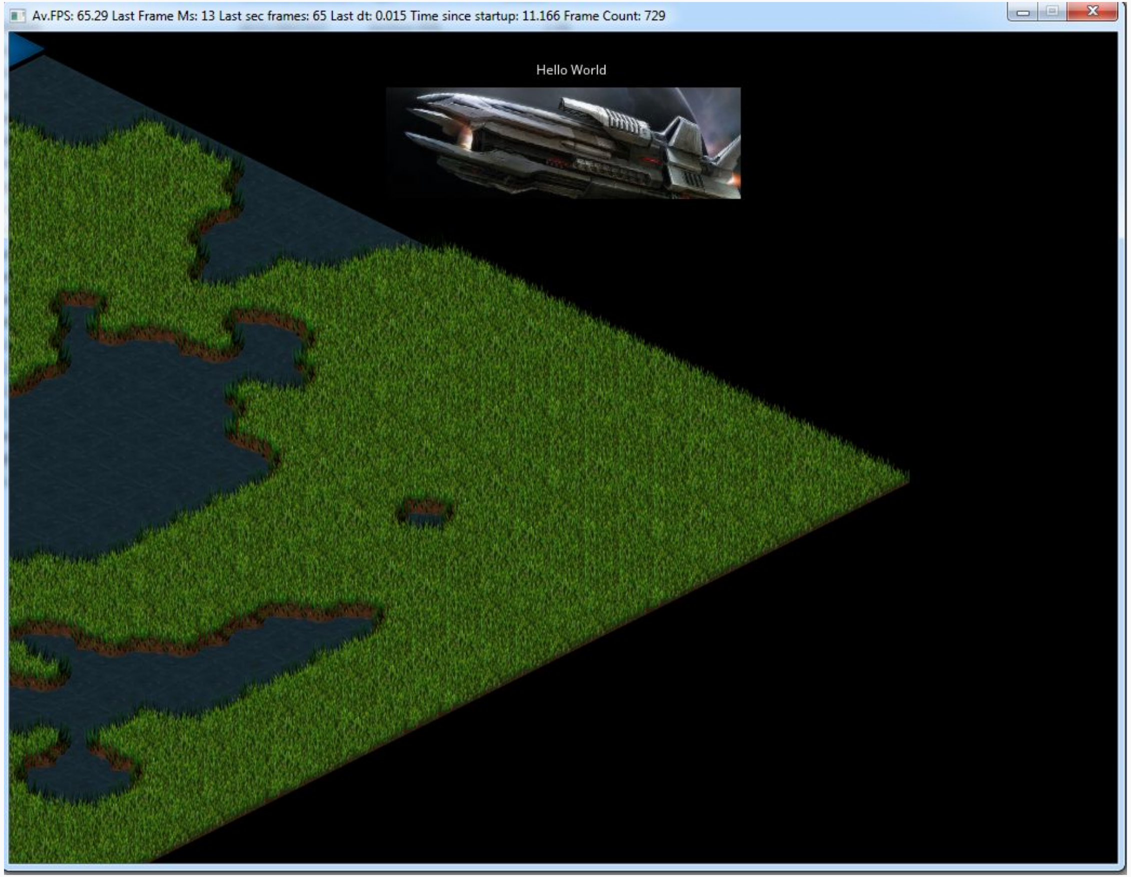
UI Basics

User Interface in video games

- Grab your UI structure and implement it to achieve the first results
- In the UI the spritesheet is normally called an “atlas”
- Your goal is to **only** create the result of solution.exe
- *ModuleFont* is already created (including a default font in fonts/ folder)
- *ModuleGUI* is just an empty skeleton but deals with atlas loading
- Atlas included in gui/ folder

Atlas and expected result





Av.FPS: 65.29 Last Frame Ms: 13 Last sec frames: 65 Last dt: 0.015 Time since startup: 11.166 Frame Count: 729

TODO 1

“Create your structure of classes”

- Now apply your design choices to code
- You could use another file instead
- Keep it simple for now

TODO 2

“Create the factory methods”

- Those are the methods that external modules can use
- They should return a pointer to be used to work with the UI element
- Remember that each **new** must have a **delete** somewhere!

TODO 3

“Create the image (rect {485, 829, 328, 103}) as a UI element”

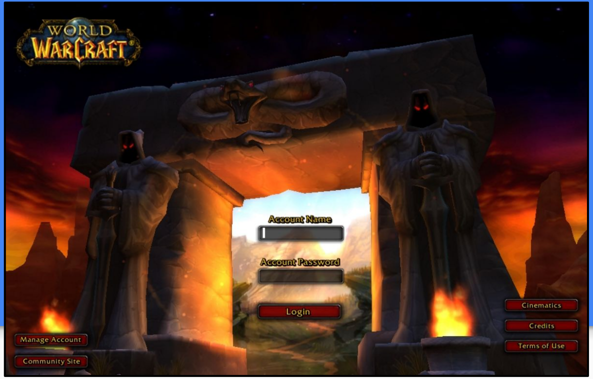
- Now let’s produce the same result as solution.exe
- Just create the UI elements and they should work automatically

TODO 4

“Create the text "Hello World" as a UI element”

- The text is 25 pixels above the image centered in its middle
- To print text in the screen:
  - Module Fonts allows us to load a font file (TTF)
  - “Print” will generate a texture based on a string and a loaded font
  - Just use that texture to draw it on the screen

Homework



- Add a factory method to create an image from a file that is not in the atlas
- Try emulating the World of Warcraft login screen! (all but input text)
- You can find a WoW UI art package in campus web
- Including original WoW fonts