



A few notes...

- Feel free to interrupt!
- Slides are at https://icculus.org/SteamDevDays/
- Today is a high-level overview.





- Hacker, game developer, porter
- Port games, build tools
- Freelance
- 15 years experience













- New, unsaturated market
- Low barrier to entry
- No walled garden
- SteamOS and Steam Machines

MythBusting



- Distro fragmentation isn't real
- Most hardware is supported
- GPU drivers are good
- Linux users spend money on software
- Tech you need is available for Linux





- Middleware is largely available
- Engines you use (Unity3D, etc) work
- Tools you use (Perforce, etc) work
- Cool new tech (Oculus, etc) work



The Bad News

• If you <u>need</u> Visual Studio, this is harder.



So wait...no Visual Studio?!

- Emacs and vi
- Sublime Text
- QtCreator
- Code::Blocks
- Eclipse
- Many other options



The Porting Process



Start with Windows

- Move to SDL2 on Windows
- Move to OpenGL on Windows
- Then start port to Linux





- GCC, Clang, Intel C++
- · Makefiles, CMake, Premake, scons



Mining from VS2008 .vcproj

```
<File
RelativePath="src\audio\player.cpp"</pre>
```



Mining from VS2010 .vcxproj

<ClCompile Include="src\audio\player.cpp" />



#define STUBBED(x) printf("STUBBED: %s\n", x)



```
#define STUBBED(x) do { \
  static bool seen_this = false; \
  if (!seen_this) { \
    seen_this = true; \
    fprintf(stderr, "STUBBED: %s at %s (%s:%d)\n", \
            x, ___FUNCTION___, __FILE___, __LINE___); \
} while (0)
```



MessageBox(hwnd, "Out of memory", "Error", 0);



```
#if WINDOWS
MessageBox(hwnd, "Out of memory", "Error", 0);
#else
STUBBED("Need a Linux msgbox");
#endif
```



Don't do this.

```
#if LINUX some_non_windows_thing(); #endif
```



Do this!

```
#if !WINDOWS
some_non_windows_thing();
#endif
```



Definitely don't do this.

```
#if WINDOWS
some_windows_thing();
#elif PLAYSTATION
some_ps_thing();
#endif
```



Do this!

```
#if WINDOWS
  some_windows_thing();
#elif PLAYSTATION
  some_ps_thing();
#else
  #error Oh no.
#endif
```



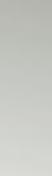
Inline assembly

```
_asm {
    mov ecx, 1
    mov eax, ecx
}
```



Inline assembly

```
_asm___volatile__(
"movI $1, %%ecx \n"
"movI %%ecx, %%eax \n"
: "a"
  :/* no inputs */
: "ecx"
```





Inline assembly

- Don't use inline asm. Ever.
- Seriously, don't do it.
- Compiler intrinsics
- SDL2 atomics, SDL_GetTicks(), SDL_GetPerformanceCounter(), etc.
- nasm





- Stub out problem code
- Fix simple things now
- Don't touch anything else (for now)





- Middleware
- System dependencies





- Use SDL2!
- Use OpenGL!
- (maybe) Use OpenAL!
- Use the Steam Runtime!



Filesystem gotchas

- Its paths are '/' instead of '\\'
- It's single root, no drive letters
- It's more locked down
- It's multiuser
- It's always Unicode (sort of)
- It's case-sensitive!





- All system calls and filenames are UTF-8
- "wchar_t" is UCS-4 (32 bits per codepoint)
- Memory bloat isn't usually an issue, serialization is
- iconv, or roll your own



• GDB 7

Get it debugged

- UndoDB
- WinGDB
- VisualGDB
- QtCreator
- Valgrind
- LLVM's AddressSanitizer

See next two talks in this room!



Debugging OpenGL

- ApiTrace
- gDebugger
- See Jason/Rich/Dan's talk, immediately following this one.





- perf
- Zoom
- ApiTrace
- Telemetry





- Simple tarballs
- MojoSetup
- Ship it on Steam!
- Avoid distro packages





- https://icculus.org/SteamDevDays
- icculus@icculus.org
- @icculus on Twitter
- http://gplus.to/icculus
- · (If you hire me, you can forget this whole talk!)



Other talks to see

- Moving your game to OpenGL: 3pm today, this room
- Getting started Debugging on Linux: 4pm today, this room
- Breakout: Programming on Linux, OpenGL: 5pm today, this room
- Beyond Porting: 3pm tomorrow, this room
- Game Development with SDL 2.0: 5pm tomorrow, Room 611/613