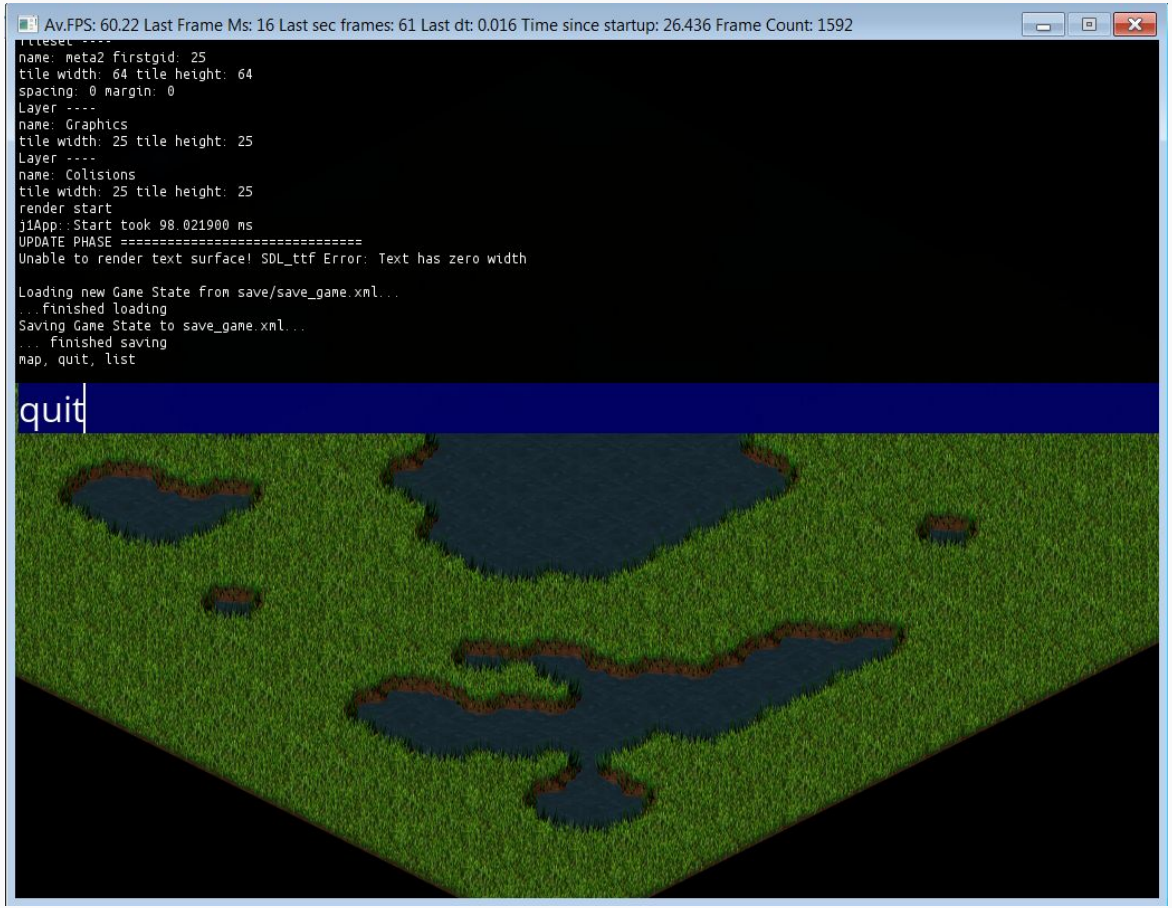
# Game Development

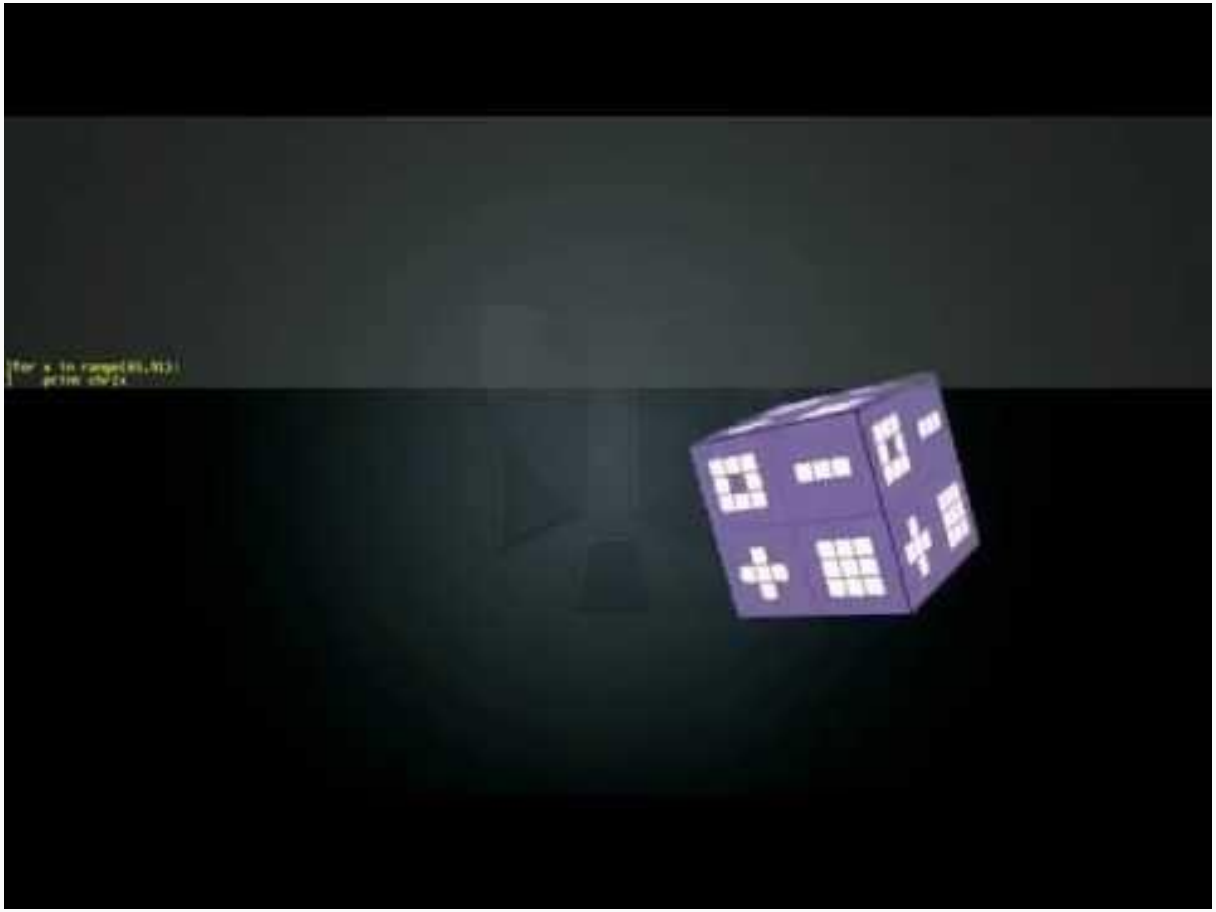
Quake Console

# Check solution.exe for a WIP console implementation




I. Menorad
0
2
0:00
100
999
998
999
985
Lightning Gun
100
998
55


Crysis 3
Dev Console



Custom Engine Console

# General Implementation

- Create a new Module for the Console
- It should use the UI Module to create needed Labels and Input Text
- Listen to input so on º/ESC the console should open / close
  - Tilde should be SDL_SCANCODE_GRAVE
- It should accept scroll with center mouse
- Re-route all LOG entries to the output
- You can have it define its properties (size, colors, font) in config.xml

# Implementing Commands

- Create a small class that represents a command
  - Which properties would you use for it ? (name, amount of arguments, … ?)
- The console should process input the same way Zork parsed commands
- Now each module should be able to generate its own commands:
  - Module Render: turn on/off sync, fullscreen, etc..
  - App: quit, max fps, pause, load level, etc…
- Have a function to generate a new command

```
App->console->CreateCommand(const char* command, j1Module* callback, uint min_arg, uint max_args);
```

# Extra Features

- Allow mixing uppercase and lowercase as commands
- Pressing UP arrow should write the previous command in history
- Pressing DOWN arrow should write the next command in history
- Pressing TAB should try to autocomplete
  - Only if there is 1 hit while you search for commands, if not... make a beep sound?
- Allow argument packing with quotes

```
]log “hello world” ERROR
```

# Homework

1. Create a console that appears on a keystroke (ESC / º)
2. It should contain a scrollable text with all available LOG
3. Have a subsystem to create and parse commands
4. Create the following commands (with Module responsible):
    a. **log** (App): will log all arguments as one string
    b. **help** (Console): list all console commands
    c. **save/load**(App): save/load the state of the game into/from a file given by an argument
    d. **bind** (Console): bind a key to a console command (*bind f1 “god_mode 1”*)