Some pointers on Code Quality

by Hans Wichman

Some pointers on code quality

- **Do's:**
  - Use Consistent Coding Conventions
  - Use explicit access modifiers: private, unless ...
  - Use descriptive self documenting class/method/variable names
  - Use documentation that explains the why & what where necessary
  - Use short/readable/cohesive classes/methods that do only a couple of things (aka a practical applied version of the single responsibility principle)
  - Use whitespace, newlines and tabs for layout and indenting (or *auto-format* your code!)

- **Don'ts:**
  - No god classes (no single Main class with 5000 lines of code)
  - No god methods (no methods that are 500 lines of code)
  - No ifception (no 5 times deeply nested if, for, while, etc structures)
  - Don't repeat yourself → Move duplicate code into a utility or base class
  - Don't use magic values (see C# Essentials)

Documenting your code

- Imagine this piece of code:

```
//check health, weapon status and stunned
if (health > 0 && hasWeapon && stunnedCounter < 0) {
        //...
}
```

- Although not bad, there are some improvements to be made...

Document the intent

- The documentation is basically stating what we can read ourselves in code, but doesn't explain *why* we are doing what we are doing
- Documenting the intent would be better:

```
//check if we are able to attack
if (health > 0 && hasWeapon && stunnedCounter < 0) {
        //...
}
```

- Better!

Self documenting code

- Whenever possible, see whether your code can be self documenting
- Compare this with the previous slide:

```
bool isAbleToAttack = health > 0 && hasWeapon && stunnedCounter < 0;
if (isAbleToAttack) {
        //...
}
```

- Better! (but everyone is entitled to their own opinion ☺)