

Common MMP₁ Problems

Schwer lesbarer Code

- Gut geschriebener Code braucht kaum Kommentare!
- Sie selbst verstehen Ihren Code vermutlich, wenn Sie grade stark am MMP₁ arbeiten.
- Aber verstehen Sie ihn in 2 Monaten auch noch? ;-)

Schwer lesbarer Code

- Vermeiden Sie Magic Values
- Verwenden Sie sprechende Variablennamen und Enums
- Bemühen Sie sich um eine konsistente Klammersetzung

Zu lange / schlecht lesbare Funktionen

- Richtwert – Funktion sollte auf den Screen passen
- Das bedeutet, ca. 50 Zeilen
- Nicht zu viele Verschachtelungen von Schleifen & If / Switch – Statements
- Haben Sie keine Angst davor, aus einer Funktion heraus weitere Funktionen aufzurufen!

Public Member Variablen

- Sind kritisch.
- Verwenden Sie, wenn schon, Properties.
 - public get, aber private set!

Eine Klasse sollte Logik in Form von Methoden zur Verfügung stellen. Es ist schlechter Stil, wenn die Member Variablen einfach von außen überschrieben werden.

Public Member Variablen

```
public class Game {  
    void Update() {  
        foreach (Bullet bullet in Bullets) {  
            if (bullet.Intersects(player)) {  
                player.Health-=1;  
                player.Attackable=false;  
                player.TimeUntilAttack=0.5f;  
                bullet.IsVisible = false;  
                GotHit.Play(0.3f, 0.0f, 0.0f);  
            }  
        }  
    }  
}
```

Vererbung – Machen Sie das NICHT!

```
public class Enemy {  
  
    public virtual void Draw(...) {  
        if (this is Boss) {  
            // do sth  
        }  
        else {  
            // do sth else  
        }  
    }  
}  
  
public class Boss : Enemy {  
  
}
```

Wie gehe ich am besten mit verschiedenen Game States um?

- Ihr Spiel kann sich in unterschiedlichen Zuständen befinden, z.B.
 - Intro Screen wird gezeigt
 - Highscore
 - Reguläres Spiel läuft
- Je nachdem, müssen Sie unterschiedliche Dinge darstellen -> Wie geht das am Besten?
- <https://blogs.msdn.microsoft.com/etayrien/2006/12/12/basic-game-engine-structure/>