

# TetPMC

## A1: Pflichtenheft:

### Ziel:

Im Sinne dieser Projektarbeit soll ein funktionsfähiger Klon einer alten Tetris-Variante entstehen (GameBoy-Pocket, NES - Variante o.ä.).

/F01/ Framework zum Exportieren eines Downscale-Spiels erstellen

/F02/ Spiellogik und Grid erstellen

/F03/ Sound, Animation und gute Grafik

/F04/ Menüs, Sauberkeit und Standalone

/F05/ Speicher und Variabilität

### Umsetzung:

Hauptsächlich wird in Greenfoot wird unter Nutzung der eigenen Klassen entwickelt. Zeitweise wird auch Visual Studio Code genutzt.

Zur Erzeugung von Sicherheitsredundanz wird GitHub verwendet.

Externe Bibliotheken wie libgdx werden aufgrund des engeren Zeitrahmens und der ungetesteten Kompatibilität nicht verwendet.

Dies birgt weiterhin den Vorteil, dass alles minimalistisch und einfach gehalten ist. Das KISS-Prinzip soll ebenso Ziel des Spiels sein (aber keine 4kb-Begrenzung). Die ersten beiden Ziele sind essentiell, das Dritte und Vierte möglich, das Fünfte Overkill.

### /F01/

Zuerst muss Dokumentation gelesen werden, um herauszufinden was Greenfoot alles kann. Danach wird die Arbeitsumgebung mit allen zugehörigen Dateien (+UML-Diagramme) erstellt. Hierzu gehört auch die Klassenmodellierung.

### /F01-1/

Nun werden alle Dateien für ein Lauffähiges Programm, das nichts kann erstellt (wie z.B. eine Welt, Renderer etc.).

## /F02/

Da das Spiel schön aussehen soll, aber nicht nur die Pixelgröße von einem 'Block' haben soll, muss ein Block mehr als einen Pixel einnehmen können. Hierfür einen guten und leistungsfähigen Renderer zu erstellen soll Ziel dieser Etappe sein.

### /F02-1/

Im Spiel sollen einzelne Reihen (=Subblöcke) gelöscht werden und Blöcke miteinander kollidieren. Das muss der Renderer ebenso schaffen.

### /F02-2/

Verschiedene Umgebungsvariablen wie die Laufzeit einer Runde, die Punktzahl und ein Highscore sollen für erweiterte Funktionalität sorgen. Dafür muss man diese erstmal lokal speichern können.

## /F03/

Der Grundstein des Spieles ist hier schon gelegt. Nun kann man sich Gedanken über weitere Ausgestaltung machen.

### /F03-1/

Zuerst kann die Grafik mit einem Hintergrund und schönen Steinen verbessert werden.

### /F03-2/

Tetris verfügt in vielen Varianten auch über Animationen bei z.B. dem Aufprallen eines Steines oder des Löschens einer Reihe. Diese gilt es (sofern mit Greenfoot möglich) zu implementieren.

### /F03-3/

Zuletzt kann man noch Soundeffekte hinzufügen. Hierfür wäre eine robuste und fehlerresistente eigene Abspielklasse interessant.

### /F03-4/

Spätestens hier sollte dringend der Code aufgeräumt und verschönert werden sowie ein Großteil der Dokumentation entstehen.

## /F04/

Ab diesem Unterpunkt wird nichts mehr direkt am Spiel gemacht. Untermenüs für z.B. Einstellungen und ein Splash sind eine gute Addition für das Spiel. Außerdem soll es exportierfähig werden.

/F05/

Mit diesem Ziel sollen verschiedene weiterführende Dinge implementiert werden:

/F05-1/

Auch nach Schließen des Programms sollen Kerndaten wie Highscores oder Einstellungen gespeichert werden können (->JSON)

/F05-2/

Weitere Einstellungen wie z.B. das Ändern von Blockskins oder der Auflösung sollen hinzugefügt werden.

/F05-3/

Hier erfolgt das letzte Aufräumen und Exportieren.

So kann das Spiel aussehen:

