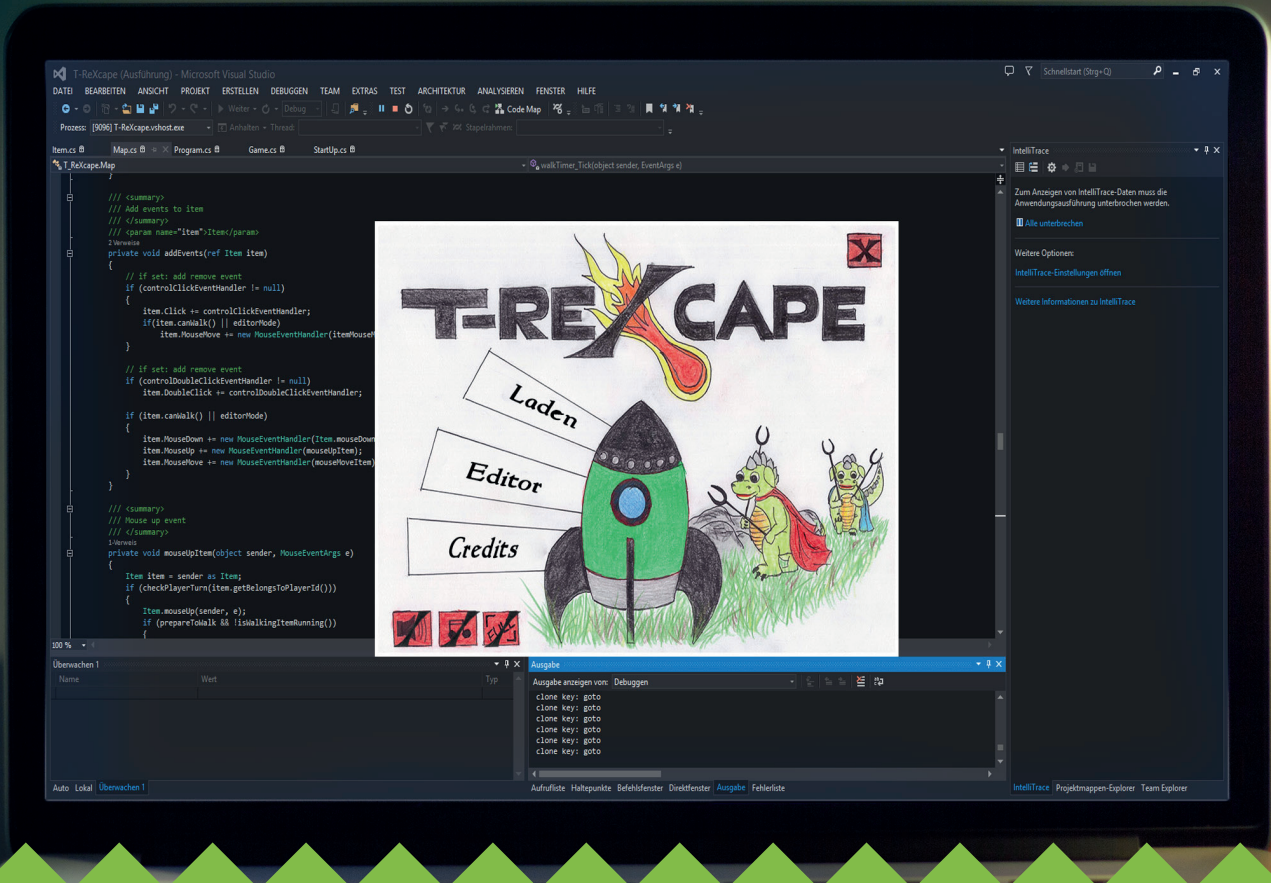


# DOKUMENTATION



## T-ReXcape

Eine Projektarbeit von:

Andreas Heckel  
Alena Will

Mischa Helfenstein  
Tim Weis

Berufsbildende Schule | Kaiserslautern  
Projektabgabe: 09.07.2015

## Inhaltsverzeichnis

1. Beschreibung der Projektarbeit
  - a) Teilziele und Teilaufgaben
  - b) Vorgaben
2. Ressourcen- und Ablaufplanung
  - a) Team
  - b) Zeitliche Planung
  - c) Projektphasen
  - d) Aufgabenverteilung
3. Tätigkeitsangaben im Zeitraster
  - a) Erledigung der Aufgabe
  - b) Vergleich der Zeiten
4. Durchführung
5. Qualitätssicherung
  - a) Testen
  - b) Soll/Ist - Vergleich
6. Reflexion der Vorgehensweise
7. Fremde Quellen

## Vorwort

Bevor wir mit der eigentlichen Dokumentation beginnen, möchten wir ein paar Worte bezüglich dem Grundgedanken hinter dem Projektes erwähnen.

Denn wir haben uns dazu entschlossen unser Projekt als Open-Source zu deklarieren und arbeiten dementsprechend von Anfang an öffentlich.

### **Warum?**

Wir sind der Meinung, dass damit anderen Personen geholfen werden kann und zugleich ein Nutzen für uns entsteht.

Einerseits kann die Qualität des Projektes gesteigert werden.

Dies geschieht entweder über Kommentare, Issues (Bug-Reports) oder eben direkten Verbesserungen am Quellcode welche über einen Pull-Request von aller Welt eingereicht werden können.

Andererseits können Personen daraus ihr eigenes Spiel erstellen oder eben anhand des Quellcodes eigene Probleme lösen. Auch Menschen, die anfangen sich mit C# und Windows Forms zu befassen können grundlegende Konzepte näher an einem Praxisobjekt kennenlernen.

Zusätzlich wird durch die Versionsverwaltung ein paralleles Arbeiten am Projekt ermöglicht und somit die Effizienz gesteigert.

## 1) Beschreibung der Projektarbeit

### 1a) Teilziele und Teilaufgaben

Das konkrete Ziel des Projektes „T-ReXcape“ ist die Entwicklung eines rundenbasierten Mehrspielespiels auf Basis von C# und Windows Forms. Als Besonderheit soll das Projekt vollständig über [auf Papier] handgezeichnete Grafiken verfügen. Zudem verfügt das Spiel über einen speziell für dieses Spiel erstellten Soundtrack. Um das Projekt in einem angemessenen Umfang fertigstellen zu können, wird es in mehrere Teilziele und Aufgaben unterteilt. Diese werden folgend genannt und näher erläutert:

- **Planung**

Bevor mit dem Projekt begonnen werden kann, muss sich innerhalb des Teams auf eine Idee geeinigt werden. Des Weiteren müssen innerhalb dieser Phase / Aufgabe auch die einzelnen Teilaufgaben unter den Teammitgliedern verteilt werden. Auch die Konzeption des Spiels (Klassenstruktur / allg. Aufbau) spielt hierbei eine wichtige Rolle.

- **Allgemeine Klassenstruktur**

Innerhalb dieses Vorgangs werden die allgemein benötigten Klassen erstellt. Dies zeigt den Vorteil der objektorientierten Programmierung, da diese Klassen in allen Teilen des Projektes benutzt werden können. So wird beispielsweise eine Map-Klasse erzeugt, die die Items des sogenannten Map-Panels verwaltet. Auch die Items erhalten ihre eigene Klasse um deren Größe, Funktionen, Eigenschaften zu verwalten. Die Items werden in der ItemCollection gesammelt um somit immer einen Überblick über alle verfügbaren Objekte zu haben. Innerhalb der Config-Klasse werden die Items initialisiert, aber auch wichtige Informationen verwaltet, wie z.B. die Filter für die FileDialogs. Das Speichern und Laden der Map wurde auch in eine eigene Klasse ausgelagert. Dies ermöglicht, dass der Editor sowie das Spiel selbst auf die Methoden der Klasse zugreifen können. Ein weiteres Feature welches zur allgemeinen Klassenstruktur gehört ist die Soundklasse. Diese ermöglicht das Abspielen von Soundeffekten, sowie das Starten und Stoppen des spielweiten Soundtracks. Somit zeigt sich, dass ein Großteil des Projektes auf der Klassenstruktur aufbaut.

- **Startbildschirm**

Der Startbildschirm bietet dem Nutzer eine einfache Möglichkeit die

einzelnen Teile des Projektes einzusehen und zu benutzen. Darum hat man hier direkten Zugriff auf den Editor und das Spiel. Wichtig zu nennen ist auch, dass die bereits geladenen Karten hier in einem kleinen DropDown Menü angezeigt werden. Weiteren Zugriff gibt es auf die Credits sowie die Checkboxen für Fullscreen, Soundtrack und Soundeffekten.

- **Editor**

Der Editor bildet ein wichtiges Element des Projektes. Mit diesem können die Nutzer ihre eigenen Karten erstellen aber auch das Team Karten zum Testen und dem Release generieren. Im Editor sollte darauf geachtet werden, dass dieser sehr einfach zu benutzen ist und so viele Fehler wie möglich abfängt. Das Platzieren von Items soll über zwei Wege möglich sein. Zum einen soll dies über ein Kontextmenü geschehen, aber auch über ein eigenes Panel, welches alle verfügbaren Items anzeigt. Die Items können nach dem Platzieren über Drag'n'Drop verschoben und an einer andern Stelle platziert werden. Mit gedrückter Maustaste und dem „Ziehen“ in eine Richtung kann die Ausrichtung des Items geändert werden. Mit einem Rechtsklick können bereits platzierte Objekte wieder gelöscht werden. In einem Info-Tab kann die Spielfeldgröße dynamisch angepasst werden. Hier kann auch angesehen werden, welche Items (Objekte) bereits platziert wurden und wie viele gesamt platziert werden können.

- **Spiel / Spiellogik**

Das Spiel ist wie bereits oben genannt rundenbasiert, es gibt Hindernisse und nur ein Ziel für die beiden Spieler. Hierfür muss die Spiellogik genau an das Spiel angepasst werden. So wird zum Beispiel bei einem Loch der Spieler an die Position zurückgesetzt, an der er den Zug begonnen hat. Das Starten der Figur wurde mit Rücksprache des Lehrers leicht angepasst. So zieht der Spieler seine Figur mit gedrückter Maustaste in die Richtung in die er laufen möchte und lässt zum Loslaufen die Maustaste los.

Außerdem muss darauf geachtet werden, dass die Spielfiguren nicht aus der Karte herauslaufen können. Ein weiteres Feature ist das Aufsammeln von kleineren Items, die zu ganzen Items zusammengefügt werden. So kann ein Spieler zum Beispiel seinen Gegner eine Mauer in den Weg platzieren, sodass dieser einen Umweg gehen muss. Durch die Möglichkeit des Vollbildmodus wird die Größe der Karte und der sog.

„Blocksize“ (Größe eines Rasterelements) dynamisch angepasst. Dies erfordert auch mehrere Berechnungen/Implementierungen, die in die Zeit eingeplant werden müssen. Auch im Spiel gibt es die Möglichkeit die Soundeffekte und den Soundtrack einzeln an- und auszuschalten.

- **Testen**

Die Teilaufgabe des Testens muss differenziert betrachtet werden. In der Planungsphase entschloss sich das Team dauerhaft zu testen. Dies hat den Vorteil, dass eine Person eine Funktion testet, deren Implementierung er nicht kennt und somit einen neutralen Blick darauf hat. Außerdem ist man während der Entwicklung meist dazu geneigt die einzubauende Funktion auf möglichst allen wegen zu testen. Jedoch wurde das Projekt während des Projektabschlusses noch einmal komplett von allen Teammitgliedern getestet um möglichst alle Fehlerquellen und Bugs zu finden.

- **Grafik**

Da das Spiel auf handgezeichneten Grafiken basiert, mussten diese natürlich erstellt werden. Zuerst werden diese skizziert und danach koloriert. Grafiken wurden für die Items, Spieler, aber auch für den Startbildschirm und das Pausenmenü erstellt. Zusätzlich zum Zeichnen mussten die Bilder freigestellt und im Fall der Spielerfiguren auch animiert werden.

- **Anpassungen / Bugfixing**

Zu dieser Teilaufgabe gehören allgemeine, kleine Änderungen am Code und Anpassungen von verschiedenen Code-Teilen. Jedoch müssen auch die gefundenen Fehler behoben werden. Diese Fehler können einfach, aber auch teilweise sehr schwer zu finden sein. Somit ist der Aufwand dieser Aufgabe nicht zu unterschätzen.

- **Karten erstellen**

Wie der Name der Aufgabe schon sagt, müssen für die Abgabe mehrere Karten zur Verfügung gestellt werden. Dies geschieht mit dem projekteigenen Level-Editor. Zu beachten gilt jedoch hierbei, dass die Level fair gestaltet werden.

- **Dokumentation**

Abschließend muss für das Projekt, die jetzt vorliegende, Dokumentation erstellt werden. Hierin wird auf die Planung, den Zeitplan sowie die



Umsetzung eingegangen. Auch Probleme und Verbesserungsvorschläge werden festgehalten. Dies ermöglicht auch Personen, die keinen Einblick in die Entwicklung hatten eine Vorstellung über das Projekt zu gewinnen. Auch hinzuzählen das Erstellen des geforderten Projekttagebuchs sowie der Kundendokumentation.

### **1b) Vorgaben**

Da es sich bei dem Projekt um eine schulische Leistungsermittlung hält, gibt es einige Vorgaben:

- Programmiersprache C# / Windows Forms
- Rundenbasierender Mehrspielmodus
- Maussteuerung
- Spieler kann nur in eine Richtung laufen (bis zu einem Hindernis oder anderen besonderen Feldern)
- Rasterbasiertes Spielfeld
- Hindernisse auf dem Spielfeld (u.a. „Umlenkungen“)
- Level Editor
- Das Spiel ist gewaltfrei und einem Schulprojekt angemessen
- Zeitrahmen: 7 Wochen (21.05.15 – 09.07.15)

## **2) Ressourcen und Ablaufplanung**

### **2a) Team**

Das Team besteht aus vier Mitgliedern, welche jeweils Auszubildende Fachinformatiker sind. Drei der vier Personen sind in der Fachrichtung Anwendungsentwicklung tätig. Zu nennen sind hier Alena Will, Andreas Heckel und Mischa Helfenstein. Tim Weis, das vierte Teammitglied, ist ein Auszubildender der Fachrichtung Systemintegration.

## 2b) Zeitliche Planung

| Teilaufgabe                | Veranschlagte Zeit (Stunden) |
|----------------------------|------------------------------|
| Planung                    | 2                            |
| Allgemeine Klassenstruktur | 15                           |
| Startbildschirm            | 4                            |
| Editor                     | 10                           |
| Spiel / Spiellogik         | 12                           |
| Testen*                    | 4                            |
| Grafik                     | 10                           |
| Anpassungen / Bugfixing    | 6                            |
| Karten erstellen           | 2                            |
| Dokumentation              | 8                            |
| Gesamt                     | 73                           |

\* Die veranschlagte Zeit bezieht sich nur auf das Testen zum Projektabschluss

## 2c) Projektphasen

- **Planung**

Der Start des Projektes und somit auch die zeitlich zuerst erledigte Aufgabe liegt in der Planung. Dies wird somit direkt in der ersten Woche, in den Schulstunden erledigt.

- **Umsetzung**

Die Umsetzung umfasst den Zeitraum zwischen Planung und Projektabschluss, also ca. den Umfang von der 1. Woche bis zum Ende der 6. Woche. Zu den Aufgaben der Umsetzung gehört das Erstellen der Klassenstruktur gefolgt vom Erstellen der einzelnen Forms und der Spielelogik. Währenddessen werden auch die benötigten Grafiken erstellt und animiert.

- **Projektabschluss**

Die letzte eigentliche Phase besteht im Projektabschluss. Hierzu zählen unter anderem das endgültige und abschließende Testen, das Anpassen sowie Fixen von gefundenen Bugs. Dies geschieht am Ende der 6. Projektwoche und wird bis zum Ende des Projektes weitergeführt um ein möglichst gutes Ergebnis zu erreichen.

- **Dokumentation**

Neben den bisher genannten Aufgaben und Phasen läuft die Dokumentation mehr oder weniger eigenständig dauerhaft nebenher. Vor allem das Erstellen des Projektstagebuches wird in regelmäßigen



Abständen erledigt. Die Vollständige Dokumentation sowie die Kundendokumentation werden in der letzten Woche erledigt, um somit alle Features, Probleme, Zeiten eintragen zu können.

## 2d) Aufgabenverteilung

|                         |  |
|-------------------------|--|
| <b>Planung</b>          | Alena Will, Andreas Heckel, Mischa Helfenstein, Tim Weis   |
| <b>Umsetzung</b>        | <i>Programmierung:</i> Andreas Heckel, Mischa Helfenstein, (Alena Will), (Tim Weis)<br><i>Grafik:</i> Alena Will, (Andreas Heckel), (Mischa Helfenstein) |
| <b>Projektabschluss</b> | <i>Testen:</i> Alena Will, Andreas Heckel, Mischa Helfenstein, Tim Weis<br><i>Bugfixing:</i> Andreas Heckel  |
| <b>Dokumentation</b>    | <i>Dokumentation:</i> Mischa Helfenstein<br><i>Kundendoku:</i> Mischa Helfenstein<br><i>Projekttagebuch:</i> Mischa Helfenstein, Tim Weis                |

\* () bedeutet, dass dieses Teammitglied hierbei kleinere Aufgaben übernimmt

## 3) Tätigkeitsangaben im Zeitraster

### 3a) Erledigung der Aufgaben

In den ersten Stunden wurde das Projekt von allen Mitgliedern geplant und die Aufgaben an die einzelnen Personen verteilt. Danach haben Andreas Heckel und Mischa Helfenstein damit begonnen die allgemeine Klassenstruktur aufzubauen. Kleinere Klassen wurden hierbei von Alena Will und Tim Weis erstellt. Während der zweiten, dritten und vierten Woche wurde von den beiden Hauptprogrammierern die Forms (Start / Editor / Spiel) erstellt. Diese wurden meist mit Platzhalter Grafiken erstellt. Dies hatte den Vorteil, dass Alena Will die Grafiken mit höherem Zeitaufwand erstellen konnte und somit eine bessere Qualität abgeliefert wurde. Die Grafiken wurden danach von Andreas und Mischa freigestellt sowie animiert. Woche Nummer fünf und sechs bestanden hauptsächlich aus dem Einfügen der Grafiken sowie dem Entwickeln der Spiellogik und kleineren Anpassungen. Die letzten beiden Wochen bestanden hauptsächlich aus dem Hinzufügen kleinerer Features, Fixen von Bugs und Behebung von Konsistenzfehlern. Zudem wurden in der letzte Woche von Mischa Helfenstein die Dokumentation sowie eine kundengerechte Dokumentation

angefertigt. Tim Weis war während des gesamten Zeitraums für das Erstellen des geforderten Projekttaggebuchs zuständig. Abschließend wurde das Projekttaggebuch von Mischa Helfenstein noch einmal überarbeitet um alle geforderten Punkte einfließen zu lassen.

### 3b) Vergleich der Zeiten

| Teilaufgabe                       | Veranschlagt | Benötigt  |
|-----------------------------------|--------------|-----------|
| <b>Planung</b>                    | 2            | 2         |
| <b>Allgemeine Klassenstruktur</b> | 15           | 17        |
| <b>Startbildschirm</b>            | 4            | 3         |
| <b>Editor</b>                     | 10           | 10        |
| <b>Spiel / Spiellogik</b>         | 12           | 14        |
| <b>Testen*</b>                    | 4            | 3         |
| <b>Grafik</b>                     | 10           | 9         |
| <b>Anpassungen / Bugfixing</b>    | 6            | 6         |
| <b>Karten erstellen</b>           | 2            | 1         |
| <b>Dokumentation</b>              | 8            | 10        |
| <b>Gesamt</b>                     | <b>73</b>    | <b>75</b> |

### 4) Durchführung

Die Vorgehensweise ist recht einfach zu beschreiben. Zu aller Anfang wurden die Aufgaben verteilt, sodass jeder wusste, was zu seinem Aufgabengebiet gehört. So wurde der Großteil der Entwicklung der Klassen, Forms und der Spiellogik von Andreas Heckel und Mischa Helfenstein übernommen. Kleinere Programmierarbeiten wurden den restlichen beiden Teammitgliedern zugeteilt. Die Absprache zwischen den Hauptprogrammierern lief reibungslos über gut strukturierte Commits und Issues ab. Aber auch mindestens zweimal in der Woche wurde der aktuelle Stand miteinander via Voicechat besprochen. Durch die Commit-Nachrichten konnten die Entwickler jeweils sehen, was der andere bereits gemacht hat. Wichtige Aufgaben und Probleme wurden mittels Issues (Github) notiert, diskutiert und zugeteilt. Die Commit-Nachrichten konnten auch sehr gut von Tim Weis als Grundlage für das Projekttaggebuch benutzt werden. Während der Entwicklung stellten sich zu den geplanten Grafiken noch weitere heraus, wodurch Alena Will mitgeteilt werden konnte, welche Grafiken zusätzlich benötigt werden. Dies zeigt beispielsweise schon eine kleine Anpassung von Teilaufgaben, da Dinge benötigt wurden, die nicht bedacht oder eingeplant wurden. In den wöchentlichen Besprechungen wurde der aktuelle Stand mit

dem gewünschten Ziel verglichen. Hierbei konnte jedes Teammitglied seine Meinung und Verbesserungswünsche äußern. Somit zeigt sich eine sehr gute, dynamische aber auch klare Kommunikationsstruktur innerhalb des Teams. Bei Problemen mit bestimmten Implementierungen sprachen sich die beiden Hauptentwickler untereinander ab. Abweichungen gab es neben den zusätzlichen Grafiken nur bei gestrichenen Items und Objekten, welche nicht in den geplanten Zeitrahmen passten, jedoch auch bei der Planung bereits als optional deklariert wurden.

## 5) Qualitätssicherung

### 5a) Testen

Der Testvorgang muss, wie bereits oben beschrieben, unter mehreren Gesichtspunkten betrachtet werden. Einerseits wird während der Entwicklung vom jeweiligen Programmierer direkt getestet um die Funktionalität der Implementierung zu prüfen. Hierbei können die anderen Teammitglieder sehr hilfreich sein, da diese die genaue Implementierung nicht kennen und somit sog. „Leichtsinnfehler“ erkennen und melden können. Außerdem besteht durch den Open-Source Gedanken die Möglichkeit, dass sich jede Person der Welt den Programmcode laden kann und Fehler über das Issue-System melden und genaue Fehlerberichte hinzufügen kann. Diese Möglichkeit besteht auch über den Abgabezeitraum hinaus und sichert somit langfristig die Qualität des Projektes. Als weitere Punkt muss das abschließende Testen des Projektes betrachtet werden. Hierbei wird im Team, zusammen, Stück für Stück das Spiel getestet und somit Fehler erkannt aber auch Inkonsistenzen festgestellt.

### 5b) Soll/Ist – Vergleich

Grundlegend wurden alle vorgesehenen und wichtigen Ziele erreicht. Somit war das Projekt ein voller Erfolg. Jedoch waren noch einige Items mehr vorgesehen, welche jedoch zeitlich nicht in den Rahmen passten. Auch könnten einige Klassen weiter untergliedert werden, aber dies sind nur Verbesserungen auf hohem Niveau. Daraus zeigt sich, dass die Planung funktioniert hat und das Ziel erreicht wurde.

## 6) Reflexion der Vorgehensweise

- **Positive Aspekte der Vorgehensweise**

Ein großer positiver Aspekt der Vorgehensweise war, dass sich die Mitglieder bei Design- und Spielfragen schnell auf eine Einigung/Lösung festlegen konnten. Dies erleichtert und erspart vor allem viel Kommunikation, welche nicht benötigt wäre. Hierin liegt auch ein weiterer positiver Punkt, denn die Kommunikation außerhalb der Schule funktionierte besser als zuerst erwartet wurde. Auch die Verteilung der Aufgaben erschien passend. Zusätzlich sollte auch der Open Source Gedanke als positiver Aspekt angesehen werden. Denn somit können andere Personen an dem Projekt mithelfen oder ihre eigene Kreation daraus entwickeln. Auch fördert dies den Teamgedanken hinter einem Projekt ohne kommerziellen Hintergrund.

- **Negative Aspekte der Vorgehensweise**

Negativ muss angemerkt werden, dass die Teammitglieder teilweise Aufgaben erledigen mussten, die eigentlich nicht zu ihren Aufgabenbereichen gehörten. Dies war vor allem der Fall, wenn man mit dem Ergebnis einer Aufgabe nicht zufrieden war und diese somit, mit Teamrücksprache, verbessern musste. Somit wurde teilweise mehr Zeit benötigt als ursprünglich gedacht. Des Weiteren waren die Ambitionen für das Projekt etwas zu hoch gesteckt, wodurch zwar die Qualität steigt jedoch auch der Zeitaufwand für manche Teammitglieder sehr gestiegen ist. Auch wurde die geplante Zeit, wenn auch nur leicht, überschritten.

- **Fazit**

Zusammenfassend zeigt sich, dass das Projekt trotz einiger kleiner Probleme ein Erfolg war. Die Probleme konnten meist recht zügig gelöst werden oder stellten sich als nicht schwerwiegend heraus.

## 7) Fremde Quellen

| Benutzer Inhalt             | Link / Person   |
|-----------------------------|---|
| IniFile.cs<br>(abgewandelt) | <a href="http://stackoverflow.com/questions/217902/reading-writing-an-ini-file">http://stackoverflow.com/questions/217902/reading-writing-an-ini-file</a> |
| Soundtrack                  | Pascal Richter  |