

Decks sind nicht mehr in einer LinkedList, weil diese einfaches Einfügen von mehreren Elementen sowie einfaches Iterieren nicht erlauben. Ursprünglich war eine LinkedList angedacht, da sie zwar für Zugriffe auf die Mitte der Liste wesentlich langsamer ist, aber für das Entfernen und Hinzufügen von Elementen am Anfang und Ende der Liste in konstanter Zeit erfolgt, weil sie nicht auf einer Array sondern auf durch Pointer verkettete Listenelementen basiert.

Das Format der Kartendatei blieb etwa beim im Pflichtenheft beschriebenen Format. Die Statuswerte in der einzelnen Disziplinen wurden jedoch nicht in ein gesondertes "stat"-Objekt gepackt sondern sind nun auf einer Ebene mit den anderen Eigenschaften um das Laden der Datei durch Model-Binding zu vereinfachen.

Die Benutzeroberfläche des Spiels entspricht im groben der Planungsskizze, die meisten Änderungen wurden gemacht, weil GTK# keine einfache Möglichkeit bietet um Bilder im Hintergrund von anderen Elementen anzuzeigen.

Der Spielablauf läuft im groben nach dem im Programmablaufplan beschriebenen Schema ab. Da der Programmablaufplan jedoch keine Besonderheiten wegen CPU-Spielern etc. vorsieht weicht er dennoch stellenweise stärker vom Resultat ab.

Die Zeit- und Ressourcenplanung wurde relativ eingehalten. Wir haben keine Zeiträume, sondern nur eine grobe Reihenfolge angegeben. Da Tim auf der Arbeit wegen eines Projekts die erste Zeit vereinnahmt war. Der Großteil des Programms wurde aufgrund der vorher genannten Dinge, sowie dadurch dass André gesundheitliche Probleme hatte und sogar eine Woche krank geschrieben werden musste, erst gegen Ende des Zeitraums geschrieben. Die Aufteilung der Aufgaben ist auch nicht ganz wie ursprünglich geplant abgelaufen. Jedoch haben wir die Anfänglich verlorene Zeit durch Überstunden wieder aufholen können, um Ihnen ein hervorragendes Produkt zu liefern!

Auch war ursprünglich Windows Forms angedacht. Jedoch mussten wir das, da Tim auf Mac OSX arbeitet, lassen. Stattdessen haben wir Mono mit GTK# verwendet, welches wesentlich schwerer zu handhaben ist und vor allem auf Windows Probleme hat ans Laufen zu kommen.

Das heißt die **Mono-Runtime** sowie vielleicht auch **GTK#** sind nun Vorraussetzungen für das Projekt. Aufgrund eines uns selbst durch die Macht von Google nicht lösbaren Fehlers läuft das Programm mittlerweile nicht auf Andrés Laptop. Auf seinem Heimcomputer läuft es jedoch problemlos. André weigert sich fortan mit Mac-Programmierern zusammen zu arbeiten. Tim überlegt sich auch nun mindestens zweimal, ob er das nächste größere C# Projekt nicht doch lieber in einer Windows Virtualisierung umsetzt. & Hoffte dass André auch in Zukunft nochmal mit ihm zusammenarbeiten wird.

Die Zusammenarbeit, Koordinierung und Kommunikation verlief besser als in früheren Projekten. Besonders die gemeinsame Vorplanung, wie auch zwischenzeitliches per-programming (in der Schule, sowie online mit screensharing), waren besonders gehaltvoll und trieben das Projekt gut voran.

Bekannte mögliche Probleme sind:

- Falsche JSON-Dateien können zum Absturz führen
- Die Deckgrößenauswahl stimmt nur für eine Gesamtanzahl von 32 Karten
- Zu wenige Karten im Deck können zusammen mit der Deckgrößenauswahl zum Absturz führen