Adam Shafei Len Williamson Steffen Pegenau

1. Übungszettel in Software Engineering

Teilaufgaben und Schätzungen

Aufgabe	Begründung/Kommentar	Schätzung [h]
Klasse GUIConsole	Starten, Einstellen, Spielen	**SUMME: 6,25**
- Eingabe - - Spieleranzahl - - Spielfelddimensionen - - Spielernamen - - Zug (Spieler x setzt Wand y)	Einarbeitung in Eingabemethoden auf der Konsole Konsole, gemäß Erfahrung geringe Komplexität, schnell umsetzbar	0,75
- Darstellung Spielfeld	Darstellung soll Datenstruktur der Karte interpretieren, sodass eine Anpassung der Darstellung möglich ist, ohne die Datenstruktur grundsätzlich zu ändern. Darstellung muss sich bei großen Feldern anpassen.	1
- Anzeige - - Punkte - - Gewinner - - Zugzahl	Anzeige zum richtigen Zeitpunkt, sonst keine besondere Komplexität. Inhalte werden über Methoden der jeweiligen Klassen gut abrufbar sein.	0,5
- Tests schreiben	Schwierig, da möglichst viele Fälle (umfangreich) abgedeckt sein müssen und Übung fehlt. Erfordert viel Kreativität.	2,5
- Fehlerkorrekturen & Optimierungen	Etwas geringere Komplexität als die Tests. Nachdem die Tests laufen, sollten nicht mehr viele Korrekturen/Optimierungen nötig sein	1,5

Klasse DotsNBoxesEngine	Einhaltung der Regeln, Verwaltung des Spielstandes, Punkteberechnung	**SUMME: 7,5**
- Datenstruktur Karte (2D-Array) mit Initialisierung	Grundsätzliche geringe Komplexität, aber durch Zweidimensionalität wenig intuitiv	1
- Zug durchführen (boolean: true, wenn Zug erfolgreich) - - Ist der Spieler an der Reihe? - - Ist Zug gültig? - - Ist Käsekästchen voll? - - Ist das Spiel beendet? - - Wer ist als nächstes an der Reihe?	Höchste Komplexität, da hier die komplette Spiellogik liegt (die zuerst verstanden sein muss) und viele Hilfsmethoden geschrieben werden müssen.	2
- Punkteverwaltung	Geringe Komplexität, da es sich gut über getter und setter der Klasse Player umsetzen lässt	0,5
- Tests schreiben	Wie in Klasse GUIConsole	2,5
- Fehlerkorrekturen	Wie in Klasse GUIConsole	1,5

Klasse Player	Verwaltung Name, Punktestand	**SUMME: 0,5**
- Name setzen - Name holen - Punkte setzen - Punkte holen - Punkte erhöhen um Wert x - Tests schreiben - Fehlerkorrekturen	Im Grunde nur getter- und setter-Methoden. Daher sehr geringe Komplexität	
GESAMTZEIT		**14,25**