TESTKONZEPT UND TESTPROTOKOLL

Lernende Informatiker 11. März 2020

Dominik Arnet

Pilatus Flugzeugwerke AG

Royal Game of Ur

Inhalt

1 Bericht

Lernende Informatiker	23. Dezember 2020
Projektauftrag_Royal_game_of_Ur.docx	

Projektname: **Online Variante für Royal Game of Ur**
Projektphase: ☐ **Vorstudie** ☐ **Konzept** ☒ **Realisierung** ☐ **Einführung**
Version / Status: **0.1 / Draft**
Verfasser: **Nevin Helfenstein / Lino Meyer / Lino Bucher**
Verteiler: **Nevin Helfenstein / Lino Meyer / Lino Bucher**

Ausgangslage

Das Royal Game of Ur ist das älteste Spiel, welches auf diesem Planeten existiert. Wir haben uns bereits vor dem Projekt sehr für das Spiel interessiert und waren ein wenig enttäuscht das es keine gute online Variante davon gibt. Als wir die Möglichkeit bekamen, unser Thema für die Projektarbeit im Modul M150 selber auszuwählen, haben wir uns entschlossen eine gute online Version des Spiels zu erstellen.

Ziele

Folgende Zielsetzungen sollen in diesem Projekt erreicht werden:

- Die Applikation soll nicht nur schön aussehen, sondern auch einfach in der Benützung sein
- Das Spiel kann nicht nur auf dem Computer, sondern auch auf Tablets spielbar
- Die Applikation ist mit den originalen Regeln des Royal Game of Ur implementiert
- Der User der Applikation kann gegen ein Bot spielen
- Die Applikation ist zum definierten Endtermin lauffähig, getestet und dokumentiert

Aufgaben

- Initialisierung des Projektes
- Planen der Umsetzung
- Anforderungen aufnehmen
- Anforderungsspezifikation erstellen
- Applikation realisieren und dokumentieren
- Applikation testen
- Applikation einführen
- Projektabschluss

Rahmenbedingungen (Gestaltungsbereich/betroffene Organisationseinheiten, Restriktion)

- Die vordefinierten Entwicklungsrichtlinien werden angewendet und eingehalten
- Angular Standards werden eingehalten

Projektorganisation		
Rolle	**Personentage (PT)**	Visum Vorgesetzter
Auftraggeber		
Imboden Adrian	0.5 - 1	
Projektleiter		
Nevin Helfenstein	10	
Projektmitarbeiter		
Nevin Helfenstein, Lino Meyer, Lino Bucher	10	

Umsetzung

Nachdem wir das Thema und die Ziele der Applikation definiert haben, mussten wir aussuchen mit welcher Programmiersprache wir das Projekt machen wollten. Hier haben wir uns für Angular mit Node JS entschieden. Für das Projekt verwenden wir VSCode mit den Standard Einstellungen, welche bei der Installation mitgeliefert werden. Anschliessend haben wir uns noch dafür entschieden, die Applikation funktional zu entwickeln.

Zuerst haben wir dann das Backend umgesetzt und getestet. So konnten wir anschliessend uns voll und ganz auf das Frontend konzentrieren ohne uns um das Backend zu kümmern.

Ergebnis

Unser Ergebnis ist grundsätzlich genau das was wir uns vorgestellt haben. Wir konnten eine schöne und Userfreundliche Applikation erstellen, welche wir auch ohne Probleme wirklich online schalten könnten. Während des Projektes mussten wir jedoch unsere Zeit ein wenig besser einteilen und konnten deswegen nicht alles erledigen was wir am Anfang implementieren wollten.
Zum Beispiel konnten wir nur einen simplen Bot implementieren. Hier wollten wir uns noch mehr Zeit nehmen, welche wir leider nicht hatten. Weiter konnten wir die Funktion mit dem Multiplayer auch nicht implementieren. Hier wollten wir zuerst die Möglichkeit bieten mit einem Kollegen über einen Link zusammen zu spielen. Dies konnten wir Zeittechnisch nicht mehr in Angriff nehmen.

Alles in allem sind wir mit unserem Ergebnis sehr zu frieden.

2 Testkonzept

Für die Grösse und Komplexität unseres Projekts haben wir uns entschieden das uns Unittests eine Ausreichende Testabdeckung liefern. Deshalb verwenden wir in unserem Projekt nur Unittests und keine E2E-Tests. Zum Testen verwenden wir die Test-Library Chai in der Programmiersprache Typescript in Visual Studio Code.
Die Tests können im Projektordner mit dem Konsolenbefehl «npm run tests» ausgeführt werden.
Die Konfiguration des von uns verwendeten Code Linter ist im Projekt unter dem File-Namen tslint.json zu finden. Die Codeformatierung wird im CI mithilfe des Code Linter überprüft.
Performance Tests sind für dieses Projekt nicht von Nöten, da die Performance Anforderungen an unsere Applikation minimal sind.

2.1 Akteure

Akteure	Beschreibung
Spieler	Nutzer der Webseite, die unser Spiel spielen wollen

2.2 Testfälle

Testfall-Nr. TC01	
Auslöser	Das Spielbrett soll generiert werden
Voraussetzung	Keine
Funktionsablauf	Die Methode generateStartBoard wird aufgerufen
Erwartetes Ergebnis	Das Spielbrett wird generiert mit den richtigen Feldern an den richtigen Positionen

Testfall-Nr. TC02	
Auslöser	Es soll möglich sein auf ein Feld des Spielbretts zugreifen zu können
Voraussetzung	Es wurde ein Spielbrett generiert
Funktionsablauf	Die Methode nthSquare wird aufgerufen
Erwartetes Ergebnis	Das gesuchte Feld auf dem Spielbrett wird zurückgegeben

Testfall-Nr. TC03	
Auslöser	Es soll möglich sein die vier Würfel zu werfen
Voraussetzung	Keine
Funktionsablauf	Die Methode throwDice wird aufgerufen
Erwartetes Ergebnis	Es wird eine Zahl zwischen 0 und 4 zurückgegeben

Testfall-Nr. TC04	
Auslöser	Es soll möglich sein einen Spielstein auf das Spielbrett zu legen
Voraussetzung	Ein Spielbrett ist vorhanden
Funktionsablauf	Die Funktion placeStoneOnBoard wird aufgerufen
Erwartetes Ergebnis	Es befindet sich ein Stein auf dem Spielbrett

Testfall-Nr. TC05	
Auslöser	Es soll möglich sein einen Stein zu bewegen
Voraussetzung	Ein Spielstand ist vorhanden und es befindet sich ein Stein auf dem Brett
Funktionsablauf	Die Funktion moveStone wird aufgerufen
Erwartetes Ergebnis	Der Stein hat sich auf dem Brett bewegt

Testfall-Nr. TC06	
Auslöser	Es soll möglich sein einen Stein eines anderen Spielers zu schlagen
Voraussetzung	Ein Spielstand ist vorhanden und es befinden sich zwei Steine von verschiedenen Spielern/Farben auf dem Spielbrett
Funktionsablauf	Die Funktion moveStone wird aufgerufen
Erwartetes Ergebnis	Der Stein, der sich bewegt schlägt den anderen Stein, welcher zurück ins Steinlager des geschlagenen Spielers geht

Testfall-Nr. TC07	
Auslöser	Es soll nicht möglich sein einen Stein der eigenen Farbe zu schlagen
Voraussetzung	Ein Spielstand ist vorhanden und es befinden sich zwei Steine von dem gleichen Spieler/Farbe auf dem Spielbrett
Funktionsablauf	Die Funktion validMove wird aufgerufen
Erwartetes Ergebnis	Der Stein hat sich nicht bewegt

Testfall-Nr. TC08	
Auslöser	Es soll nicht möglich sein einen Stein eines anderen Spielers zu schlagen, der sich auf einem Spezialfeld befindet
Voraussetzung	Ein Spielstand ist vorhanden und es befinden sich zwei Steine von verschiedenen Spielern/Farben auf dem Spielbrett. Der Stein der geschlagen werden soll befindet sich auf einem Spezialfeld
Funktionsablauf	Die Funktion validMove wird aufgerufen
Erwartetes Ergebnis	Der Stein hat sich nicht bewegt

Testfall-Nr. TC09	
Auslöser	Ein Stein der auf dem Zielfeld landet soll als beendet gelten
Voraussetzung	Ein Spielstand ist vorhanden und ein Stein befindet sich kurz vor dem Zielfeld
Funktionsablauf	Die Funktion moveStone wird aufgerufen
Erwartetes Ergebnis	Der Stein wird auf das Zielfeld bewegt und landet dadurch bei den beendeten Steinen des Spielers

Testfall-Nr. TC10	
Auslöser	Ein Spieler gewinnt wenn er 7 (alle) beendete Stein hat
Voraussetzung	Ein Spielstand ist vorhanden mit 6 Steinen im Ziel und einem kurz davor
Funktionsablauf	Die Funktion moveStone wird aufgerufen
Erwartetes Ergebnis	Der Spieler gewinnt das Spiel

Testfall-Nr. TC11	
Auslöser	Ein Spieler kann erneut würfeln wenn sein Stein auf einem Spezialfeld lande
Voraussetzung	Ein Spielstand ist vorhanden mit einem Stein auf dem Spielbrett
Funktionsablauf	Die Funktion moveStone wird aufgerufen
Erwartetes Ergebnis	Der Spieler kann erneut würfeln

3 Projektmanagement und Planung

Als Projektmanagement Methode wurde Kanban verwendet. Im GitHub ist das Kanban-Board ersichtlich. Nach der Themenfindung unseres Projekts, haben wir die verschiedenen Varianten des Royal Game of Ur analysiert und die für uns geeignete Version gewählt. Diese haben wir mithilfe von Kanban Items Schritt für Schritt implementiert.

4 Wissensbeschaffung

Wir haben uns im Internet über die verschiedenen Spielarten informiert und die für uns am besten geeignete gewählt. (MasterofGames, 2020)

5 Konzeptionelle Umsetzung

Zu Beginn unseres Projektes haben wir ein Mockup erstellt.

6 Abgrenzungen und Schnittstellen

6.1 Abgrenzungen

Um den Aufwand im Bereich des realistischen zu halten, haben wir uns für folgende Abgrenzungen entschieden.

- Kein Online Multiplayer
- Auch lokal nur Einzelspieler möglich
- Eine Spielvariante
- Gegner ist ein einfacher Computerspieler

6.2 Schnittstellen

Wir verwenden keine Schnittstellen, da unser Frontend direkt mit dem Backend verbunden ist.

7 Lösungsvarianten

Wir wollten keine Desktopanwendung machen und haben uns daher für eine Webanwendung entschieden. Für ein Projekt dieser Grösse ist das Sinnvoll und es ist auch eine modernere Herangehensweise für so ein Projekt. Ziemlich früh im Projekt haben wir gemerkt das die uns gewohnte Herangehensweise des objektorientierten Programmierens nicht sehr gut geeignet ist für diese Umsetzung. Daher haben wir uns entschieden unser Projekt funktional umzusetzen.

8 Potenzielle Features

Potenzielle Features haben wir lokal getestet, da unser Projekt keine Online Funktion benötigt, um funktionieren zu können. Für neue Features haben wir nicht mit Pull Requests gearbeitet, da wir stets im Pair-Programming gearbeitet und da unsere Änderungen gleich besprochen haben.

9 Entwicklungsumgebung

Die Verwendung und Installation unserer Entwicklungsumgebung haben wir im Wiki unseres GitHub beschrieben. (GitHub, 2020)

10 Arbeitsjournal

N. Helfenstein, L. Meyer, L. Bucher

Donnerstag, 12. November 2020

Um mit dem Projekt zu starten, mussten wir uns erst Gedanken darüber machen, was wir überhaupt erstellen möchten. Unsere Idee war dann das uralte Spiel «Royal Game of Ur» in einer Webapplikation zur Verfügung zu stellen. Da nun das Projektthema feststand, haben wir ein Git-Repository initialisiert. Ausserdem mussten wir uns überlegen, mit welcher Programmiersprache wir das Projekt umsetzten wollen. Wir haben uns entschieden, das Projekt mit Typescript zu schreiben, dass wir dann eine uns noch eher unbekannte Programmiermethode zur Verfügung steht. Nämlich die funktionale Umsetzung.
Heute haben wir ausserdem das Backlog erstellt und bereits einige Tätigkeiten erstellt.

Durch die Hilfe von Adrian haben wir die Programmiersprache ausgewählt. Probleme hatten wir heute keine Erwähnenswerte.

Donnerstag, 19. November 2020

Heute haben wir mit der Backendprogrammierung begonnen. Dazu haben wir ein Angular Projekt aufgesetzt, dieses auch richtig geroutet und einige unnötigen Zeilen Code gelöscht. Wir haben unsere Typen definiert und eine Methode implementiert, welche vier Random Zahlen generiert und so das Würfeln simuliert. Nebst dem Code haben wir für die Funktionen auch Tests geschrieben.

Ein Problem hatten wir beim Routing denn dieses wurde nicht so gemacht, wie wir es wollten. Luca Bucher hat uns dann geholfen das Problem zu lösen, da er sich mit Angular sehr gut auskennt. Das Problem lag bei einer Zeile Code im HTML, welche wir fälschlicherweise gelöscht haben.

Donnerstag, 26. November 2020

Da das Routing und das Projekt richtig angelegt sind und auch die Typen existieren, konnten wir mit der eigentlichen Programmierung im Backend beginnen. Da für uns das funktionale Programmieren noch neu war, kamen wir nicht so schnell voran und wir mussten uns erst an diese Art gewöhnen. Heute haben wir die Funktion erstellt, welche ein neues Board generieren kann. Um für uns das Testen einfacher zu machen, haben wir uns eine Hilfsmethode erstellt, welche ein Feld mithilfe von mitgegebenen Parametern ausgeben kann. Auch haben wir einige Änderungen an der Typendefinition gemacht. Nebst dem Code haben wir für die Funktionen auch Tests geschrieben.

Freitag, 27. November 2020

Wir haben heute noch mehr Test erstellt und haben auch alte Tests verbessert. Ausserdem haben wir einen Fehler in der Hilfsfunktion von Gestern festgestellt und haben diesen auch noch entfernt.

Donnerstag, 3. Dezember 2020

Heute konnten wir wieder am Backend weiterarbeiten. Wir haben die Funktion «Move», «Capture» und «place Stone on Field» hinzugefügt und getestet.

Donnerstag, 10. Dezember 2020

Wir konnten heute einen grossen Fortschritt im Backend erreichen. Nun kann man im Spiel ein Stein ins Ziel bringen und ausserdem gewinnen, wenn alle Steine im Ziel sind. Auch haben wir eine Funktion hinzugefügt,

welche testet, ob der gewählte Spielzug valid ist. Wenn man auf ein Spezialfeld kommt, kann man neuerdings auch den Spielzug wiederholen. Nebst den Funktionen haben wir auch einige Bugs gefixt und die genannten Funktionen auch getestet.

Donnerstag, 17. Dezember 2020

Da wir nun mit dem grössten Teil des Backendes fertig sind, haben wir heute mit dem Frontend begonnen. Dies gab vorher noch keinen Sinn, da zuerst das Spiel an sich stehen sollte und ein leeres Board auf einem Localhost nichts bringt. Für das Frontend haben wir die Felder und die Steine als Vektorgrafiken erstellt. Die Felder haben wir dann in einem HTML Grid dargestellt. Dies dient uns später für das Bewegen der Steine. Dazu haben wir auch eine Navbar erstellt und Spielerinformationen hinzugefügt. Diese zeigen an, wie viele Steine noch nicht gespielt wurden und wie viele Steine bereits im Ziel sind. Auf dem Frontend kann man nun auch Würfeln.

Freitag, 18. Dezember 2020

Heute haben wir nur kurz am Projekt gearbeitet. Es wurden die Spielerinformationen auf der Seite zentriert und ausserdem eine Manual-Page hinzugefügt. Diese zeigt die Spielregeln auf, nach denen wir das Spiel entwickelt haben.

Mittwoch, 23. Dezember 2020

Wir haben uns Zeit genommen, um unsere Tests zu überarbeiten. Einige mussten neu genannt werden und auch die Tests selbst mussten teils geändert werden. Ausserdem haben wir eine About-Page erstellt und in die Navbar hinzugefügt.

Freitag, 25. Dezember 2020 / Samstag, 26. Dezember 2020

Am 25. Dezember haben wir das Workflow Projekt und das Heroku Projekt aufgesetzt. Bei dem Workflow Projekt gab es keine Probleme und deshalb konnte dies auch speditiv erledigt werden. Bei dem Heroku Projekt verlief das ein wenig anders. Hier gab es leider mehrere Fehler, welche wir durch Ausprobieren Schritt für Schritt lösen mussten.

Das aufsetzten des Github Wokflow Projekts und der ersten YAML-Pipeline waren kein Problem. Als wir aber anschliessend unser Projekt via Heroku live schalten wollten, gab es in der Konsole von Heroku immer Fehler. Die Lösung zu diesem Problem war, dass "ng serve" nicht in der Produktionsumgebung verwendet werden durfte. Also mussten wir einen eigenen Express Server aufsetzen und diesen dann in der Produktionsumgebung verwenden. Danach hat es einwandfrei funktioniert.

Sonntag, 27. Dezember 2020

Heute haben wir unser Frontend-Branch mit dem Master-Branch gemerged, merge-conflicts gelöst und auf Heroku kontrolliert, dass alles richtig läuft.

Montag, 28. Dezember 2020

Am 28. Dezember haben wir die Controlling-Buttons zum Spielen hinzugefügt. Zum einen den Würfel-Knopf und zum anderen den Knopf, um einen neuen Stein auf das Spielfeld zu legen.

Dienstag, 29. Dezember 2020

Da nun das Frontend grösstenteils steht, haben wir uns heute dem Bot gewidmet. Diesen konnten wir mit zwei Funktionen lösen. Die eine Funktion gibt ein Array mit Indexen von Feldern zurück, auf welchen der Spieler einen Stein mit einem validen Spielzug hat. Die andere Funktion gibt zurück, ob ein neuer Stein auf das Spielfeld gelegt werden kann. Bei beiden Funktionen wird der Würfelwert berücksichtigt.

Mittwoch, 30. Dezember 2020

Heute haben die Spielzüge «Move» und «Place new stone» für den Spieler auf dem GUI umgesetzt. Dies haben wir so gelöst, dass jeweils der aktuelle «GameState» mit der zu tätigenden Aktion verändert wird und dann mit einer Render Funktion wird das Brett jeweils komplett gerendert. Diese Methode hilft uns später dann auch für den Upload eines vorherig gespielten Spielstatus. Ausserdem haben wir heute den Download des aktuellen Spielstatus mit einem Button ermöglicht.

Freitag, 1. Januar 2021

Vorgestern haben wir die Spielzüge eines Spielers implementiert. Heute haben wir dann das Gegenstück dazu, den Bot, hinzugefügt. Auch mussten wir einen Bug lösen, der sich in unserer «Move» Funktion befand. Wir haben ausserdem eine Hilfsfunktion erstellt, welche irgendein Feld auf dem Brett in das «nthSquare» eines gewünschten Spielers konvertiert.

Samstag, 2. Januar 2021

Wenn ein Spieler auf ein Feld mit einer Rose kommt, darf er erneut Würfeln und somit auch fahren. Dies haben wir heute nebst einem Bugfix erledigt.

Sonntag, 3. Januar 2021

Um den Spielstatus, welcher zuvor gedownloadet wurde, zu laden, haben wir heute bereits die Seite dafür zur Verfügung gestellt.

Dienstag, 5. Januar 2021

Beim Spielen unseres Spiels haben wir einen Bug in der Verzögerung der AI gefunden. Der Spieler kann in dieser Zeit weiterspielen. Diesen Bug haben wir heute gelöst.

Mittwoch, 6. Januar 2021

Heute haben wir die Seite gelöscht, auf der man den Spielstatus uploaden soll. Wir dachten es seit übersichtlicher, wenn der Upload Button direkt neben dem Download Button ist und auch wenn alle Knöpfe auf einer Seite sind. Daher haben wir den Knopf auf der Homepage erstellt und die dazu nötigen Funktionen geschrieben. Es kann nun nur ein JSON hochgeladen werden, welches die Attribute «Player», «AI» und «board» hat.

Donnerstag, 7. Januar 2021

Da nun heute der Abschluss des Projekts bevorsteht, mussten noch einige Sachen erledigt werden. Wir haben die Gewinnfunktion hinzugefügt. Wenn also ein Spieler gewinnt, wird er auf eine Seite umgeleitet und es wird im mitgeteilt, dass er gewonnen oder verloren hat. Dort hat er auch die Möglichkeit, nochmals zu Spielen. Ausserdem haben wir die Funktion erstellt und im GUI implementiert, um den vorherigen Spielstand zu rekonstruieren und von dort aus weiterzuspielen. Auch haben wir das Linting gemacht und die Linting - Pipeline korrigiert. Um das Projekt zu beenden mussten wir noch die Dokumentation beenden.

11 Schlusswort

Während dem Projekt haben wir sehr viel gelernt, da für uns alle ungewohnt sind funktional zu Programmieren. Nach einer kurzen Eingewöhnungsphase kamen wir trotzdem sehr gut voran. Wir denken es war die richtige Entscheidung dieses Projekt funktional zu programmieren. Wir hatten auch nicht so viel Erfahrung mit Typescript selbst. Das Ganze mit den Pipelines und der Herokuapp war uns ebenfalls noch neu, aber es ist sehr toll wie das jetzt funktioniert und wir werden das auch in Zukunft bei unseren Projekten wieder so verwenden. Es hat sehr viel Spass unser Projekt zu entwickeln, trotz einiges langsamen Starts.

12 Literaturverzeichnis

GitHub. (2020). Von https://github.com/NevinJulian/RoyalGameOfUr/wiki/IDE-aufsetzen abgerufen

MasterofGames. (2020). Von https://www.mastersofgames.com/rules/royal-ur-rules.htm#:~:text=It%20is%20generally%20agreed%20that,move%20one%20of%20their%20pieces abgerufen