User Stories und Tasks  
JReversi

Oliver Aeschbacher, René Kamer

|  |  |
| --- | --- |
| **Layout: Spielfeld** | Est: 6h |
| Implementation des klassischen Reversi Spielfelds | |
| Prio: 1 | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task1 Datenstruktur für das Spielfeld bereitstellen | | Task2 GUI Repräsentation des Spielfelds implementieren (mit Spielfeld-Nummerierung) | | Task3 Abbildungsfunktion von Text-Repräsentationen auf die Datenstruktur erstellen | |
| Zust: OA | Est: 1h | Zust: OA | Est: 4h | Zust: RK | Est: 1h |

|  |  |
| --- | --- |
| **Design + Funktion: Ziehender** | Est: 4h |
| Es soll der Spieler angezeigt werden, der am Zug ist. | |
| Prio: 1 | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task4 GUI-Informationsfeld implementieren | | Task5 Spielerklasse erstellen | | Task6 Zugfunktion (Spielerwechsel) implementieren | |
| Zust: RK | Est: 1.5h | Zust: OA | Est: 1.5h | Zust: RK | Est: 1h |

|  |  |
| --- | --- |
| **Design + Funktion: Zug-Angabe** | Est: 4h |
| Zu beiden Spielern soll der letzte Zug angezeigt werden sowie die Anzahl Steine die liegen. | |
| Prio: 3 | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task7 Funktion für das Zählen der Anzahl Steine implementieren | | Task8 Letzter Zug anzeigen lassen (Auslesefunktion) | | Task9 GUI Repräsentation der liegenden Spielsteine implementieren (Anzahl anzeigen im Infofeld) | |
| Zust: OA | Est: 1h | Zust: RK | Est: 1h | Zust: OA | Est: 2h |

|  |  |
| --- | --- |
| **Design + Funktion: Spielkontrolle** | Est: 6h |
| Der Spieler soll aufgeben können sowie ein Unentschieden anbieten, ein neues Spiel beginnen oder das Spiel abbrechen können (= Verloren) | |
| Prio: 4 | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task10 Implementation: Unentschieden anbieten | | Task11 GUI Button: Unentschieden | | Task12 GUI Button: neues Spiel starten | |
| Zust: OA | Est: 1h | Zust: RK | Est: 1h | Zust: RK | Est: 1h |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task13 GUI Button: Aufgabe des Spielers | | Task14 Implementation: Aufgabe des Spielers | | Task15 GUI Button: Spiel abbrechen | |
| Zust: RK | Est: 1h | Zust: OA | Est: 1h | Zust: RK | Est: 1h |

|  |  |
| --- | --- |
| **Startseite: Spieleinstellungen** | Est: 5h |
| Man soll wählen können, ob gegen Mensch oder Maschine gespielt wird. (Startbildschirm oder in Game?). Design und Funktion. | |
| Prio: 2 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task16 GUI Repräsentation implementieren (Radio-Button, Startknopf) | | Task17 Implementation: Wählen der richtigen Klasse abhängig vom Radio-Button | |
| Zust: OA | Est: 3.5h | Zust: OA | Est: 1.5h |

|  |  |
| --- | --- |
| **Spielansicht: Darstellung der Steine** | Est: 4h |
| Die Spielsteine sollen auf dem Spielfeld dargestellt werden. | |
| Prio: 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task18 Darstellung eines Spielsteines implementieren | | Task19 Implementation: Auslesen des Spielfeldes und setzen der Steine auf dieses. | |
| Zust: OA | Est: 2.5h | Zust: RK | Est: 1.5h |

|  |  |
| --- | --- |
| **Spiellogik: Wo dürfen Steine gesetzt werden** | Est: 5h |
| Es muss berechnet werden, wo die Steine gesetzt werden dürfen (nach Regeln). | |
| Prio: 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task20 Algorithmus zur Berechnung der erlaubten Felder erstellen. | | Task21 Anzeige der erlaubten Felder implementieren | |
| Zust: RK | Est: 3h | Zust: RK | Est: 2h |

|  |  |
| --- | --- |
| **Spiellogik: Schlagen der Steine** | Est: 5h |
| Wenn ein Spielstein gesetzt wird, müssen Steine umgedreht werden. Dies soll hier realisiert werden. | |
| Prio: 1 | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task22 Algorithmus zur Berechnung, welche Steine umgedreht werden sollen, entwickeln | | Task23 Algorithmus aus Task22 implementieren | | Task24 Umdrehen der Steine im GUI darstellen | |
| Zust: RK | Est: 1.5h | Zust: RK | Est: 1h | Zust: OA | Est: 2.5h |

|  |  |
| --- | --- |
| **Ansicht: Anzeige der Spielsteine, welche Umgedreht werden** | Est: 4h |
| Bei der Anzeige der möglichen Züge soll auch angezeigt werden, wie viele Steine sich umdrehen würden, wenn an diesem Ort gesetzt wird. | |
| Prio: 3 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task25 Anzeige der Zahlen im Feld implementieren | | Task26 Mittels vorher angelegtem Algorithmus die drehenden Steine bestimmen und ablegen. | |
| Zust: OA | Est: 1h | Zust: OA | Est: 3h |

|  |  |
| --- | --- |
| **Spiellogik: Passen** | Est: 5h |
| Es soll ein Algorithmus laufen, welcher erkennt, ob man passen muss und dies direkt vornehmen (mit Angabe im Spielerfeld „letzter Zug“) | |
| Prio: 3 | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task27 Implementation:  Mittels Algorithmus, welcher die erlaubten Züge bestimmt, ermitteln, ob es einen möglichen Zug gibt.  (in 1. Iteration mittels „passen Knopf“ implementieren) | | Task28 Passen implementieren (Wechsel zum anderen Spieler) als Funktion | | Task29 GUI-Anzeige im Spielerfeld implementieren | |
| Zust: OA | Est: 2.5h | Zust: OA | Est: 1h | Zust: RK | Est: 1.5h |

|  |  |
| --- | --- |
| **Spiellogik: Setzen eines Steines** | Est: 3h |
| Beim Setzen eines Steines soll das Spielfeld entsprechend aktualisiert werden. | |
| Prio: 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task30 Funktion zum Setzen des Steines in der logischen Struktur realisieren | | Task31 Anzeige des gemachten Zuges implementieren mittels entsprechender Funktionen | |
| Zust: OA | Est: 2h | Zust: RK | Est: 1h |

|  |  |
| --- | --- |
| **Ansicht: Anzeige der Eröffnung** | Est: 6h |
| Beim Eröffnungsspiel soll die Eröffnung (falls offiziell vorhanden) angezeigt werden. | |
| Prio: 4 | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task32 Eröffnungs-Bibliothek online konsultieren. | | Task33 Eröffnungs-Bibliothek in entsprechender Struktur realisieren | | Task34 Ablage für die bereits getätigten Züge implementieren | |
| Zust: RK | Est: 1h | Zust: RK | Est: 0.5h | Zust: RK | Est: 0.5h |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Task35 Funktion zum Vergleich der Ablage mit der Eröffnungsbibliothek implementieren | | Task36 GUI Anzeigefeld für die Anzeige der Eröffnung implementieren | | | Task37 Anzeige der Eröffnung implementieren | | |
| Zust: OA | Est: 1h | Zust: OA | | Est: 2h | Zust: OA | Est: 1h | |
| **Code: Erstes Refactoring vor Rest der Iteration 3** | | | Est: 8h | | | |
| Bevor mit der dritten Iteration angefangen wird, wird ein Refactoring durchgeführt. | | | | | | |
| Prio: 1 | | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task38 Refactoring Gameplay | | Task39 Refactoring GUI | | Task40 Refactoring Settings | |
| Zust: RK | Est: 3h | Zust: OA | Est: 3h | Zust: RK | Est: 2h |

|  |  |
| --- | --- |
| **Funktion: Implementation der Heuristik** | Est: 10h |
| Es wird eine Heuristik für die Bewertung der Spielsituationen im Spielbaum angelegt. Je eine für „Einfach“, „Mittel“ und „Schwer“ | |
| Prio: 1 | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task41 Implementation Heuristik „Einfach“ | | Task42 Implementation Heuristik „Mittel“ | | Task43 Implementation Heuristik „Schwer“ | |
| Zust: RK | Est: 2h | Zust: RK | Est: 3h | Zust: RK | Est: 5h |

|  |  |
| --- | --- |
| **Funktion / Logik: Implementation des Spielbaumes** | Est: 8h |
| Für die Bewertung des Spiels muss ein Spielbaum angelegt werden. | |
| Prio: 1 | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task44 Implementieren einer geeigneten Node | | Task45 Verknüpfen der Nodes in einen Spielbaum | | Task46 Implementation des Traversierens des Spielbaums | |
| Zust: OA | Est: 3h | Zust: OA | Est: 3h | Zust: OA | Est: 2h |

|  |  |
| --- | --- |
| **Funktion / Logik: Implementation des Alpha-Beta Algorithmus** | Est: 15h |
| Die Implementation des Alpha-Beta Algorithmus ermöglicht dem Computer die Wahl des idealen Spielzugs. | |
| Prio: 1 | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task47 Implementieren des MiniMax Algorithmus (Vorbereitung) | | Task48 Erweiterung des MiniMax Algorithmus auf Alpha-Beta Algorithmus | | Task49 Verknüpfung des Algorithmus mit der Heuristik | |
| Zust: OA/RKA | Est: 5h | Zust: OA/RKA | Est: 5h | Zust: OA/RKA | Est: 2h |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task50 Ausgiebige Tests des Alpha-Beta Algorithmus | | Task51 Verknüpfen des errechneten Zuges mit der Move Methode | | |
| Zust: RK | Est: 2h | Zust: OA | | Est: 1h |
| **Funktion / Logik: Verfeinern der Spielstärke (Heuristik)** | | | Est: 5h | | |
| Um eine annehmbare Spielstärke zu entwickeln, muss einige Male gespielt werden und jeweils die Heuristik angepasst werden. | | | | | |
| Prio: 3 | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task52 Test-Spiele absolvieren | | Task53 Anpassungen in der Heuristik | |
| Zust: OA | Est: 2.5h | Zust: RK | Est: 2.5h |

|  |  |
| --- | --- |
| **Gesamtprogramm: Beta Tests durch externe Tester** | Est: 8h |
| Um eine hohe Qualität zu gewährleisten, werden Tests mit externen Beta Testern gemacht und die auffälligen Probleme noch behandelt. | |
| Prio: 4 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task54 Begleitung der Beta-Tests | | Task55 Anpassungen am Programm | |
| Zust: OA / RKA | Est: 5h | Zust: RKA / OA | Est: 3h |

|  |  |
| --- | --- |
| **Dokumentation: Finalisieren der Dokumentation** | Est: 3h |
| Die während des Projekts geführte Dokumentation wird nochmals durchgelesen und kleinere Anpassungen vorgenommen. | |
| Prio: 3 | |

|  |  |
| --- | --- |
| Task56 Finalisieren der Dokumentation | |
| Zust: OA / RKA | Est: 3h |

|  |  |
| --- | --- |
| **Einlesen in Spieltheorie und Algorithmen** | Est: 5h |
| Um das Vorgehen abzuschätzen muss sich das Team zuerst in die Materie einlesen. | |
| Prio: 2 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Task57 Einlesen Spieltheorie | | Task58 Einlesen in Algorithmen (MiniMax, Alpha Beta) | |
| Zust: OA/RKA | Est: 2.5h | Zust: OA/RKA | Est: 2.5h |