Funktionale und objektorientierte Programmierkonzepte
Projekt

TECHNISCHE UNIVERSITÄT DARMSTADT

Entwurf

Achtung: Dieses Dokument ist ein Entwurf und ist noch nicht zur Bearbeitung/Abgabe freigegeben. Es kann zu Änderungen kommen, die für die Abgabe relevant sind. Es ist möglich, dass sich **alle** Aufgaben noch grundlegend ändern. Es gibt keine Garantie, dass die Aufgaben auch in der endgültigen Version überhaupt noch vorkommen und es wird keine Rücksicht auf bereits abgegebene Lösungen genommen, die nicht die Vorgaben der endgültigen Version erfüllen.

FOP Projekt **Gesamt:**
Die Siedler von Catan

Beachten Sie die Seite *Verbindliche Anforderungen für alle Abgaben* im Moodle-Kurs.

Verstöße gegen verbindliche Anforderungen führen zu Punktabzügen und können die korrekte Bewertung Ihrer Abgabe beeinflussen. Sofern vorhanden, müssen die in der Vorlage mit `TODO` markierten `crash`-Aufrufe entfernt werden. Andernfalls wird die jeweilige Aufgabe nicht bewertet.

Die für diese Hausübung relevanten Verzeichnisse sind `src/main/java/hProjekt` und ggf. `src/test/java/hProjekt`.

Organisation und Information

Grundlegende Informationen und Bonus

Das FOP-Projekt ist zwar nicht verpflichtend und auch für die formale Prüfungszulassung (Studienleistung) ausdrücklich nicht notwendig, aber mit dem FOP-Projekt können zusätzliche Bonuspunkte erreicht werden. Es wird jedoch dringend empfohlen am FOP-Projekt teilzunehmen, da Sie hier richtig programmieren lernen. In den weiterführenden Semestern wird dies mehr oder weniger stillschweigend vorausgesetzt. Voraussetzung für den Bonus ist die bestandene Studienleistung. Neben den Bonuspunkten der Hausübungen können Sie sich im Projekt weitere Bonuspunkte für die Klausur erarbeiten. Im Projekt können Sie zusätzlich XX Punkte erreichen, welche am Ende durch X.X geteilt und auf die nächstkleinere, ganze Zahl abgerundet werden. Zusammenfassend können Sie also bis zu XX Punkte zusätzlich für Ihr Punktekonto erarbeiten. Sie werden das Abschlussprojekt in Teams von genau vier Personen bearbeiten.

Zeitplan im Überblick

- Date - Time: Deadline zur Anmeldung
- Date - Time: Verbindliche Teamtrainings der HDA
- Date - Time: Poolsprechstunden
- 15.03.2023 - 23:50 Uhr: Abgabe des Projekts

Poolsprechstunden

Für eventuell anfallende Fragen bieten die Projekttutoren vom **Datum-von - Datum-bis** Sprechstunden an. Eine Übersicht über die Sprechstundentermine finden Sie im Projektabschnitt des zugehörigen moodle-Kurses.

Moodle - Forum

Es werden für Sie zwei Foren im Projektabschnitt des Moodle-Kurses bereitgestellt. In einem Forum können Sie Fragen hinsichtlich der Organisation und des Zeitplans stellen. Das andere Forum ist für Fragen rund um die Projektaufgaben gedacht. Diese werden natürlich auch außerhalb der Poolsprechstunden beantwortet.

Anmeldung (bis Datum)

Um sich anzumelden, bilden Sie Gruppen aus **genau** 4 Studierenden. Sollten Sie noch keine Gruppe haben, so steht Ihnen im Projektabschnitt des **moodle**-Kurses ein Forum zur Gruppenfindung zur Verfügung. Bei der Eintragung in die Gruppen wählen Sie gleichzeitig Ihren Termin für das Teamtraining der HDA aus. Alle Gruppenmitglieder müssen sich manuell in die Gruppen eintragen.

Abgabe des Projekts

Das Projekt ist bis zum **15.03.2023 um 23.50 Uhr Serverzeit** auf moodle abzugeben. Das Projekt exportieren Sie genauso wie die Hausübungsabgaben als ZIP-Archiv, hier gelten die gleichen Konventionen. Die Benennung erfolgt folgendermaßen: Projektgruppe xxx, wobei der Suffix xxx durch Ihre Gruppennummer zu ersetzen ist. Dabei gibt eine Person aus Ihrer Gruppe das gesamte Projekt ab. Neben dem Code geben Sie zusätzlich eine PDF-Datei ab, diese soll sich ebenfalls im ZIP Archiv befinden. In der PDF finden sich zum einen die, für unser Verständnis notwendigen, Dokumentationen der weiterführenden Aufgaben sowie die Lösungen der Theorieaufgaben.

Verbindliche Anforderung (Für die Abgabe des Projekts)**:**

Nachdem eines Ihrer Teammitglieder das Projekt auf moodle hochgeladen hat, müssen alle anderen Teammitglieder diese Aufgabe im entsprechenden Modul bestätigen. Andernfalls wird die Aufgabe nur im Entwurfsmodus gespeichert und nicht als Abgabe gewertet. **Es werden keine Abgaben im Entwurfsmodus akzeptiert. Diese werden nicht bewertet und unabhängig vom Inhalt der Abgabe mit 0 Punkte bewertet.** Geben Sie daher Ihr Projekt nicht kurz vor der Deadline ab, da Ihre Teammitglieder genügend Zeit haben müssen, um die Abgabe zu bestätigen.

Plagiarismus und die Nutzung von KI

Selbstverständlich gelten die gleichen Regelungen zum Plagiarismus und zur Nutzung von KI, wie auch bei den Hausübungsabgaben. Daher hier nochmal der Hinweis, dass Ihr gemeinsames Repository für die Gruppenarbeit privat sein sollte. Beachten Sie: Sollte Ihre Dokumentation der Lösungswege unvollständig sein oder uns den Anlass für einen Verdacht geben, dass Sie nicht nur innerhalb der eigenen Gruppe gearbeitet haben, so müssen Sie damit rechnen, dass Sie Ihre Ergebnisse bei einem privaten Testat bei Prof. Weihe persönlich vorstellen und erläutern müssen.

Inhaltliche Information zum Projekt

Im Laufe des Projekts werden Sie eine Implementation des beliebten Brettspiel „Die Siedler von Catan“ erarbeiten und diese ergänzen. Dazu wird Ihnen eine Vorlage zur Verfügung gestellt, in welcher Sie, je nach Aufgabenstellung, Code ergänzen oder erweitern müssen, damit eine lauffähige Version des Brettspiels entsteht. Unter anderem werden Sie viele bekannte Themen aus der Vorlesung behandeln und anwenden müssen, sodass das Projekt eine gute Vorbereitung auf die Klausur darstellt. Selbstverständlich wünschen wir Ihnen viel Spaß bei der Implementierung des Brettspiels und wir freuen uns auf die, hoffentlich gelungenen, Implementationen.

Spielekonzept Die Siedler von Catan

Im Brettspiel „Die Siedler von Catan“ erschaffen die Spieler gemeinsam eine Inselwelt, auf der sie Siedlungen bauen, Straßen errichten und Rohstoffe sammeln. Das Ziel ist es, 10 oder mehr Siegpunkte zu erreichen, indem man Siedlungen erweitert und Entwicklungskarten erwirbt.

Spielematerial

- Hexagonfeldern mit unterschiedlichen Geländeformen: Wälder, Hügel, Felder, Berge und Wüsten
- Siedlungen und Städte in verschiedenen Farben für jeden Spieler
- Straßen für den Bau von Wegen zwischen den Siedlungen
- Rohstoffkarten für Ressourcen wie Holz, Lehm, Getreide, Erz und Wolle
- Zwei Würfel zur Generierung von Ressourcen
- Entwicklungskarten mit verschiedenen Funktionen
- Einem Räuber

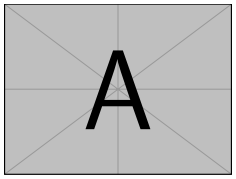

Abbildung 1: Ausschnitt eines Spielfeldes mit bebauten Siedlungen, Straßen und einem Räuber

Spieleablauf

Start

Die Spieler platzieren abwechselnd ihre Siedlungen und Straßen auf der Insel. Jeder Spieler beginnt mit zwei Siedlungen und zwei Straßen.

Aktionen

Reihum würfeln die Spieler zu Beginn ihres Zuges. Die gewürfelte Zahl bestimmt, welche Felder Rohstoffe produzieren. Spieler erhalten Ressourcen, wenn die Zahl auf den Feldern gewürfelt wird, die an ihre Siedlungen angrenzen. Danach können die Spieler handeln, Straßen und Siedlungen bauen oder Entwicklungskarten erwerben.

Wertung während des Spiels

- **Siedlungen:** Jede Siedlung bringt 1 Siegpunkt.
- **Städte:** Verbesserte Versionen von Siedlungen, die 2 Siegpunkte bringen.
- **Längste Handelsstraße:** Bonuspunkte für den Spieler mit der längsten durchgehenden Straße (2 Siegpunkte).
- **Größte Rittermacht:** Bonuspunkte für den Spieler mit den meisten Ritterkarten (2 Siegpunkte).

Spielende

Das Spiel endet, wenn ein Spieler mindestens 10 Siegpunkte erreicht. Dann erfolgt die Endwertung:

- **Siegpunkte für Siedlungen und Städte:** 1 Siegpunkt für jede Siedlung, 2 Siegpunkte für jede Stadt.
- **Längste Handelsstraße:** Bonuspunkte (2 Siegpunkte) für den Spieler mit der längsten durchgehenden Straße.
- **Größte Rittermacht:** Bonuspunkte (2 Siegpunkte) für den Spieler mit den meisten Ritterkarten.

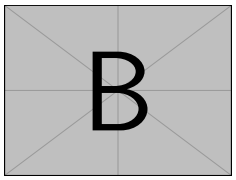

Abbildung 2: Das Beispiel zeigt die Punktezählung für den gelben Spieler. Durch den Besitz der längsten Handelsstraße und drei Siedlungen hat der gelbe Spieler insgesamt 5 Siegpunkte erreicht.

Zusätzliche Informationen

Weitere Informationen sowie die offizielle Spielanleitung erhalten Sie unter den nachfolgenden Links:

- Offiziele Anleitung
- Wikipedia Artikel
- Youtube Video, welches die Spielregeln erläutert

Implementationsstruktur

HexGrid - Das Spielfeld

Das HexGrid repräsentiert das Spielfeld als hexagonale Struktur. Es besteht aus einzelnen Tiles, die die Spielumgebung definieren.

Das Spielfeld verwendet ein **axiales Koordinatensystem** mit den Achsen q, r und s. Die Koordinaten müssen der Restriktion q + r + s = 0 entsprechen. Durch dieses System können Tiles auf dem Spielfeld eindeutig lokalisiert werden. Die s-Koordinate wird berechnet als s = -q - r.

Die Verwendung von **Spiralringen** ist eine effektive Methode, um systematisch durch die Hexagon-Kacheln zu navigieren. Unser Algorithmus teilt das Spielfeld in Ringe auf und füllt dann die Hexagon-Kacheln in jedem Ring nacheinander. Dies ermöglicht eine geordnete Durchquerung des HexGrids, was für die Implementierung des Setzen der einzelnen Tiles auf dem Spielfeld entscheidend ist. Spiralringe bieten auch die Möglichkeit, die Anzahl der Hexagon-Kacheln in einem größeren Hexagon zu berechnen und können für die Bestimmung von Bewegungsbereichen in Spielen verwendet werden.

Eine äußerst **informative Erklärung** mit anschaulichen Grafiken, die das axiale Koordinatensystem veranschaulichen, steht für eine vertiefte Einsicht unter dem folgenden Link zur Verfügung: Hexagonal Grids. Bei der Umsetzung des Spielfeldes haben wir uns stark von dieser Erklärung inspirieren lassen, um die grundlegenden Konzepte klar und verständlich zu integrieren.

TilePosition

Die TilePosition-Klasse ist zentral für die Positionierung auf dem Spielbrett im axialen Koordinatensystem. Sie repräsentiert eine Position durch die axialen Koordinaten q und r, wobei die s-Koordinate berechnet wird, um die Bedingung q + r + s = 0 zu erfüllen. Mit Funktionen wie add, subtract und scale ermöglicht die Klasse grundlegende Rechenoperationen zwischen Positionen. Eine wichtige Funktion ist neighbours, die alle benachbarten Positionen einer gegebenen Position zurückgibt. Die Richtungen um eine Position werden durch EdgeDirection definiert, wobei Funktionen wie left und right die Navigation zwischen benachbarten Richtungen erleichtern. Diese Richtungen sind relevant für den Spielaufbau, insbesondere für den Bau von Straßen und Siedlungen. Zusätzlich werden Richtungen in Bezug auf Schnittpunkte (IntersectionDirection) definiert, wobei jede Richtung ”linkeünd ”rechte”Nachbarrichtungen hat, was beim Aufbau von Straßen und Siedlungen eine Rolle spielt.

Tile

Die Tile-Klasse fungiert als einer der zentralen Elemente in der Implementierung des Spiels ”Die Siedler von Catan”. Jedes Tile repräsentiert einen Baustein des Spielfelds und bietet Zugriff auf verschiedene Eigenschaften und Operationen. Diese umfassen Höhe und Breite als ObservableDoubleValues, den Tile-Typ (WOODLAND, MEADOW, FARMLAND, HILL, MOUNTAIN und DESERT), eine mit dem Tile verbundene Würfelzahl und die Position im axialen Koordinatensystem. Das Tile ist eng mit dem HexGrid verbunden und ermöglicht es, die Nachbarn eines Tiles abzurufen sowie Informationen zu Intersections (Schnittstellen) und den darauf platzierten Siedlungen und Straßen zu erhalten. Es gibt auch Funktionen zum Platzieren und Abrufen von Straßen und Siedlungen. Darüber hinaus kann überprüft werden, ob sich der Räuber derzeit auf dem Tile befindet. Die verschiedenen Tile-Typen sind Farben und Ressourcenarten zugeordnet. Insgesamt bildet die Tile-Klasse die Grundlage für die Implementierung der Spielmechanik von ”Die Siedler von Catanäuf dem HexGrid, indem es die Struktur und Beziehungen zwischen den Bausteinen des Spielfelds definiert.

Intersection

Die Intersection-Klasse repräsentiert die Schnittpunkte auf dem Spielfeld. Es bietet verschiedene Funktionen zur Interaktion mit den Spielmechaniken. Durch die Klasse können Spieler Siedlungen auf den Schnittpunkten platzieren, vorhandene Siedlungen zu Städten verbessern und Handelsplätze (Ports) nutzen. Zudem ermöglicht es den Zugriff auf verbundene Straßen, angrenzende Schnittpunkte und benachbarte Tiles. Die Funktionen umfassen das Abrufen von Informationen, wie platzierten Siedlungen, Upgrades zu Städten, zugehörigen Häfen und verbundenen Straßen.

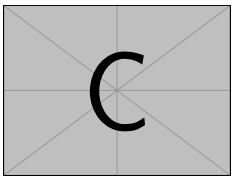

Abbildung 3: Die Abbildung zeigt das Spielfeld. Die roten Punkte stellen die einzelnen Schnittpunkte dar.

Das Klasse erleichtert auch den Zugriff auf benachbarte Schnittpunkte und Tiles sowie das Prüfen der Verbindung zu bestimmten Positionen.

Road

Die ”RoadKlasse fungiert als Bindeglied im HexGrid und repräsentiert die Straßen, die die verschiedenen Siedlungen und Städte der Spieler verbinden. Diese Klasse ermöglicht die Definition von Straßenabschnitten zwischen zwei Positionen auf dem Spielfeld und hält Informationen über den Besitzer der Straße fest. Durch die Implementierung der ”RoadKlasse erhalten Spieler Zugriff auf wichtige Funktionen, die die Interaktionen mit benachbarten Straßen und Kreuzungen erleichtern. Die Methode ”getAdjacentTilePositions()”gibt die Positionen der benachbarten Tiles der Straße zurück, während ”getIntersections()”die Kreuzungen liefert, die mit der Straße verbunden sind. Besonders relevant ist die Fähigkeit, zu prüfen, ob eine Straße mit einer anderen verbunden ist (”connectsTo()”). Dies ermöglicht Spielern, die Kontinuität ihrer Straßennetze zu überwachen. Zusätzlich erlaubt ”getConnectedRoads()”den Zugriff auf alle benachbarten Straßen, die mit den anliegenden Kreuzungen verbunden sind.

Player

Die ”PlayerKlasse bietet einen umfassenden Zugriff auf die verschiedenen Aspekte eines Spielers. Hier erhält man Einblicke in die Ressourcenverwaltung, den Status von Siedlungen und Straßen sowie die Entwicklungskarten des Spielers. Über die Klasse können Spieler ihre aktuellen Ressourcen und deren Mengen abfragen sowie Ressourcen hinzufügen oder entfernen. Informationen zu den von einem Spieler kontrollierten Straßen und Siedlungen sind ebenfalls zugänglich. Weiterhin besteht die Möglichkeit, die verbleibenden Optionen für den Bau von Straßen und Siedlungen zu ermitteln. Die Entwicklungskarten des Spielers, einschließlich ihrer Anzahl und Typen, können über die Schnittstelle verwaltet werden. Darüber hinaus bietet die Schnittstelle Zugriff auf zusätzliche Informationen wie die Anzahl der gespielten Ritter und die Spielerfarbe.

Implementation des Models

Im Package *projekt.model* sind die oeben genannten Klassen implementiert. Um die Klassenstruktur und die verschiedenen Beziehungen visuell zu veranschaulichen, haben wir ein Diagram mit Hilfe von **Unified Modeling Language (UML)** modelliert. Wie ein UML Diagram zu lesen ist, können Sie auf dem Wikipedia-Artikel nachlesen. Abbildung 4 zeigt das UML-Diagram der verschiedenen Klassen zur Implementierung des Models vom Spiel.

Design Patterns

Die Struktur und die Implementation des Projekts orientieren sich am Entwurfsmuster Model-View-Controller. Dieses Design Pattern wird im Abschnitt Model-View-Controller des Foliensatzes Fehlersuche und fehlervermeidender Entwurf

behandelt. Wie Sie anhand des Aufbaus des Projekts nachvollziehen können, unterteilt sich das Projekt in drei wichtige Komponenten. Die drei Packages sind:

- projekt.model
- projekt.view
- projekt.controller

Das Package projekt.model implementiert im Wesentlichen die grundlegenden Daten wie den Aufbau eines Spielers oder des Spielfeldes. Das zweite Package projekt.view stellt die Daten der Komponente Model dar und interagiert mit dem Benutzer. Bei Änderungen der View wird die Komponente Controller benachrichtigt. Es werden keine Daten in der Komponente View verarbeitet, sondern nur entgegengenommen. Das Package projekt.controller lenkt und verwaltet die beiden anderen Komponenten.

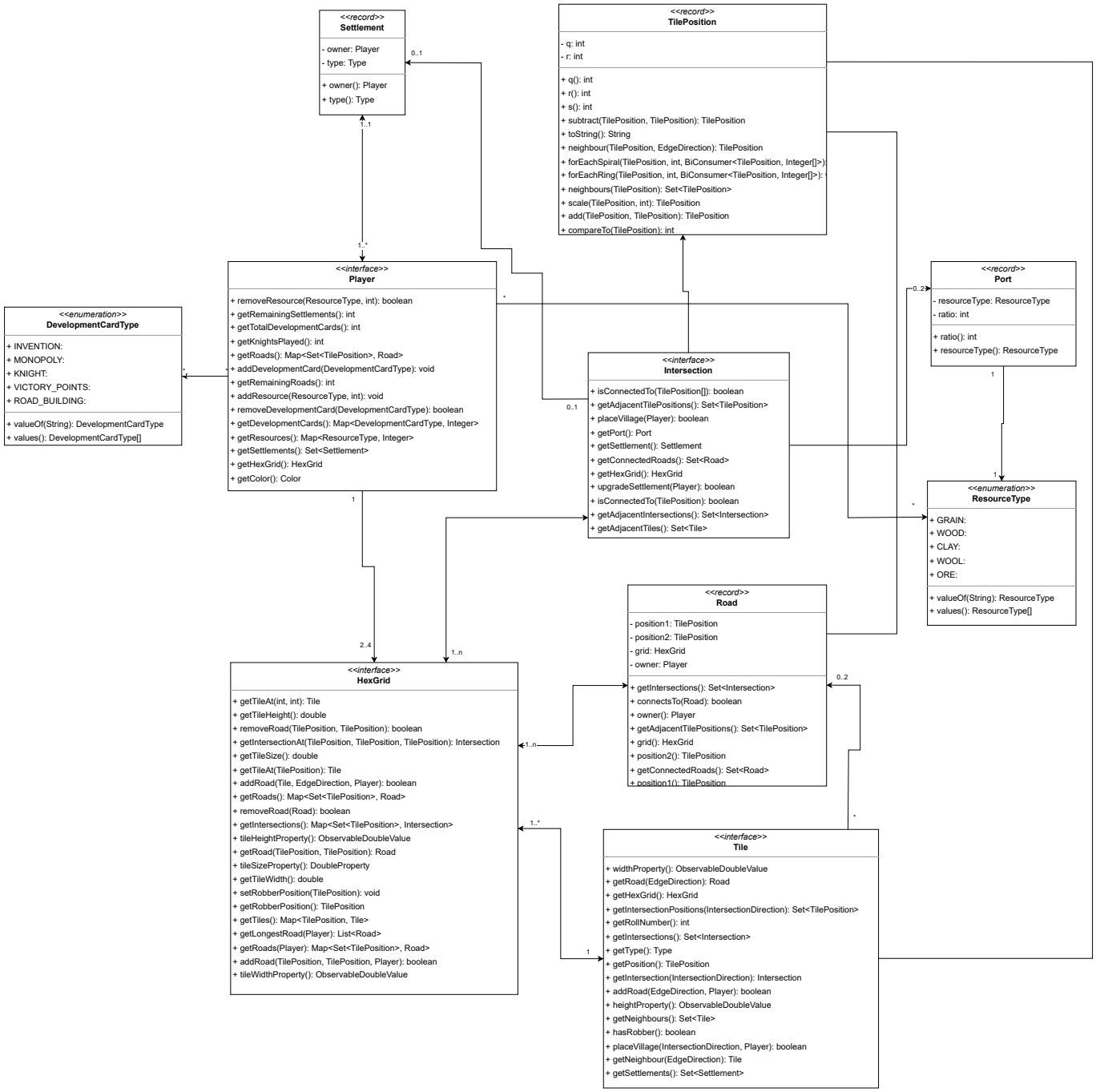

Abbildung 4: Die Abbildung zeigt die Beziehungen der einzelnen Klassen vom model.