

## Spielidee

Website: [Moodle-Service der Universität Wien](#)  
Kurs: 2020W Software Engineering 1  
Buch: Spielidee

Gedruckt von: Srdanovic Nemanja  
Datum: Freitag, 16. Oktober 2020, 12:22

## Inhaltsverzeichnis

### **1. Allgemeiner Leitfaden zur Absolvierung der Übung**

### **2. Hintergrundinformationen**

### **3. Beschreibung der Spielidee**

- 3.1. Beispielhafte Visualisierung einer 8 x 8 Karte
- 3.2. Detaillierte Beschreibung der Bewegungen

# 1. Allgemeiner Leitfaden zur Absolvierung der Übung

In diesem wie auch in jedem anderen Projekt gilt, dass Fehler, die früh entdeckt werden, weniger Aufwand für deren Behebung nach sich ziehen als Fehler, die erst später gefunden werden. Dies gilt auch für die Übung und zugehörige Aufgaben, welche Sie im Rahmen von SE 1 umsetzen bzw. implementieren müssen. Sollten Sie daher beispielsweise einen Fehler in Phase 2 Ihres Konzeptes finden, kann es notwendig sein, auch Phase 1 anzupassen. Achten Sie daher darauf, alle Anforderungen korrekt zu erfüllen und sprechen Sie sich hierzu auch mit anderen Studierenden ab (z.B. zu gegenseitigen Reviews oder um ein Verständnis für die Aufgabenstellung zu entwickeln).

*Achtung:* Dies bedeutet nicht, dass Sie die Aufgaben gemeinsam erledigen oder kopieren dürfen, dies wird als Plagiat behandelt und entsprechend bewertet!

Source-Code-Management Lösungen wie GIT oder SVN sind Standard im Alltag von Softwareentwicklungsprojekten und müssen daher auch von Ihnen hier eingesetzt werden. Committen Sie Ihre Ergebnisse und Zwischenergebnisse früh und häufig (Empfehlung: mindestens einmal täglich). Erstellen Sie Branches für einzelne Features und führen Sie diese später wieder auf den Hauptzweig zurück. Dadurch können Sie Fehler schneller analysieren und, falls notwendig, auf alte Stände zurückgreifen. Source-Code-Management ist eine **zusätzlich Sicherheitsebene für Sie**, verwenden Sie es daher im eigenen Interesse sinnvoll.

Im Folgenden wird die Übungsangabe beschrieben. Diese ist als Freitext formuliert. Diesem müssen Sie zur Erfüllung der gestellten Aufgaben Anforderungen, User-Stories etc. entnehmen. Darauf aufbauend erfolgt danach eine zweistufige Implementierung. Dieser Ansatz bietet Ihnen viel Freiraum, es gibt daher selten nur die eine und einzige richtige Lösung. Stattdessen sind mehrere Lösungen möglich, von denen sich die eine oder andere vielleicht besser eignet oder schneller umsetzen lässt.

Investieren Sie daher die notwendige Zeit, sich Ihre Lösungen **zuerst durchzudenken**, bevor Sie loslegen. Dies wird Ihnen in der Folge viel Aufwand ersparen. Für die Bewertung ist es jedoch (fast) nicht relevant, zu welcher Lösung Sie gelangen, solange die Anforderungen der jeweiligen Teilaufgabe erfüllt sind und die Qualität der Lösung zufriedenstellend ist. Nicht spezifizierte Bestandteile der Teilaufgaben können Sie zu Ihrem Vorteil auslegen.

## 2. Hintergrundinformationen

Vom letzten Game Jam haben Sie zahlreiche Ideen für ein neues, kompetitives, **selbstablaufendes Zwei-Spieler-Spiel** mitgenommen. Bei einem solchen selbstablaufenden Spiel werden alle Aktionen automatisch durch künstliche Intelligenzen (kurz KIs) berechnet und ausgeführt. Hierbei verfolgen die menschlichen Spieler das eigentliche Spielgeschehen ähnlich eines Livestreams nur passiv, *es sollten daher keinerlei Userinteraktionen erforderlich sein*. Dieses Konzept möchten Sie nun umsetzen und in der Folge erfolgreich an zahlreiche Kunden (die menschlichen Spieler) weltweit vermarkten, welche beispielsweise versuchen können, einander mit immer ausgefeilteren künstlichen Intelligenzen zu übertreffen. Die Spieler sollen jeweils ortsungebunden paarweise gegeneinander antreten können.

Andere Game Jam-Teilnehmer haben Ihnen mehrfach berichtet, wie erfolgreich diese ihre Entwicklungen mit strukturierten Ansätzen aus dem Software Engineering Bereich vorangetrieben haben. Sie haben sich daher entschieden, diesen Vorbildern zu folgen und Ihr Vorgehen genau zu planen sowie in drei Teilaufgaben aufzuspalten, die nacheinander abgearbeitet werden.

*Um die Umsetzung des Spiels im Umfang angemessen zu halten, wird auf Berechtigungen, Zertifikate, Authentifizierung, Distribution, rechtliche Belange, externe Geräte und Schnittstellen etc. die in echten Projekten relevant wären, so weit wie möglich **verzichtet**.*

### 3. Beschreibung der Spielidee

Die grundlegende Spielidee ist, dass zwei KIs auf der gleichen Spielkarte eine vergleichbare Aufgabe erfüllen müssen. Die KI, die diese schneller erfüllt, gewinnt das Spiel; welches rundenbasiert abläuft (daher setzt jede KI abwechselnd eine Spielaktion). Die Karte, auf welcher gespielt wird, ist hierbei nicht fest vorgegeben, sondern wird von beiden KIs selbstständig und automatisch beim Start des Spiels erstellt (jeweils eine Hälfte der Karte).

Die Spielfigur der KI startet anschließend jeweils an einer von ihr selbst definierten Position (ihrer Burg) auf dem von ihr erstellten Teil der Karte. Anschließend muss sie die Karte erkunden, um einen versteckten Schatz zu finden. Hierzu wird vom Server jeweils ein Schatz auf jeder Hälfte der Karte versteckt. Ein Schatz ist hierbei jeweils der KI zugeordnet, die auf dieser Kartenhälfte startet. **Ein Schatz kann nur von der Spielfigur der zugeordneten KI "aufgenommen" und "verwendet" werden.** Um den Schatz zu finden, bewegen beide KIs ihre Spielfigur über die Karte und decken jeweils mit ihrer Spielfigur den versteckten eigenen Schatz und die gegnerische Burg auf. Der Schatz einer KI wird immer auf der Kartenhälfte versteckt, die von der jeweiligen KI erstellt wurde. Bis auf die Schätze, die gegnerische Burg sowie die gegnerische KI in den ersten 10 Spielzügen eines Spiels ist bei Spielbeginn für beide KIs bereits die ganze Karte sichtbar.

**Achtung:** Die ersten 10 Spielzüge eines Spiels hat die vom Server gemeldete Position des Gegners zufällig zu sein und kann sich daher auch auf Ihrer eigenen Spielhälfte (beide KIs befinden sich dadurch auf einer Kartenhälfte) oder einem unerwarteten Ort befinden – zeigen Sie diese trotzdem einfach an (der von der LV-Leitung bereitgestellte Server setzt dies so um bzw. stellt dies sicher). Dies verhindert, dass die gegnerische Burg über die Spielerposition zu leicht gefunden werden kann und erlaubt Ihnen jedoch immer davon ausgehen zu können, dass Sie Informationen über zwei Spieler erhalten (hierdurch müssen Sie keine Fallunterscheidung im Client programmieren, dies erleichtert Ihnen die Entwicklung!).

Sobald eine KI "ihren" Schatz gefunden bzw. aufgedeckt hat, muss sie ihre Spielfigur zu diesem bewegen, um ihn aufnehmen zu können. Das Aufnehmen des Schatzes geschieht automatisch, sobald die jeweilige Spielfigur das Feld mit dem Schatz betritt (hierbei wird der gerade aufgenommene Schatz auch von der Karte entfernt). Sobald der Schatz aufgenommen wurde, muss die KI ihre Spielfigur auf den Teil der Karte bewegen, der von der gegnerischen KI erstellt wurde, um dort die gegnerische Burg zu finden bzw. aufzudecken. Sobald dies gelungen ist, muss sie sich zur gegnerischen Burg bewegen, die Bewacher der Burg mit dem gerade gefundenen Schatz bestechen und so die Burg "übernehmen". Hat die KI dies geschafft, gewinnt diese und damit auch der menschliche Spieler, der diese KI erstellt hat. Das Aufnehmen eines Schatzes und der Kauf der Burg erfolgen automatisch, sobald eine Spielfigur sich auf dem passenden Spielfeld der Karte befindet.

Bereits zu Beginn des Spiels ist das Terrain der Karte beiden KIs vollständig bekannt, jedoch fehlen Informationen über die gegnerische Burg und dem eigenen Schatz, die erst aufgedeckt ("gefunden") werden müssen. Der Schatz der gegnerischen KI ist niemals sichtbar und kann auch nicht aufgenommen werden.

Die Spielaktionen selbst werden rundenbasiert durchgeführt. Daher kann jede KI immer nur eine Aktion setzen (z.B. einen Bewegungsbefehl oder die Übertragung einer Kartenhälfte) und muss danach warten, bis die andere KI ihre Aktion gesetzt hat. Eine KI kann hierbei nicht auf das Setzen einer Aktion verzichten, sondern muss z.B. immer eine Bewegungsaktion durchführen. Eine KI darf aber auch keine Aktion setzen sobald die andere KI an der Reihe ist. Alle Aktionen und Spielfiguren werden hierbei nicht von Menschen repräsentiert und gesteuert, sondern durch künstliche Intelligenzen, welche selbstständig die nächsten Aktionen errechnen.

Um die Spiele für die Zuschauer spannend zu gestalten, wurde festgelegt, dass ein Spiel insgesamt nicht länger als **200 Spielaktionen** dauern darf und eine KI für **jede dieser rundenbasierten Spielaktion nicht mehr als 3 Sekunden** Bedenkzeit zur Berechnung erhält (relevant sind hierbei Spielerbewegung und Kartengenerierung). Sollten diese Bedingungen nicht erfüllt werden, verliert die KI, welche gerade an der Reihe ist, das Spiel automatisch und der zugehörige menschliche Spieler bekommt dies mitgeteilt. Der andere menschliche Spieler wird über seinen Sieg informiert. Das gleiche automatische Vorgehen wird auch angewendet, wenn auf normalem Weg gewonnen wird (eine künstliche Intelligenz erreicht mit ihrem Schatz als Erster die gegnerische Burg).

Die von den KIs bespielte Karte wird bei Beginn des Spiels von beiden beteiligten künstlichen Intelligenzen kooperativ erstellt. Hierzu erstellt jede der beiden KIs zufällig eine Hälfte der finalen Spielkarte (mit je 4 x 8 Feldern für eine endgültige Kartengröße von 8 x 8 oder 4 x 16). Zu beachten ist hierbei, dass die Kartenhälften entweder an den kurzen (Kartenmaße 4 x 16) oder den langen Seiten (Kartenmaße 8 x 8) kombiniert werden – dies kann der Client nicht beeinflussen, sondern wird vom Server durchgeführt. Die Karten sind in Felder aufgeteilt, zwischen denen sich die Spielfiguren bewegen können (waagrecht und senkrecht). Jedes Feld repräsentiert genau eine von drei möglichen Terrainarten, nämlich Wasser, Wiese und Berg.

Burgen und Schätze werden ebenfalls auf eines der Felder gesetzt. Allerdings können die **Burg und der Schatz nur auf Wiesenfeldern und nicht auf demselben Feld platziert werden.** Außerdem darf **Wasser unter keinen Umständen betreten** werden. Bewegt sich die Spielfigur einer KI in ein Wasserfeld, verliert die KI automatisch. Dies passiert auch wenn die Spielfigur mit dem gesetzten Bewegungsbefehl die Karte verlassen würde (z.B., wenn sich die Spielfigur bereits am linken Rand der Karte befindet und sich trotzdem weiter nach links bewegen möchte bzw. bewegt).

Wasser und Wiesen können jeweils mit einer Bewegungsaktion betreten und im Fall von Wiesen auch wieder verlassen werden. Sobald eine Wiese betreten wurde, wird aufgedeckt, ob sich auf dieser Wiese eine Burg oder ein Schatz befindet bzw. wird dieser direkt aufgenommen. Im Gegensatz zu Wiesen benötigen Berge zwei Bewegungsaktionen, um diese (das Bergfeld) zu betreten und zwei zusätzliche, um den Berg wieder zu verlassen. Als Belohnung für diese zusätzlichen Mühen decken Bergfelder dafür Schätze und Burgen in bis zu einem Feld Entfernung rund um den Berg auf. Daher zusätzlich zum Bergfeld selbst werden alle Felder mit einer Entfernung von eins (auch diagonal) aufgedeckt die sich rund um die Spielfigur bzw. den gerade betretenen Berg befinden, sodass Burgen und der der jeweiligen KI zugehörige Schatz darauf sichtbar werden.

Liegen beispielsweise zwei Bergfelder nebeneinander und der Spieler möchte von einem zum anderen überwechseln, werden vier Bewegungsbefehle benötigt (sieben Spielrunden), für zwei nebeneinanderliegende Grasfelder wären 2 Bewegungsbefehle notwendig. Details dazu werden im Unterpunkt "Detaillierte Beschreibung der Bewegungen" beschrieben. Es verbleibt in der Visualisierung des Spielfeldes und den vom Server verteilten Positionsdaten die Spielfigur so lange am ursprünglichen Spielfeld, bis sie vollständig am anderen Berg angekommen ist (nach den vier Bewegungsschritten in Richtung zweites Bergfelds) – dies trifft auch auf alle anderen Feldarten zu.

Während den benötigten Bewegungen kann jederzeit eine neue alternative Richtung eingeschlagen werden. Bereits ausgeführte Bewegungsbefehle verfallen dann, daher wird die Anzahl der notwendigen Bewegungsbefehle von Neuem bestimmt und gezählt. Wurden beispielsweise drei Bewegungsbefehle in eine Richtung geschickt um zwischen zwei Bergen zu wechseln und danach erfolgt ein Bewegungsbefehl in eine andere Richtung gehen die drei bereits gesendeten Bewegungsbefehle "verloren", selbst wenn danach wieder ein Bewegungsbefehl in die alte zuvor verwendete Richtung versendet werden würde.

**Achtung:** Eine Spielfigur kann sich nur horizontal und vertikal zu direkt benachbarten Feldern bewegen, das Überspringen von Feldern ist daher nicht möglich.

Nicht nur das Verhalten der Spielfiguren bzw. KIs ist bestimmten Regeln und Beschränkungen unterworfen, sondern auch die von beiden KIs zu erstellenden Kartenhälften. Die erstellten Karten sollten dabei zufällig generiert und nicht statisch vorgegeben werden. Außerdem muss jeder Terraintyp vorkommen, das heißt **jede Kartenhälfte muss mindestens 3 Bergfelder, 15 Wiesenfelder und 4 Wasserfelder beinhalten**. Weiters dürfen **keine Inseln** generiert werden, daher dürfen Berge oder Wiesen niemals vollständig von Wasser oder Kartengrenzen (oder einer Kombination aus beidem) umschlossen werden. Abschließend gilt, dass **jede Kartengrenze** einer zu generierenden Kartenhälfte nur zu **weniger als der Hälfte aus Wasserfeldern** (maximal 3 Wasserfelder an den langen und 1 Wasserfeld an den kurzen Seiten) bestehen darf, um den Wechsel zwischen beiden Spielkartenhälften nicht zu verhindern. Es gilt außerdem, dass zu jedem Feld einer Kartenhälfte (das nicht aus Wasser besteht) von jedem anderen Feld der Kartenhälfte (welches nicht aus Wasser besteht) ein Weg aus einer Reihe von validen Bewegungsbefehlen generiert werden können muss, um dieses Feld zu betreten. Diese Bedingungen sind zu überprüfen.

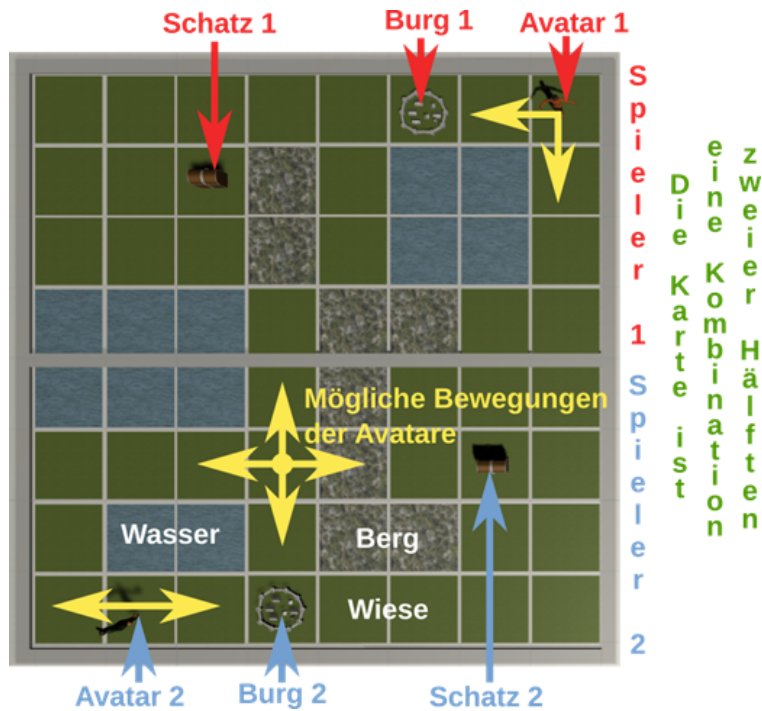
Sollte eine der beiden von den KIs erstellten Kartenhälften gegen diese Bedingungen verstoßen, verliert die KI, welche die verstoßende Kartenhälfte generiert hat. Die andere KI gewinnt hierdurch automatisch.

Während des Spiels müssen die Karte und deren bekannte Eigenschaften auf den Rechnern (den Clients) der beiden menschlichen Spieler visualisiert werden, z.B., gegnerische Spielfiguren (beide Spielfiguren sind immer für beide KIs sichtbar), Burgen, Felder (und deren jeweiliges Terrain) etc. Dies dient dazu, dass menschliche Spieler mit dem Verhalten ihrer künstlichen Intelligenzen mitfeiern können.

Die gesamte Karte hat immer eine **fixe Ausdehnung von 64 Feldern**. Schätze werden, anders als Burgen, nicht während der Generierung der Kartenhälften von den Spielern (genauer, den künstlichen Intelligenzen) platziert, sondern automatisch und zufällig von einer dritten unabhängigen Instanz (dem Server, siehe Teilaufgabe 3). Diese dritte Instanz stellt auch die Einhaltung aller Regeln sicher, indem diese deren Einhaltung überprüft bzw. erzwingt.

### 3.1. Beispielhafte Visualisierung einer 8 x 8 Karte

Eine mögliche Visualisierung der Spielkarte, Burgen etc. ist in der unterhalb angeführten Grafik ersichtlich.





**Achtung:** Da die KI auf der eigenen Burg startet, deren Position aber nicht von Beginn an für den Gegner erkenntlich sein soll, damit diese auch wirklich gesucht werden muss, muss der Server den Gegner die ersten 10 Spielzüge auf einer zufälligen Position „positionieren“. Dargestellt werden außerdem mögliche Bewegungsrichtungen und die Generierung der Karte, basierend auf zwei Kartenhälften.




Weiters stellt die Darstellung oben nur eine mögliche Art dar wie die Karte aufgebaut sein kann (8 x 8), alternativ könnten die Kartenhälften auch an den kurzen Seiten kombiniert werden, um eine 4 x 16 Karte zu erreichen.

### 3.2. Detaillierte Beschreibung der Bewegungen





**Beispiel für den Bewegungsablauf zwischen zwei benachbarten Wiesenfeldern:** Gerade Spielrunden werden hierbei von der gegnerischen KI berechnet bzw. ausgeführt und sind daher nicht angeführt (dies gilt auch für die folgenden Beispiele).

Spielrunde 1	Spielrunde 3
 <p>Wiese 2 Wiese 1</p>	 <p>Wiese 2 Wiese 1</p>
Die Spielfigur der KI befindet sich auf Wiesenfeld 1 und führt eine Bewegungsaktion in Richtung Wiese 2 durch. Runde 1 dient dazu, das Wiesenfeld zu verlassen.	Eine zweite Bewegungsaktion Richtung Wiese 2 wird durchgeführt, um das Wiesenfeld zu betreten. Runde 3 dient dazu, das Wiesenfeld zu betreten.

**Beispiel für den Bewegungsablauf zwischen Wiesenfeld und Bergfeld:** Die Darstellung erfolgt für Wiesenfeld auf Bergfeld, die Bewegung von einem Bergfeld auf ein Wiesenfeld erfolgt analog.

Spielrunde 1	Spielrunde 3	Spielrunde 5
 <p>Wiese Berg</p>	 <p>Wiese Berg</p>	 <p>Wiese Berg</p>
Die Spielfigur der KI befindet sich auf dem Berg und führt eine Bewegungsaktion in Richtung Wiese durch.	Eine zweite Bewegungsaktion Richtung Wiese wird durchgeführt, um den Berg zu verlassen. Runde 1 und 3 dienen dazu, den Berg zu verlassen.	Der Berg wurde nun verlassen. Eine Bewegungsaktion Richtung Wiese wird durchgeführt. Die Spielfigur der KI ist dadurch auf dem Wiesenfeld angekommen. Runde 5 dient dazu die Wiese zu betreten.

**Beispiel für den Bewegungsablauf zwischen zwei benachbarten Bergfeldern:**

Spielrunde 1	Spielrunde 3	Spielrunde 5	Spielrunde 7
 <p>Berg 2 Berg 1</p>	 <p>Berg 2 Berg 1</p>	 <p>Berg 2 Berg 1</p>	 <p>Berg 2 Berg 1</p>
Die Spielfigur der KI befindet sich auf Berg 1 und führt eine Bewegungsaktion in Richtung Berg 2 durch.	Eine zweite Bewegungsaktion Richtung Berg 2 wird durchgeführt, um Berg 1 zu verlassen. Runde 1 und 3 dienen dazu, Berg 1 zu verlassen.	Berg 1 wurde nun verlassen. Da die Spielfigur der KI noch nicht auf dem zweiten Bergfeld angekommen ist, wird diese aber noch auf Berg 1 dargestellt. Eine Bewegungsaktion in Richtung Berg 2 wird durchgeführt.	Eine weitere Bewegungsaktion in Richtung Berg 2 wird durchgeführt. Nach dieser Aktion befindet sich die Spielfigur nun auf Berg 2. Runde 5 und 7 dienen dazu, Berg 2 zu betreten.