

Software Engineering 1

Abgabedokument

Teilaufgabe 1

(Anforderungsanalyse und Planungsphase)

Persönliche Daten, bitte vollständig ausfüllen:

Nachname, Vorname:	Stoykova, Yana
Matrikelnummer:	
E-Mail-Adresse:	
Datum:	25.03.2024

Aufgabe 1: Anforderungsanalyse

Typ der Anforderung: funktional

Anforderung 1

- **Beschreibung:** Registrierung der Clients – Nach Start des Spiels registrieren sich die Clients mit einer eindeutigen SpielID für das Spiel am Server.
- **Bezugsquelle:**
Spielidee, „Nach Start des Clients registrieren sich die KIs für das Spiel am Server...“
Netzwerkprotokoll, „Sobald ein Spiel erstellt wurde und beiden Clients die eindeutige SpielID (siehe oben) bekannt ist können sich die Clients für das neue Spiel registrieren.“

Anforderung 2

- **Beschreibung:** Abfrage des Spielstatus – Sobald der Client den Server nach dem Spielzustand fragt, muss der Server die Karte und alle Spielaktionsinformationen an den Client übermitteln.
- **Bezugsquelle:**
Spielidee, „Wie dies geschieht können die KIs nicht beeinflussen und können/müssen daher vom Server den Zustand der gesamten Karte abfragen.“
Netzwerkprotokoll, „Bei einer Abfrage über den Spielzustand werden vom Server alle für die Clients relevanten Details zum Spiel zurückgeliefert. Dies sind die vollständige oder teilweise Karte (je nachdem ob erst eine oder beide Clients ihre Kartenhälfte zugesandt haben) sowie Informationen, um die Aktionen der Clients abzustimmen (ob ein Client eine Aktion durchführen soll oder ob ein Client gewonnen bzw. verloren hat).“

Anforderung 3

- **Beschreibung:** Schatzverstecken – Wenn der Server die beiden Kartenhälfte zu einer vollständigen Karte zusammenfügt, muss er auf jeder von ihnen einen Schatz verstecken.
- **Bezugsquelle:**
Spielidee, „Hierzu wird je ein Schatz vom Server auf jeder der beiden Kartenhälften versteckt.“

Anforderung 4

- **Beschreibung:** Schrittweise Bewegung – Eine Spielfigur darf sich nur schrittweise zu horizontal und vertikal benachbarte Felder bewegen.
- **Bezugsquelle:**
Spielidee, „Eine Spielfigur kann sich nur horizontal und vertikal zu direkt benachbarten Feldern bewegen, das Überspringen von Feldern ist nicht möglich.“
Spielidee, „Um den Schatz zu finden, bewegen beide KIs ihre Spielfigur über die Karte und decken dabei mit ihrer Spielfigur *schrittweise Kartenfelder auf*.“
Netzwerkprotokoll, „Die Übertragung der Bewegungen erfolgt immer schrittweise,...“

Anforderung 5

- **Beschreibung:** Erreichbarkeit von Feldern – Alle potenziell erreichbare Felder müssen von der Spielfigur betretbar sein.
- **Bezugsquelle:**

Spielidee, „Dürfen keine nicht erreichbaren aber potentiell betretbaren Felder enthalten sein.“

Anforderung 6

- **Beschreibung:** Anzahl Aktionen bei Bergbewegungen – Das Betreten und Verlassen eines Bergs erfordert jeweils 2 Bewegungsaktionen.
- **Bezugsquelle:**

Spielidee, „Im Gegensatz zu Wiesen benötigen Berge zwei Bewegungsaktionen, um diese (das Bergfeld) zu betreten und zwei zusätzliche, um den Berg wieder zu verlassen.“

Typ der Anforderung: nicht funktional

Anforderung 7

- **Beschreibung:** Spielaktionen-Limit – Das Spiel muss maximal 320 Aktionen beinhalten.
- **Bezugsquelle:**

Spielidee, „Um die Spiele für die Zuschauer spannend zu gestalten, wurde festgelegt, dass ein Spiel insgesamt nicht länger als 320 Spielaktionen (und damit 320 Runden) dauern darf.“

Anforderung 8

- **Beschreibung:** Dauer einer Spielaktion – Ein Spielaktion darf maximal 5 Sekunden dauern.
- **Bezugsquelle:**

Spielidee, „Für jede dieser rundenbasierten Spielaktion hat die KI maximal 5 Sekunden Bedenkzeit.“

Anforderung 9

- **Beschreibung:** Zeitspanne zwischen Statusabfragen – Zwischen zwei Spielstatusabfragen vom gleichen Client sollen mindestens 0.4 Sekunden liegen.
- **Bezugsquelle:**

Netzwerkprotokoll, „Um zu verhindern, dass der Server überlastet wird sollte zwischen zwei vom gleichen Client durchgeführten Abfragen zum Spielstatus mindestens eine Zeitspanne von 0,4 Sekunden vergehen.“

Typ der Anforderung: Designbedingung

Anforderung 10

- **Beschreibung:** Spielarchitektur – Die Spielarchitektur folgt einem klassischen Client/Server-Modell.

- **Bezugsquelle:**

Spielidee, „Die *Grobarchitektur* ist damit als klassische Client/Server Architektur vorgegeben.“

Anforderung 11

- **Beschreibung:** Nachrichtenaustausch – Der Nachrichtenaustausch zwischen Client und Server erfolgt über das HTTP Protokoll.

- **Bezugsquelle:**

Netzwerkprotokoll, „Die technologische Basis des Nachrichtenaustauschs stellt eine Restschnittstelle dar, daher es wird das HTTP Protokoll verwendet sowie die zugehörigen Operationen GET und POST.“

Anforderung 12

- **Beschreibung:** Command-line interface – Der Client verwendet command-line interface, um die wechselnde Zustände der Karte für Anwender zu veranschaulichen.

- **Bezugsquelle:**

Spielidee, „Während des Spiels müssen die Karte und deren bekannten Eigenschaften und wichtige Spielzustände von den Clients mittels command-line interface (CLI) für Anwender nachvollziehbar visualisiert werden.“

Aufgabe 2: Anforderungsdokumentation

Dokumentation Anforderung

- **Name:** Anzahl Aktionen bei Bergbewegungen
- **Beschreibung und Priorität:** Das Betreten und Verlassen eines Bergs erfordert jeweils 2 Bewegungsaktionen.
Priorität: Hoch
- **Relevante Anforderungen:**
 - o Erreichbarkeit von Kartenfelder – Alle potenziell erreichbare Felder müssen von der Spielfigur betretbar sein.
 - o Schrittweise Bewegung – Eine Spielfigur darf sich nur schrittweise zu horizontal und vertikal benachbarte Felder bewegen.
 - o Spielaktionen-Limit – Das Spiel muss maximal 320 Aktionen beinhalten.
- **Relevante Business Rules:**
 - o Das Spiel ist rundenbasiert, deshalb müssen die Spieldaten nach jeder Runde aktualisiert werden.
 - o Das Spiel darf maximal 320 Spielaktionen beinhalten.
 - o Der Spieler muss in jeder Runde eine Aktion durchführen, wenn er auf der Reihe ist.
- **Impuls/Ergebnis - Typisches Szenario: Bewegungsablauf zwischen Wiesenfeld und Bergfeld**
Vorbedingungen:
 - o Das Spiel ist bereits von einem Menschen gestartet.
 - o Die Clients haben sich mit einer eindeutigen SpielID für das Spiel am Server registriert.
 - o Die Clients haben ihre SpielerIDs vom Server bekommen.
 - o Die Clients haben 2 Kartenhälften erstellt und diese an den Server gesendet.
 - o Der Server hat die beiden Kartenhälften zu einer vollständigen Karte mit 2 versteckten Schätzen zusammengefügt.
 - o Das Spiel beginnt.
 - o Spieler 1 befindet sich auf ein Wiesenfeld.
 - o Runde 7 beginnt. *(Ich wähle ein Zufallszahl, um das Beispiel zu veranschaulichen.)***Hauptsächlicher Ablauf:**
 - o Impuls: Spieler 1 macht ein Schritt vertikal nach oben („Up“) in Richtung Berg.
 - o Ergebnis: Dieses Schritt dient dazu, das Wiesenfeld zu verlassen. Runde 8 beginnt. Spieler 2 führt eine Aktion durch.
 - o Impuls: Runde 9 beginnt. Spieler 1 macht noch ein Schritt nach oben („Up“) in Richtung Berg.
 - o Ergebnis: Dieses Schritt dient dazu, den Berg zu besteigen. Runde 10 beginnt. Spieler 2 führt eine Aktion durch.
 - o Impuls: Runde 11 beginnt. Spieler 1 macht noch ein Schritt nach oben („Up“) in Richtung Berg.
 - o Ergebnis: Dieses Schritt dient dazu, auf der Spitze des Berges zu stehen.**Nachbedingungen:**
 - o Spieler 1 findet heraus, ob der Schatz auf einem der direkt benachbarten Felder versteckt ist.
 - o Das Spiel dauert so lange, bis einer der Spieler seinen Schatz findet und die gegnerische Burg übernimmt. Er gewinnt.
 - o Der andere Spieler verliert.

- **Impuls/Ergebnis - Alternativszenario: Bewegungsablauf zwischen 2 benachbarte Bergfeldern**

Vorbedingungen:

- o Das Spiel ist bereits von einem Menschen gestartet.
- o Die Clients haben sich mit einer eindeutigen SpielID für das Spiel am Server registriert.
- o Die Clients haben ihre SpielerIDs vom Server bekommen.
- o Die Clients haben 2 Kartenhälften erstellt und diese an den Server gesendet.
- o Der Server hat die beiden Kartenhälften zu einer vollständigen Karte mit 2 versteckten Schätzen zusammengefügt.
- o Das Spiel beginnt.
- o Spieler 1 befindet sich auf ein Bergfeld.
- o Runde 7 beginnt. *(Ich wähle ein Zufallszahl, um das Beispiel zu veranschaulichen.)*

Hauptsächlicher Ablauf:

- o Impuls: Spieler 1 macht ein Schritt vertikal nach oben („Up“) in Richtung Berg.
- o Ergebnis: Dieses Schritt dient dazu, vom Berg herunter zu kommen. Runde 8 beginnt. Spieler 2 führt eine Aktion durch.
- o Impuls: Runde 9 beginnt. Spieler 1 macht noch ein Schritt nach oben („Up“) in Richtung Berg.
- o Ergebnis: Dieses Schritt dient dazu, den Berg zu verlassen. Runde 10 beginnt. Spieler 2 führt eine Aktion durch.
- o Impuls: Runde 11 beginnt. Spieler 1 macht noch ein Schritt nach oben („Up“) in Richtung Berg.
- o Ergebnis: Dieses Schritt dient dazu, den Berg zu besteigen. Runde 12 beginnt. Spieler 2 führt eine Aktion durch.
- o Impuls: Runde 13 beginnt. Spieler 1 macht noch ein Schritt nach oben („Up“) in Richtung Berg.
- o Ergebnis: Dieses Schritt dient dazu, auf der Spitze des Berges zu stehen.

Nachbedingungen:

- o Spieler 1 findet heraus, ob der Schatz auf einem der direkt benachbarten Felder versteckt ist.
- o Das Spiel dauert so lange, bis einer der Spieler seinen Schatz findet und die gegnerische Burg übernimmt. Er gewinnt.
- o Der andere Spieler verliert.

• **Impuls/Ergebnis - Fehlerfall: Wasser beim verlassen eines Bergfeldes**

Vorbedingungen:

- o Das Spiel ist bereits von einem Menschen gestartet.
- o Die Clients haben sich mit einer eindeutigen SpielID für das Spiel am Server registriert.
- o Die Clients haben ihre SpielerIDs vom Server bekommen.
- o Die Clients haben 2 Kartenhälften erstellt und diese an den Server gesendet.
- o Der Server hat die beiden Kartenhälften zu einer vollständigen Karte mit 2 versteckten Schätzen zusammengefügt.
- o Das Spiel beginnt.
- o Spieler 1 befindet sich auf ein Bergfeld.
- o Runde 7 beginnt. *(Ich wähle ein Zufallszahl, um das Beispiel zu veranschaulichen.)*

Hauptsächlicher Ablauf:

- o Impuls: Spieler 1 macht ein Schritt vertikal nach links („Left“).
- o Ergebnis: Dieses Schritt dient dazu, vom Berg herunter zu kommen. Runde 8 beginnt. Spieler 2 führt eine Aktion durch.
- o Impuls: Runde 9 beginnt. Spieler 1 macht noch ein Schritt nach links („Left“).
- o Ergebnis: Dieses Schritt dient dazu, den Berg zu verlassen. Runde 10 beginnt. Spieler 2 führt eine Aktion durch.
- o Impuls: Runde 11 beginnt. Spieler 1 macht noch ein Schritt nach links („Left“).
- o Ergebnis: Spieler 1 hat versucht auf ein Wasserfeld zu bewegen und ist im Wasser ertrunken.

Nachbedingungen:

- o Spieler 2 gewinnt.
- o Spieler 1 verliert.
- o Das Spiel ist vom Server beendet.

• **Benutzergeschichten:**

- o Als Anwender möchte ich, dass das Besteigen und Verlassen eines Berges jeweils 2 Bewegungen erfordert, damit sich das Spiel realistischer anfühlt.

Benutzerschnittstelle: Bewegungsablauf zwischen Wiesenfeld und Bergfeld

Runde 1: Erste Bewegungsaktion Richtung Berg („Up“).

B	W	W	G	G	B	B	G	W	G	Dimension der Spielkarte - 10x10
B	G	B	G	G	G	B	B	G	G	G - Gras
B	G	G	B	G	G	B	G	G	G	B - Berg
B	B	B	W	W	B	G	B	G	G	W - Wasser
G	B	B	G	B	W	G	W	B	G	* - Burg
B	W	G	G	B	B	B	G	W	B	# - Position des Spielers
B	W	B	W	G	G	B	G	B	G	
G	B	B	G	G	B	B	G	B	G	
B	G	W	G	G	B	#G	B	G	G	
G	B	W	*G	G	B	G	G	W	G	

Runde 3: Zweite Bewegungsaktion Richtung Berg („Up“).

B	W	W	G	G	B	B	G	W	G	Dimension der Spielkarte - 10x10
B	G	B	G	G	G	B	B	G	G	G - Gras
B	G	G	B	G	G	B	G	G	G	B - Berg
B	B	B	W	W	B	G	B	G	G	W - Wasser
G	B	B	G	B	W	G	W	B	G	* - Burg
B	W	G	G	B	B	B	G	W	B	# - Position des Spielers
B	W	B	W	G	G	B	G	B	G	
G	B	B	G	G	B	B	G	B	G	
B	G	W	G	G	B	#G	B	G	G	
G	B	W	*G	G	B	G	G	W	G	

Runde 5: Die Spielfigur steht auf dem Berg.

B	W	W	G	G	B	B	G	W	G	Dimension der Spielkarte - 10x10
B	G	B	G	G	G	B	B	G	G	G - Gras
B	G	G	B	G	G	B	G	G	G	B - Berg
B	B	B	W	W	B	G	B	G	G	W - Wasser
G	B	B	G	B	W	G	W	B	G	* - Burg
B	W	G	G	B	B	B	G	W	B	# - Position des Spielers
B	W	B	W	G	G	B	G	B	G	
G	B	B	G	G	B	#B	G	B	G	
B	G	W	G	G	B	G	B	G	G	
G	B	W	*G	G	B	G	G	W	G	

• **Externe Schnittstellen:**

- o Die Übertragung einer Bewegung vom Client zum Server wird ermöglicht durch das HTTP-Protokoll. Die Nachrichten werden in XML Format definiert.
- o Die Visualisierung der aktualisierten Daten passiert mithilfe von Command-line interface.

Aufgabe 3: Architektur entwerfen, modellieren und validieren

Ich lade alle 4 Diagrammen als SVG-Dateien auf GitLab.