

Abgabedokument

Teilaufgabe 1

Anforderungsanalyse und Planungsphase

Nachname, Vorname	Kral Daniel
Matrikelnummer	a11908284
E-Mail-Adresse	a11908284@unet.univie.ac.at
Datum	21. Oktober 2024

1 Aufgabe 1: Anforderungsanalyse

1.1 Funktionale Abhängigkeiten

Anforderung 1

- **Name:** Kartengenerierung
- **Beschreibung:** Der Client muss jeweils die eigene Kartenhälfte durch einen nicht-statischen Algorithmus generieren, um an einem Spiel teilnehmen zu können.
- **Bezugsquelle:** Nach Start des Clients registrieren sich die KIs für das Spiel am Server und erstellen/tauschen danach mit dem Server Kartenhälften aus. [...] Nach dem Kartenhälftenaustausch starten die Spielfiguren jeder KI jeweils [...] auf der von ihr erstellten Kartenhälfte. [...] Kartenhälften müssen zufällig mit Algorithmen generiert und nicht statisch vorgegeben werden.

Anforderung 2

- **Name:** Rundenbasierte Spielaktionen
- **Beschreibung:** Der Client muss darauf warten, dass dieser an der Reihe ist, um eine Spielaktion absenden zu können.

- **Bezugsquelle:** Die Spielaktionen selbst werden rundenbasiert durchgeführt. Jede KI kann 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.

Anforderung 3

- **Name:** Bewegung innerhalb der Spielkarte
- **Beschreibung:** Die Client KI muss garantieren, dass keine Bewegungsaktion generiert wird, die die Spielkarte verlassen würde, um das Spiel nicht automatisch zu verlieren.
- **Bezugsquelle:** Bewegt sich die Spielfigur einer KI in Richtung eines Wasserfeldes, verliert die KI automatisch. Dies passiert auch wenn die Spielfigur von der Karte flüchten würde (z.B., wenn sich die Spielfigur bereits am linken Rand der Karte befindet und sich trotzdem weiter nach links bewegt).

Anforderung 4

- **Name:** Burg auf einem Wiesenfeld
- **Beschreibung:** Der Client muss bei der Kartengenerierung auf ein Spielfeld des Terrain „Wiese“ eine Burg setzen, um nicht die Spielregeln zu verletzen.
- **Bezugsquelle:** Burgen und Schätze werden ebenfalls auf eines der Felder gesetzt. Allerdings können die Burg (vom Client) [...] nur auf Wiesenfeldern [...] platziert werden.

1.2 Nicht-funktionale Abhängigkeiten

Anforderung 5

- **Name:** Schatzsuche
- **Beschreibung:** Die Client KI soll den auf der eigenen Kartenhälfte versteckten Schatz möglichst schnell finden können.
- **Bezugsquelle:** Ziel ist es dabei möglichst schnell den versteckten eigenen Schatz [...] zu finden.

Anforderung 6

- **Name:** Maximale Bedenkzeit
- **Beschreibung:** Die Client KI soll für die Berechnung und Übertragung der eigenen Spielaktion nicht länger als 5 Sekunden brauchen.
- **Bezugsquelle:** Für jede dieser rundenbasierten Spielaktion hat die KI maximal 5 Sekunden Bedenkzeit.

Anforderung 7

- **Name:** Nachvollziehbare Visualisierung
- **Beschreibung:** Die Client-Applikation soll den derzeitigen Spielzustand und Karteninhalt in einer für die benutzende Person nachvollziehbaren Weise darstellen.
- **Bezugsquelle:** 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.

1.3 Designbedingungen

Anforderung 8

- **Name:** Architekturmuster
- **Beschreibung:** Der Client muss über den Spielverlauf mit einem Server kommunizieren, um an einem Spiel teilnehmen zu können.
- **Bezugsquelle:** [...] die KIs von einem Server unterstützt der beide Clients koordiniert, Datenaustausch ermöglicht, als Schiedsrichter auftritt und für Clients Daten speichert, aktualisiert und auswertet. Die Grobarchitektur ist damit als klassische Client/Server Architektur vorgegeben.

2 Aufgabe 2: Anforderungsdokumentation

- **Name:** Kartengenerierung
- **Priorität:** Hoch
- **Beschreibung:** Jeder Client muss jeweils die eigene Kartenhälfte durch einen nicht-statischen Algorithmus generieren, um an einem Spiel teilnehmen zu können.
- **Relevante Anforderungen**
 - Burg auf einem Wiesenfeld – Der Client muss bei der Kartengenerierung auf ein Spielfeld des Terrain „Wiese“ eine Burg setzen, um nicht die Spielregeln zu verletzen.
 - Schatzsuche – Die Client KI soll den auf der eigenen Kartenhälfte versteckten Schatz möglichst schnell finden können.
- **Relevante Business Rules**
 - Die Kartenhälfte muss eine Dimension von 5 x 10 Spielfeldern haben.
 - Die Kartenhälfte muss eine Spielfeld des Terrain „Wiese“ Burg besitzen.
 - Die Kartenhälfte muss aus mindestens 48% Wiesenfelder bestehen.
 - Die Kartenhälfte muss aus mindestens 10% Bergfelder bestehen.
 - Die Kartenhälfte muss aus mindestens 14% Wasserfelder bestehen.
 - Die Kartenhälfte muss an dem Rand aus mindestens 51% betretbare Felder bestehen.
 - Die Kartenhälfte darf keine betretbaren, aber nicht erreichbaren Felder bestehen.
- **Impuls/Ergebnis – Typisches Szenario**
 - **Vorbedingungen:** Der Client wurde von dem menschlichen Spieler gestartet. Darüber hinaus gibt es keine genaue Vorbedingung zu welchem Zeitpunkt die Karte von dem Client während des Spielstarts generiert werden muss.
 - **Hauptsächlicher Ablauf**
 - * **Impuls:** Der Client wird vom menschlichen Spieler gestartet.
 - * **Ergebnis:** Der Client initialisiert die notwendigen Daten für die Spieler-Registrierung.
 - * **Impuls:** Der Client registriert den menschlichen Spieler beim Server.
 - * **Ergebnis:** Der Client wurde erfolgreich registriert und kann nun die Kartenhälfte generieren.
 - * **Impuls:** Der Client beginnt mit der Kartengenerierung.

- * **Ergebnis:** Die Kartenhälfte wurde vom Client generiert.
- **Nachbedingungen:** Die Kartenhälfte wird nun an den Server versendet. Da die Kartenhälfte alle Regeln eingehalten hat, gibt mir der Server eine Erfolgsmeldung zurück und das Spiel kann beginnen, sobald auch der zweite Client eine Karte gesendet hat.
- **Impuls/Ergebnis – Alternativszenario**
 - **Vorbedingungen:** Der Client wurde von dem menschlichen Spieler gestartet. Darüber hinaus gibt es keine genaue Vorbedingung zu welchem Zeitpunkt die Karte von dem Client während des Spielstarts generiert werden muss.
 - **Hauptsächlicher Ablauf**
 - * **Impuls:** Der Client wird vom menschlichen Spieler gestartet.
 - * **Ergebnis:** Der Client initialisiert die notwendigen Daten für die Kartengenerierung.
 - * **Impuls:** Der Client beginnt mit der Kartengenerierung.
 - * **Ergebnis:** Die Kartenhälfte wurde vom Client generiert.
 - **Nachbedingungen:** Die Kartenhälfte ist bereit um über die Server-Schnittstelle versendet zu werden. Zuvor muss jedoch der Client sich beim Server als Spieler registrieren, bevor die Karte an den Server gesendet werden kann.
- **Impuls/Ergebnis – Fehlerfall**
 - **Vorbedingungen:** Der Client wurde von dem menschlichen Spieler gestartet. Darüber hinaus gibt es keine genaue Vorbedingung zu welchem Zeitpunkt die Karte von dem Client während des Spielstarts generiert werden muss.
 - **Hauptsächlicher Ablauf**
 - * **Impuls:** Der Client wird vom menschlichen Spieler gestartet.
 - * **Ergebnis:** Der Client initialisiert die notwendigen Daten für die Spieler-Registrierung.
 - * **Impuls:** Der Client registriert den menschlichen Spieler beim Server.
 - * **Ergebnis:** Der Client wurde erfolgreich registriert und kann nun die Kartenhälfte generieren.
 - * **Impuls:** Der Client beginnt mit der Kartengenerierung.
 - * **Ergebnis:** Die Kartenhälfte wurde vom Client generiert.
 - **Nachbedingungen:** Die Kartenhälfte wird nun an den Server versendet. Da bei der Kartengenerierung jedoch auf der Client-Seite nicht alle Regeln eingehalten wurden, sendet der Server eine Fehlermeldung zurück.

- **Benutzergeschichten:**

- Als Client möchte ich eine regelkonforme Kartenhälfte generieren, um an dem Spiel tatsächlich teilnehmen zu können.
- Als Client KI möchte ich eine von meiner Burg möglichst gut erkundbare Kartenhälfte generieren, um den Schatz auf meiner Kartenhälfte rasch finden zu können.
- Als Client KI möchte ich eine für die Gegner KI möglichst schwer begehbare Kartenhälfte generieren, um das Finden meiner Burg zu erschweren bzw. möglichst lange hinauszuzögern.

- **Benutzerschnittstelle:**

Generating random player map... DONE.

Output:

```

M G M W G W G M M G
W W M M G M G G G M
M M W G G G G M M M
G G M G M G G G W G
M W G M G M G M G G

```

- **Externe Schnittstellen:** Zum Übertragen der zufällig generierten eigenen Kartenhälfte, dient der API Endpunkt `POST /games/<SpielID>/halfmaps` der Server-Implementierung, wobei die `<SpielID>` bei der vorhergegangenen Spielerstellung erlangt wurde. Bei der Anfrage muss die bei der Spielerregistrierung erlangte `SpielerID` in einem `<uniquePlayerID>` XML-Element und die einzelnen Spielfelder der generierten Kartenhälfte in einem zugehörigen `<playerHalfMapNodes>` XML-Element übertragen werden. Jedes `<playerHalfMapNode>` XML-Element besteht aus den XML-Elementen `<X>` und `<Y>` für die Spielfeldposition als ganzzahliges Koordinatenpaar, `<fortPresent>` für ein boolesches Flag, ob sich die Burg auf diesem Spielfeld befindet, und `<terrain>`, um die Terrainbeschaffenheit des Spielfelds zwischen `Grass`, `Mountain` und `Water` zu beschreiben. Diese Gesamtnachricht muss in einem `<playerHalfMap>` XML-Element enthalten sein.

3.1 Klassendiagramm



3.2 Sequenzdiagramm 1

Das folgende Sequenzdiagramm bildet das Szenario 1 ab (Zitat aus der Aufgabenstellung): Beide Clients als fehlerfrei arbeitend annehmen. Planen und Modellieren Sie: Start Ihrer Clientanwendung bis (inklusive) sich Ihr Client bewusst wird die Karte laut Netzwerkdokumentation verschicken zu dürfen.

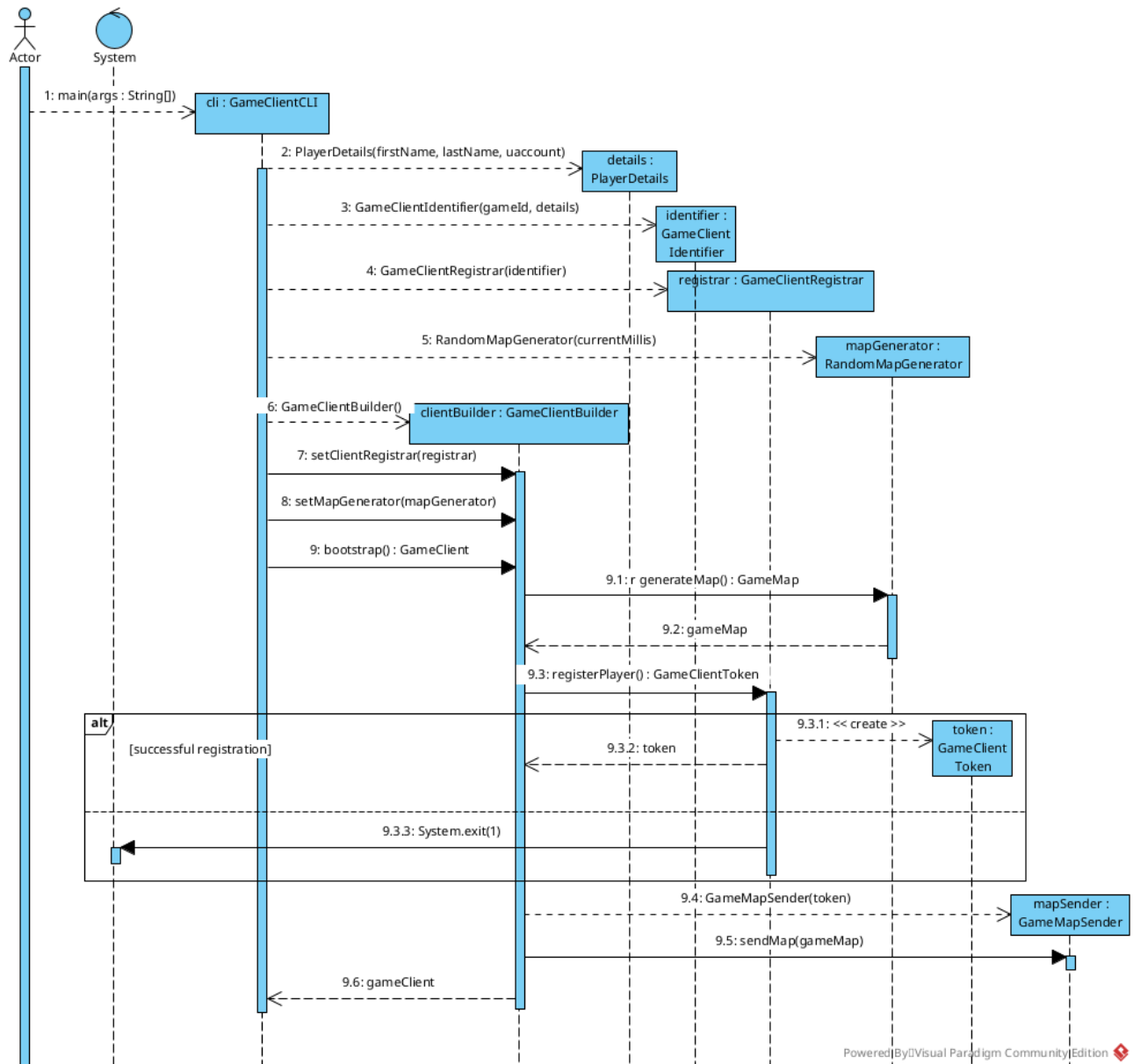


Abbildung 3.2: Sequenzdiagramm von Aufgabenstellung 1

3.3 Sequenzdiagramm 2

Das folgende Sequenzdiagramm bildet das Szenario 2 ab (Zitat aus der Aufgabenstellung): Szenario 2: Annehmen, dass der gegnerische Client gerade erst in ein Wasserfeld gelaufen ist. Planen und Modellieren Sie: Beginnend (inklusive) mit dem direkt daran anschließenden Verhalten Ihres Clients bis zum (inklusive) endgültigen Ende der Clientausführung.

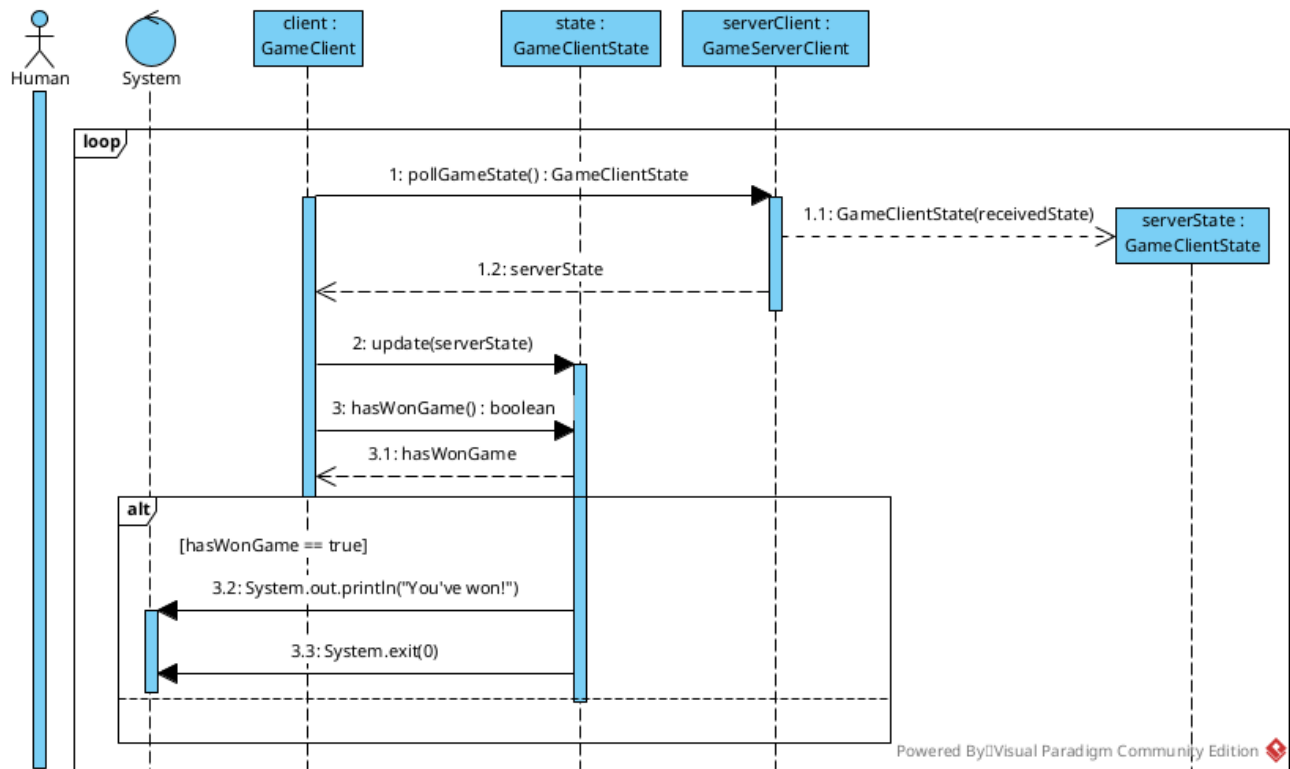


Abbildung 3.3: Sequenzdiagramm von Aufgabenstellung 2

4 Aufgabe 4: Quellen dokumentieren

4.1 Aufgabe 1: Anforderungsanalyse

- **Übernommenen Teile:** Keine übernommen Teile, außer Inspiration von den Beispielen aus Vorlesungsblock 2.
- **Quellen:** Keine externe Quelle genutzt, außer Vorlesungsblock 2.

4.2 Aufgabe 2: Anforderungsdokumentation

- **Übernommenen Teile:** Keine übernommen Teile, außer Inspiration von den Beispielen aus Vorlesungsblock 2.
- **Quellen:** Keine externe Quelle genutzt, außer Vorlesungsblock 2.

4.3 Aufgabe 3: Architektur entwerfen, modellieren und validieren

- **Übernommenen Teile:** Ich habe mich hier an die Beispiele aus den Vorlesungsblöcken 3 und 4 (Entwurf und Implementierung) gehalten, besonders bezüglich der Design Pattern beim Nachschleifen des Klassendiagramms. Gespräche mit Mitstudierenden, ohne dabei etwas zu übernehmen.
- **Quellen:** Gespräche mit Mitstudierenden Gregor und Patrick Davenport. Design Patterns von RefactoringGuru für das Strategy Pattern (<https://refactoring.guru/design-patterns/strategy>).