

AmongAlien 2.0 Architektur

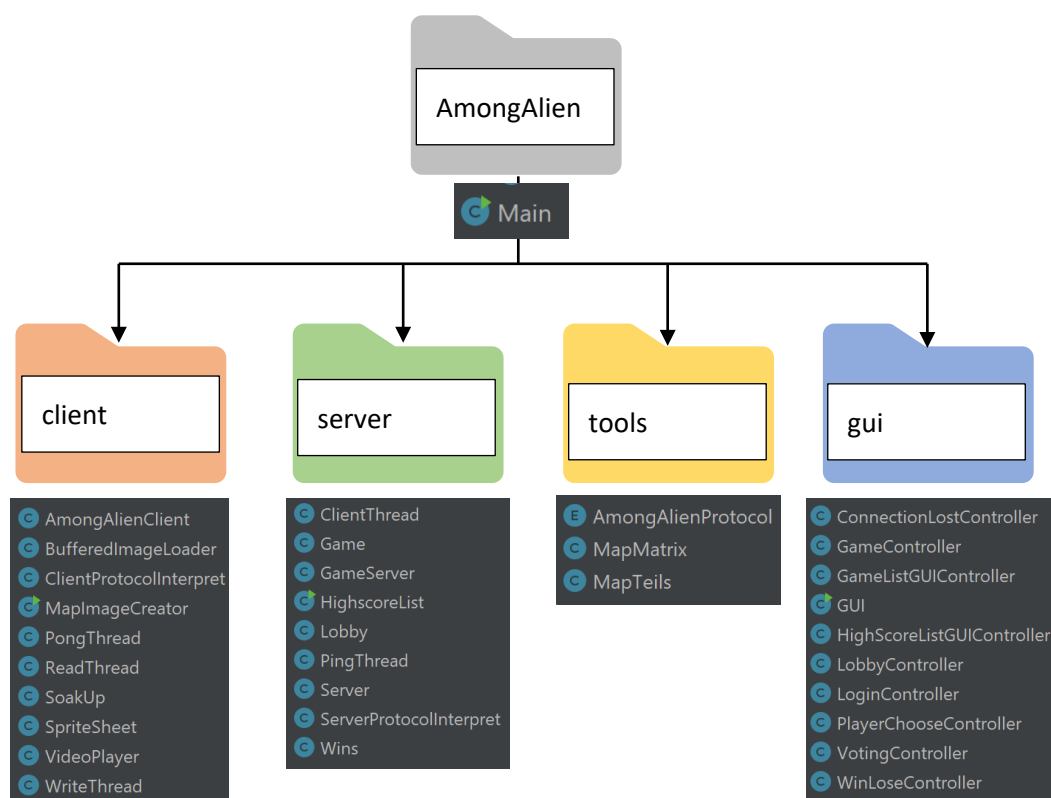
1) Einführung

AmongAlien 2.0 ist ein Computerspiel, welches als Projekt während der Vorlesung «Prorammiertprojekt» im Frühjahrssemester 2022 an der Uni Basel entstand. Das Spiel ist eine Mischung aus AmongUS! und Wehrwölfe und ist dank seinen diversen liebevoll gezeichneten Pixelgraphiken ein wahrer Hingucker. Dieses Dokument gibt einen Einblick in die Architektur des Spiels, was die Einteilung in Pakete, die Netzwerkkommunikation und den Ablauf der Spielprozesse beinhaltet.

2) Pakete

Das Hauptpaket «AmongAlien» ist in die folgenden vier Unterpakete aufgeteilt: client, server, tools und gui. Mit Ausnahme der Main-Klasse sind alle Klassen in diese Unterpakete eingeteilt. Die Main-Klasse ist Teil des AmongAlien-Pakets, da sie für den Programmstart sowie die Wahl zwischen starten von Server oder Client zuständig ist.

Im Folgenden sind die Pakete inklusive deren Klassen graphisch dargestellt und im Anschluss in Worten beschrieben.



1) client-Paket

Das client-Paket beinhaltet alle Klassen, die für die client-Seitige server-client-Kommunikation zuständig sind. Darüber hinaus sind im client-Paket auch Klassen untergebracht, die für die Graphische Realisierung der Spielwelt zuständig sind.

2) server-Paket

Das server-Paket beinhaltet alle Klassen, die für die server-Seitige server-client-Kommunikation zuständig sind. Des Weiteren sind Klassen im server-Paket eingeordnet, die für das Speichern und Berechnen von Spieldaten zuständig sind. Auch werden die Listen von Clients in einer Lobby oder in einem Game durch Klassen im server-Paket verarbeitet.

3) tools-Paket

Das tools-Paket ist das kleinste Paket und umfasst lediglich drei Klassen, die dafür aber von zentraler Bedeutung sind. Zum einen ist das AmongAlienProtocol, das Enum, welches die Kommunikation zwischen Client und Server definiert, dort untergebracht. Aber auch die MapTeils-Klasse und die MapMatrix-Klasse sind im tools-Paket zu finden, da sie der Grundbaustein der Spielwelt darstellen und sowohl von Server als auch von Client verwendet werden. Die MapTeils-Klasse repräsentiert ein einzelnes Quadrätchen der Spielwelt und definiert dessen Eigenschaften, wohingegen die MapMatrix-Klasse die Spielwelt an sich bestehend aus vielen einzelnen MapTeils repräsentiert.

4) gui-Paket

Das gui-Paket beinhaltet alle Klassen, die für die graphische Benutzeroberfläche zuständig sind.

3) Netzwerkkommunikation

Die Netzwerkkommunikation wird durch das Enum AmongAlienProtocol definiert. Das AmongAlienProtocol beinhaltet alle Befehle, mit denen die Interpreten der Server und Clients umgehen können, müssen. Zudem ist im AmongAlienProtocol beschrieben, wie die Befehle zu verwenden sind und was sie bezwecken. Zwei weitere zentrale Klassen, die für die Netzwerkkommunikation von Relevanz sind, sind der ServerProtocolInterpreter und der ClientProtocolInterpreter. Beide Klassen sind dazu gedacht, Befehle entgegenzunehmen und zu interpretieren. Hierfür bestehen beide Klassen je aus einem grossen Switch-Case-Block, der zwischen den einzelnen Befehlen unterscheiden kann und bestimmt, welche Aktion auf Grund dieses Befehls ausgeführt werden muss. Aber auch die Klassen ReadThread und WriteThread spielen eine wesentliche Rolle in der Kommunikation. Der ReadThread ist ein separat laufender Thread, der stetig auf der angegebenen Netzwerkverbindung lauscht, ob eine für ihn bestimmte Nachricht ankommt. Sollte dies der Fall sein, wird sie an den Interpreten geschickt, der dann die entsprechenden Aktionen auslöst. Der WriteThread ist ebenfalls ein separater Thread, der nichts anderes tut, als Nachrichten über die bestehende Verbindung zu versenden.

4) Ablauf von Spielprozessen

Wie ein neues Spiel gestartet, so platziert der Server zufällig eine bestimmte Anzahl an Materialien auf der Spielwelt und erstellt diese. Der Server bestimmt zufällig einen Alien. Die Clients zeigen den Spielern das Charakterauswahlfenster an, welches ebenfalls anzeigt, ob sie Human oder Alien sind. Wählt ein Client einen Charakter aus, so wird dies dem Server mitgeteilt und der Server sendet allen Clients eine aktualisierte Liste von zu Verfügung stehender Charakter. Haben alle Clients einen Charakter gewählt, wechseln die Clients auf Befehl des Servers auf die Spielweltansicht. Alle Clients befinden sich beim Lagerfeuer. Ab dann können die Clients dem Server Standortaktualisierungsanfragen senden, welche der Server prüft und dann allen Clients eine aktualisierte Positionsliste sendet. Die Clients können aber auch Anfragen zum Aufnehmen von Materialien, Abfragen von Tasks oder Erledigen von Tasks senden. Der Client, der Alien ist, pro Spieltag einen Human markieren, wozu er dem Server ebenfalls eine Anfrage senden kann. Nach Ablauf eines Spieltages, sendet der Server den Clients den Befehl, dass sie in das Abstimmungsfenster wechseln sollen. Jeder Client kann dann dem Server eine Wahanfrage senden und somit einen Spieler aus dem Spiel voten. Der Server wertet die Abstimmung aus und

sendet den Clients das Ergebnis. Dann prüft er, ob bereits eine der beiden Gruppen gewonnen hat. Falls ja, befiehlt er den Clients entweder das Gewonnen- oder das Verloren-Fenster anzuzeigen. Im Anschluss passt der Server die High-Score-Liste und die Lobby-Liste an. Wenn noch keine Gewinnergruppe feststeht, startet der Server einen neuen Tag und teilt dies den Clients mit. Ab dann beginnt das Prozedere wieder von vorne.

5) Spielwelt

Die Spielwelt besteht auf vielen einzelnen sogenannten MapTeils, welche in der MapTeils-Klasse repräsentiert sind. Diese MapTeils werden zur MapMatrix zusammengefasst. Ein MapTeil speichert die Information über die Graphik und den Verwendungszweck, den das MapTeil haben soll. Die Graphiken sind alle in der MapTex.png vorhanden und werden durch die SpreadSheet-Klasse entsprechend ausgeschnitten. Jede Graphik besitzt eine eindeutige ID, welche oben links beginnend bei der ID «0» startet und rechts unten mit der ID «99» endet. Alle Graphiken in der MapTex.png wurden von Joël Erbsland mittels Gimp erstellt. Analog zur Graphik hat auch jeder Verwendungszweck eine eindeutige ID. Es werden die folgenden Verwendungszwecke unterschieden:



ID 1 = nicht begehbar

ID 2 = begehbare MapTeil, auf welchem Materialien spawnen können

ID 3 = begehbbares MapTeil, auf welchem ein Task hinterlegt ist

ID 4 = begehbares MapTeil mit Angelrute

ID 5 = begehbare MapTeil, auf welchem keine Materialien spawnen könne

Die Spielwelt kann aus einer .txt Datei eingelesen werden. Die .txt Datei darf ausschliesslich aus Zahl-Doppelpunkt-Zahl Pärchen bestehen. Jede Zeile muss genau gleich viele Zahl-Doppelpunkt-Zahl Pärchen aufweisen, wie die erste. Die erste Zahl der Zahl-Doppelpunkt-Zahl Pärchen repräsentiert die ID der Graphik und die zweite die ID des Verwendungszweckes. Dies ist noch ein Beispiel wie ein Teil der Spielwelt in einer .txt Datei definiert ist.

