

## Aufgabenstellung Programmwurf AI Game Development 2022

Es ist ein in Unity und C# implementiertes Rundenstrategiespiel („Pummelz“) gegeben, in dem ähnlich wie beim Schach zwei Spieler mit blauen und roten Figuren gegeneinander antreten.



*Strategiespiel Pummelz*

Ebenfalls gegeben ist eine rudimentäre Greedy-KI, die Sie als Gegenspieler für Ihre KI verwenden dürfen. Nicht erlaubt ist es allerdings, den Quelltext dieser Greedy-KI ganz oder in Teilen für Ihre KI zu übernehmen. Die von Ihnen entwickelte KI sollte deutlich spielstärker als die Greedy-KI sein und diese in allen zehn gegebenen Beispielszenarien schlagen.

Abseits von der Greedy-KI dürfen sie in Ihrer KI lesend auf alle Funktionen des Spiels zugreifen, insbesondere auf den Spielzustand und die Orakel-Funktionen.

Ihre KI sollte in der Lage sein, auch mit unbekannten Spielfiguren und Szenarien zu spielen. Es ist allerdings erlaubt, Sonderfälle für spezifische Spielfiguren (anhand der unitID) zu implementieren. Es ist hingegen nicht erlaubt, Sonderfälle oder Zugreihenfolgen für spezifische Beispielszenarien (Encounter) zu implementieren (kein Scripting).

Auch ist erlaubt, selbst programmierte Algorithmen und Komponenten aus den Übungen der Vorlesungen wiederzuverwenden.

### 1. Entwurf (3 P):

Spielen Sie Pummelz und formulieren Sie **strategische Grundsätze** basierend auf Ihren Spielerfahrungen.

Basierend auf diesen Spielerfahrungen entwerfen Sie ein generelles **Vorgehen** für Ihre Spiele-KI. Nutzen Sie dazu geeignete Konzepte aus der Vorlesung (OODA-Loops, Decision Trees, etc.). Entwerfen Sie ebenfalls eine **Architektur** in geeigneter

Form (UML, FMC oder ähnlich) und dokumentieren Sie sie in Ihrem Entwurfsdokument.

Das Entwurfsdokument für Aufgabe 1 soll eine maximale Länge von 2-3 Seiten (mit Abbildungen) haben.

0 Bonuspunkte: Nennen Sie Ihren Lieblings-Pummli.

## 2. Ermittlung möglicher Spielzüge (6 P):

Ein Spielzug in Pummelz ist entweder die Bewegung einer Figur auf dem Spielfeld oder der Angriff auf eine andere Figur. Eine dritte Option ist das beenden der eigenen Runde, um den Gegner ziehen zu lassen.

Implementieren Sie eine Funktion, die **mögliche Spielzüge** einer Figur ermittelt. Dabei kann es sinnvoll sein, alle möglichen Züge zu ermitteln oder durch eine Heuristik/Pruning nur vielversprechende Spielzüge auszuwählen.

Implementieren Sie auf Basis dieser Funktion eine KI, die **zufällige Züge** ausführt. Mit dieser KI können Sie testen, ob die von Ihnen gefundenen Züge legal sind.

## 3. Zugauswahl (12 P):

Implementieren Sie basierend auf der in Teil 2 implementierten Funktion den Hauptteil Ihrer KI, die Zugauswahl gemäß der Schnittstelle. Greifen Sie dazu auf Ihren Entwurf zurück.

Achten Sie darauf, dass Ihre KI nur legale Züge verwendet. Ebenfalls verboten ist, dass die KI andere Teile des Spiels manipuliert (also mogelt).

Das Ziel Ihrer KI ist eine möglichst **hohe Spielstärke**. Bedenken Sie, dass die Bewertung und Auswahl von Zügen oder Zugkombinationen entscheidend für die Spielstärke Ihrer KI ist.

Gleichzeitig soll der **Spielfluss** für die Spieler\*innen angenehm schnell sein. Als Richtwert sollte die Berechnung eines Zuges nicht länger als eine Sekunde auf einem handelsüblichen Computer benötigen. Im Idealfall ist die Berechnung des nächsten Zuges bereits abgeschlossen, während die Animation des letzten Zuges noch läuft.

## 4. Iteration (3 Punkte):

Nachdem Sie eine erste Version Ihrer KI fertiggestellt haben, werden Sie mit aller Wahrscheinlichkeit mehrere Iterationen durchlaufen, bis Sie eine hohe Spielstärke erreicht haben.

Dokumentieren Sie diese Iterationen, Ihr Vorgehen und Ihre Fortschritte dazu in Form eines Protokolls mit 2-3 Seiten in Ihrem Entwurfsdokument.

In der Iterationsphase ist es explizit erlaubt, sich mit anderen Teams auszutauschen, beispielsweise durch ein experimentelles Duell beider KIs. Nicht erlaubt ist es hingegen, Code von anderen zu kopieren. Bitte geben Sie an, mit welchen Teams sie zusammengearbeitet haben.

**5. Evaluation (3 Punkte):** Dokumentieren Sie Ihre Ergebnisse mit 2-3 Seiten (inkl. Bilder). Gehen Sie insbesondere auf die Ermittlung der möglichen Züge, ihre Bewertungsfunktion und die Schwierigkeitsgrade ein. Zeigen Sie Ihre KI in interessanten Situationen (gut oder schlecht). Formulieren Sie einen „KI-Trick“ wie die in der Vorlesung gezeigten mit dem aus Ihrer Sicht interessantesten Aspekt ihrer KI. Gehen Sie darauf ein, wie Sie ihre KI getestet und iteriert haben.

**Technische Anforderung (3 Punkte):** Stellen Sie die Ablauffähigkeit sicher. Stellen Sie sicher, dass Sie sich an die Namenskonventionen gehalten haben. Ändern Sie nichts an dem Spiel-Quelltext, der gegeben ist. Stellen Sie sicher, dass Ihre KI nur erlaubte Spielzüge macht. Stellen Sie sicher, dass ihre KI nicht abstürzt oder in eine Endlosschleife geht. Stellen Sie sicher, dass ihre KI nicht exzessiv loggt.

### **Bewertungskriterien**

- 1. Fachliche Bewertung (ca. 50%):** Lösungsqualität und Eleganz sowie Klarheit und Umfang der Betrachtung, Spielstärke der künstlichen Intelligenz, Laufzeit-Performance, Fehlerfreiheit, Nutzung der erworbenen Kenntnisse aus der Vorlesung, Vollständigkeit der Lösung in Bezug auf die Aufgabenstellung
- 2. Dokumentation (ca. 50%):** Dokumentation des Entwurfs und der Iterationen, Codekommentare wie in der Informatik üblich wo notwendig, Qualität der Diagramme, Evaluation als Teil der Dokumentation

### **Abgabe**

Bearbeitung in Gruppen mit jeweils **genau 2 Personen** bis zum **Sonntag, 19.05.2022 16:00 Uhr** einzureichen über das Moodle Lernsystem.

### **Abzugeben sind:**

- 1. Programm:** Quellcode in genau einem C# Quellordner mit Ihrem Teamnamen. Verwenden Sie als Namespace Ihren Teamnamen (also `mg.pummelz.<teamname>`). Im Quellordner befindet sich Ihre KI, die von `MGPumStudentAIPlayerController` abgeleitet ist und die Schnittstelle erfüllt. Die Hauptklasse Ihrer KI ist benannt als `MGPum<Teamname>AIPlayerController`. Ihre KI muss durch Einfügen dieses Ordners in das unmodifizierte Spiel (als Unterordner von `pummelz/ai`) lauffähig sein. Falls Ihre KI zufällige Entscheidungen trifft, darf nur der bereitgestellte Zufallsgenerator verwendet werden. Weitere Anforderungen sind die klare Markierung der Aufgabenteile, Dokumentation (sprechende Bezeichner, Kopfkomentare der Methoden, Quelltextkommentare, wo notwendig), Matrikelnummer statt Name nutzen, achten Sie auf Eleganz und Ausdrucksfähigkeit
- 2. Entwurfs-Dokument** mit Grafiken (achten Sie darauf, dass alles vollständig enthalten ist, z. B. ob alle Diagramme lesbar sind, ob der Umbruch klappt), 2-3 Seiten Entwurf (Aufgabe 1) und 2-3 Seiten Iteration (Aufgabe 4), 2-3 Seiten Evaluation (Aufgabe 5).