TIC TAC TOE AI

Piraccini Yanick, Stadler Raphael

TBZ M254

Contents

Zielsetzung

Die Entscheidung, eine Tic Tac Toe AI im Camunda Modeler zu implementieren, ist auf das Ziel zurückzuführen, dass wir den Prozess einer AI für den User bildlich darstellen wollen, um so das Verständnis der Entscheidungen einer AI dem User näherbringen.

Dies wird nicht einfach, da eine Tic Tac Toe AI schon an sich sehr komplex ist und von viele Algorithmen Gebrauch macht, jedoch wollen wir uns dieser Aufgabe annehmen und so gut wie möglich eine Tic Tac Toe AI in Camunda Modeler nachzubilden.

Technologie

Da wir zuvor noch nie etwas mit dem Camunda Modeler gemacht hatten, mussten wir uns zuerst etwas über den Aufbau informieren, leider gab es dazu einen nicht allzu ausführliche Doku, bei der wir nur das Nötigste erfahren konnten. Ein grosses Hindernis war auch, dass wir die Modeler Sprache 7 gebrauchten und nicht die neuste Version 8. Für das UI haben wir uns für WPF entschieden, da wir uns im Bereich C# schon gut auskennen und so uns kein neues Wissen aneignen mussten. Um die Informationen vom Frontend zum Backend zu schicken, werden wir von ASP.NET Gebrauch machen, da wir schon oft mit dem server-side web-application framework gearbeitet haben.

Aufbau:

- Frontend
    - WPF
    - ASP.NET
- Backend
    - Camunda Modeler
        - Gateway Interface

Planung

Datum	Tätigkeit	Bemerkungen
05.12.2022	Vorstellung LB2 Use Case Definition	Wir entschieden uns für ein Thema und erstellten dazu 3 Use Cases, die wir umsetzen wollten.
12.12.2022	Prozess Definition und Spezifikation, Identifikation der Task- und Subtask Komponenten. User Interface Design und API-Definitionen	Wir haben uns über die Komplexität einer Tic Tac Toe AI informiert und uns Gedanken dazu gemacht, wie wir dies in einem Bpmn umsetzten wollen.
19.12.2022	UI	Wir haben mit WPF ein simples Frontend gemacht, auf dem man Tic Tac Toe spielen kann.
09.01.2023	Camunda Modeler backend	Wir haben im Camunda Modeler angefangen das Backend zu erstellen, jedoch hatten wir grosse Schwierigkeiten da wir zuvor noch nie einen REST API in einem Modeler gemacht haben
16.01.2023	Kommunikation Frontend -> backend	Wir haben mit ASP.NET die Kommunikation vom Frontend zum Backend gelöst, jedoch mussten wir das gleiche auch für das Backend machen. Dort konnten wir dies mit ServiceTasks machen, dies hat uns jedoch etwas Zeit gekostet dies herauszufinden.
23.01.2023	Camunda Modeler backend	Fertigstellen des Backends. Wir haben die Kommunikation zwischen Backend und Frontend fertiggestellt und den Minmax Algorithmus in das Camunda eingebaut.
30.01.2023	Abgabe der Arbeit, Live Demo im Klassenverband.	Steht noch bevor

Use Case

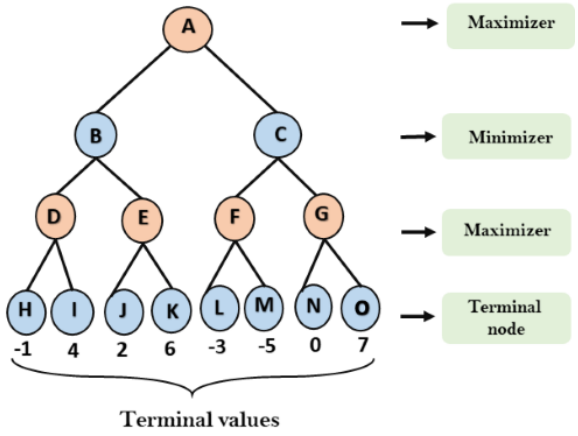
Name	**Spiel starten**	
Akteur	Spieler	
Trigger	Starten der Applikation	
Kurzbeschreibung	Der Spieler startet die Desktop-Applikation und ihm wird das Spielfeld angezeigt auf der, der Computer schon ein X platziert hat.	
Vorbedingungen	Camunda Platform Run muss laufen unter http://localhost:8080, Es muss eine Prozess Instance auf dem Camunda Platform Run vorhanden sein	
Essenzielle Schritte	***Intention der Systemumgebung***	***Reaktion des Systems***
	Anzeigen des Spielfeldes	Laden der XAML-Datei
	Web Api aufstarten	Die ASP.NET Web API wird aufgestartet und schickt einen request, um den Camunda Prozess zu starten.
	Starten des Backend Prozesses	Es wird ein POST request an das Backend geschickt, in dem der Prozess gestartet wird
	Platzieren des ersten X durch Computer	Nach dem Starten wird ein Int mit dem Standort des ersten X's an das Frontend zurückgesendet
Ausnahmefälle	keine	
Nachbedingung	keine	
Zeitverhalten	keine	
Verfügbarkeit	7/24	
Folge Use Case	Der Spieler kann sein erstes O platzieren	

Name	Spieler setzt O	
Akteur	Spieler	
Trigger	Drücken eines Beliebigen Felds	
Kurzbeschreibung	Der Spieler hat nun die Möglichkeit ein O auf dem Spielfeld zu setzten	
Vorbedingungen	Applikation gestartet, Camunda Platform Run muss laufen unter http://localhost:8080	
Essenzielle Schritte	***Intention der Systemumgebung***	***Reaktion des Systems***
	Send POST Request	Es wird ein POST request an die Camunda REST API gesendet
	Complete UserTask	Wenn alle nötigen Informationen vorhanden sind wird er UserTask abgeschlossen und der Workflow kann weitterlaufen
Ausnahmefälle	keine	
Nachbedingung	keine	
Zeitverhalten	sofort	
Verfügbarkeit	7/24	
Folge Use Case	Backend Check für bester Zug mäglich	

Name	**Wählen des Beginners**	
Akteur	Spieler	
Trigger	Auswählen des Beginners	
Kurzbeschreibung	Der Spieler hat nun die Möglichkeit auszuwählen wer starten darf.	
Vorbedingungen	Applikation gestartet, Camunda Platform Run muss laufen unter http://localhost:8080	
Essenzielle Schritte	***Intention der Systemumgebung***	***Reaktion des Systems***
	UI-Fenster mit Auswahl	Der User hat die kann zwischen Computer und Player auswählen
	Press Computer	Der Computer fängt an und es wird automatisch ein X auf dem Spielfeld platziert
	Press Player	Der Spieler kann auf dem Spielfeld ein O platzieren
Ausnahmefälle	keine	
Nachbedingung	Je nach Auswahl fängt der Computer an oder der Spieler	
Zeitverhalten	sofort	
Verfügbarkeit	7/24	
Folge Use Case	Fortfahren des Spiels	

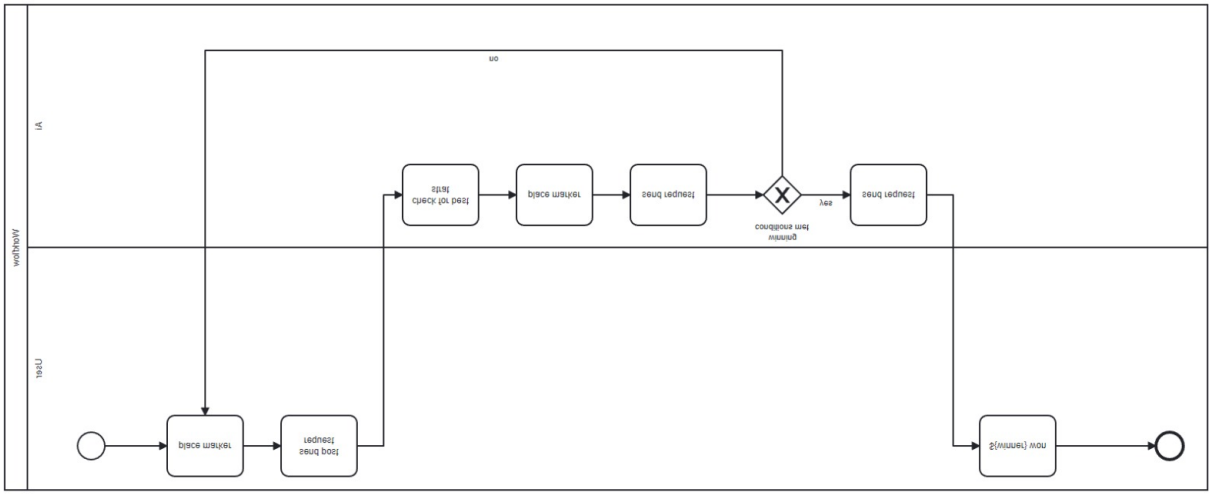
Umsetzung

Vor der Umsetzung haben wir uns zuerst Beispiele in anderen Programmiersprachen angesehen, um so die Logik hinter einer Tic Tac Toe AI zu verstehen. Wir haben herausgefunden, dass man den bestmöglichen nächsten Zug mit dem Minmax Algorithmus herausfinden kann. Zuerst wollten wir den Algorithmus in einem DMN umsetzten, jedoch bemerkten wir schnell das dies nicht möglich ist und haben uns entschieden das Ganze in einem ScriptTask umzusetzen.

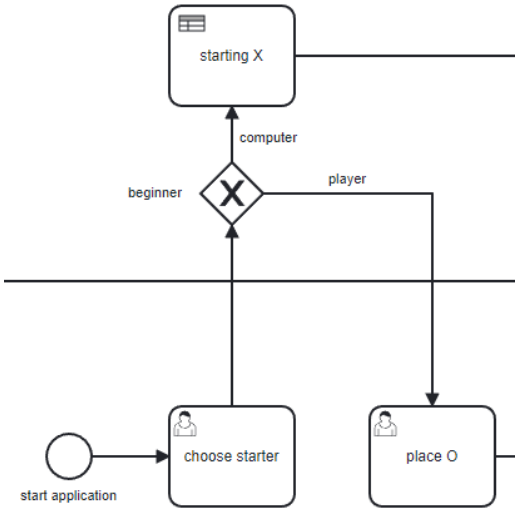


Minmax ist ein Algorithmus für Spieltheorie, der verwendet wird, um den besten Zug in einem Zwei-Personen-Spiel wie Schach, Tic Tac Toe oder Connect Four zu finden. Der Algorithmus simuliert alle möglichen Züge und Ausgänge des Spiels und bewertet sie, indem er den Nutzen für beide Spieler berechnet. Der Algorithmus wählt den Zug mit dem höchsten Nutzen für den aktiven Spieler und dem niedrigsten Nutzen für den gegnerischen Spieler. Auf diese Weise wird der beste Zug ermittelt, der dem aktiven Spieler den größtmöglichen Vorteil verschafft, ohne dabei den Gegner zu begünstigen.

Danach haben wir einen Entwurf unseres Bpmn Modelle gemacht, um so einen Eindruck zu bekommen, wie das Projekt aufgebaut sein sollte. Wie man am Ende sehen wird, wurde es um einiges verändert, da nicht alles lief, wie wir es geplant hatten.

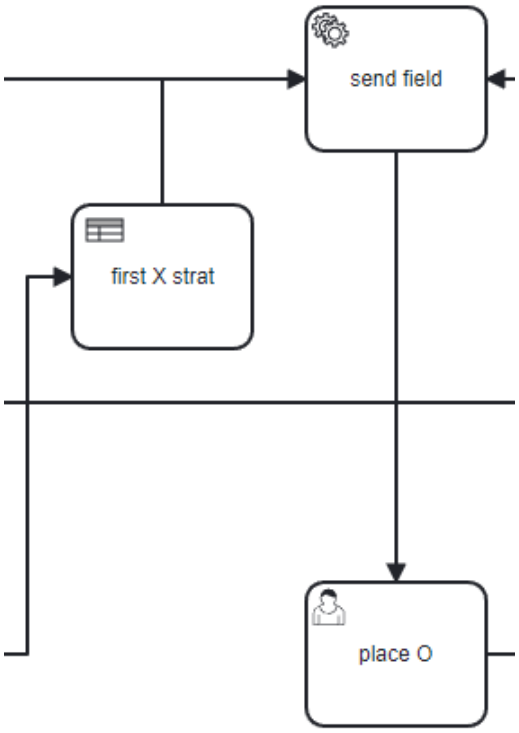


Das Endprodukt war schlussendlich eine Zusammensetzung aus Strategien wie man Tic Tac Toe gewinnt und aus dem Minmax Algorithmus.



Im ersten Schritt wird die Applikation gestartet. Der Spieler bekommt ein Fenster angezeigt, auf dem er wählen kann, wer startet. Beim Auswählen eines der zwei Optionen wird ein Post Request an die Camunda REST API gesendet, den denn UserTask «choose starter» completed und ihm die Variable «beginner» und «randomInt» mitgibt.

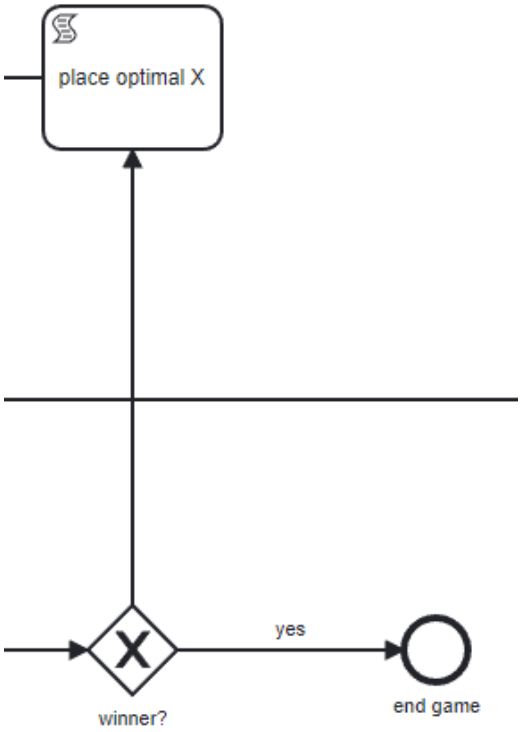
Danach kommt ein Exclusive Gatway bei dem je nach «beginner» zuerst ein X gesetzt wird oder der Spieler wieder zum Zug kommt. Wenn «beginner» == computer ist, dann setzt es in ein zufälliges Eckfeld ein X in einem DMN wird durch den «randomInt» eine zufällige Zahle zurückgegeben, an diesem Ort im Spielfeld wird dann auch das erste X platziert.



Im nächsten schritt wird mit einem Dmn das Bestmögliche X für den Computer platziert, dies haben wir auf der Internetseite wiki-how herausgefunden, um so die grösste Gewinnchance für den Computer zu erstellen. Danach werden die zwei Pfade wieder zu einem zusammen geführt, in dem sie einen POST request an unsere Frontend-API senden. Dies haben wir mit einem ServiceTask gemacht, den wir als http-connetcor definiert haben. Dass dieser funktionieren kann, mussten wir folgende jars unter /configuration/userlib hinzufügen:camunda-connect-connectors-all-1.4.0.jar

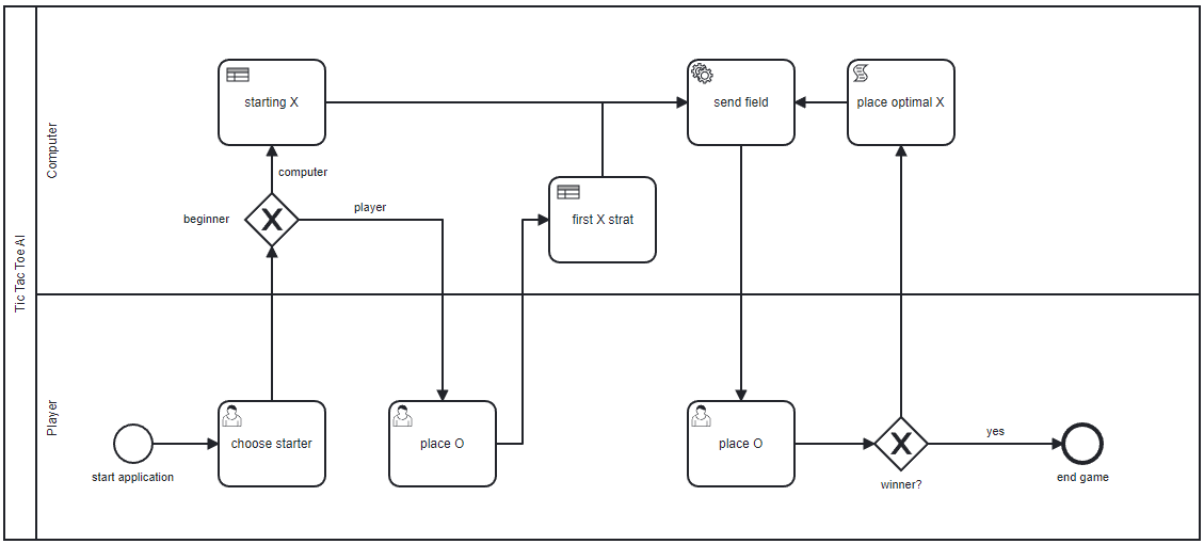
- camunda-engine-plugin-connect-7.13.0.jar
- camunda-connect-core-1.4.0.jar

Mit diesen jars wird dann der Nummer des neu besetzten Felds an das Frontend geschickt, damit es dies dann auf dem UI anzeigen kann. Nachdem wird dem Spieler wieder die Möglichkeit gegeben, ein O auf dem Spielfeld zu setzten.



Im letzten Schritt wird überprüft, ob es schon ein Gewinner gibt, dies wird vom Frontend an das Backend geschickt. Wenn die «winner» Variable true ist, dann ist das Spiel zu Ende und es wird im UI angezeigt, wer gewonnen hat. Wenn «winner» false ist, dann kommt es zu einem ScriptTask in dem der Minmax Algorithmus ausgeführt wird und den aktuell besten Zug zurückgibt.

Am Ende sah unser BPMN etwas anders aus als wir es geplant hatten, jedoch wurde die Grundidee beibehalten.



Probleme

Anfangs hatten wir grosse Schwierigkeiten, da wir mit dem Camunda Modeler nicht ganz klargekommen sind. Wir wussten nicht, wie wir ein POST Request im Backend machen, um so mit dem Frontend kommunizieren zu können. Da wir auch im Frontend eine REST API hatten, mussten wir die erhaltenen Daten vom einten Projekt zum anderen transferieren. Dies hat uns einige Zeit gekostet, aber wir haben uns schlussendlich dazu entschieden, die Feldnummer in ein Text File zu schreiben und es im anderen Projekt zu lesen.

Wir hatten auch sonst viele kleine Bugs, die uns viel Zeit gekostet haben.

Fazit

Camunda haben wir zuvor beide noch nicht gekannt, anfangs hatten wir auch Schwierigkeiten, denn sinn hinter Camunda zu verstehen. Mit der Zeit sahen wir die Vorteile, die Camunda einem bringt, aber wir sind immer noch der gleichen Einstellung, dass es bessere Lösungen gibt als Camunda vor allem für eine Tic Tac Toe AI.

Selbstbewertung

Bereich	Kriterien	Punkte
Formaler Rahmen	Aufmachung, Vollständigkeit, Rechtschreibung, Titel, Namen, Thema	3/3
Zielsetzung	Nachvollziehbarkeit	3/3
Textliche Beschreibung	Nachvollziehbarkeit, Verständlichkeit, Vollständig	4/5
Grafische Darstellung	Korrektheit, Umsetzung, Nachvollziehbarkeit, Dokumentation, Planung und Abweichungen	8/10
Umsetzung	Strukturierte Umsetzung und Funktionsfähigkeit (Präsentation Layer, Interface Layer, Workflow Layer) Lauffähig in der Camunda BPM Plattform	7/10
Komplexität der Implementierung	Abtrennung Kundenspezifisch, Standard Funktionen	6/6
Präsentation	Ablauf, Verständlichkeit, Darstellung	4/4
Note	5.5	35/41