

Hochschule Luzern
Digital Ideation Fokus Informatik
Modul: Studio Game1 Extendet
Dozierende: Dragica Kahlina, Sebastian Hollstein

Behavior-Tree und der Hund

Dominik Maag
+41 78 713 14 36
dominik.maag@stud.hslu.ch
Felsenastrasse 63
8200 Schaffhausen



Einleitung

In diversen Videospielen sind die NPCs (Non-Player Charakter) ein Bestandteil. Jeder NPC hat auch sein eigenes Verhalten. Dieses Verhalten kann in einem sogenannten Behavior-Tree dargestellt und umgesetzt werden. In diesem Paper wird dabei auf die Thematik eingegangen, wie solch ein Behavior-Tree aufgebaut ist und wie dieser funktioniert. Als Verhaltensvorlage wird das mögliche Behavior eines Hundes genommen.

Fragestellung

Folgende Frage soll beantwortet werden: Wie kann das Verhalten von einem Hund in einem Behavior-Tree dargestellt werden? Dabei wird einerseits genauer untersucht, was ein Behavior-Tree ist und andererseits, wie solch ein Tree funktioniert.

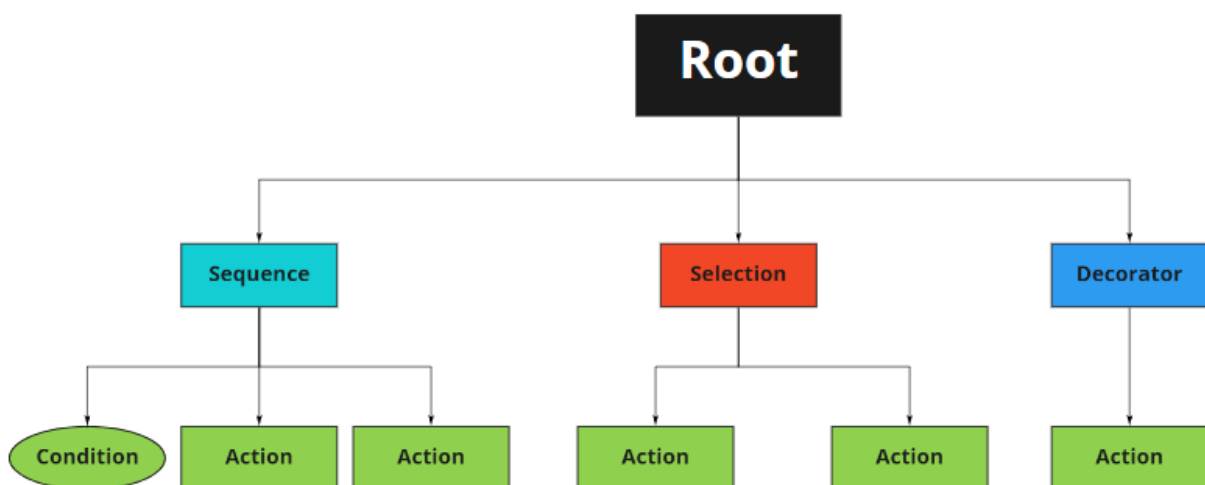
Was ist ein Behavior-Tree?

Ein Behavior-Tree bietet die Möglichkeit, das Verhalten und Handeln zwischen verschiedenen Aufgaben und Zuständen strukturiert darzustellen (Petter Ögren, 2019a). Für dies steht eine Sammlung oder auch Set von Aktionen zur Verfügung (Petter Ögren, 2019a). Nach Peter Ögren hat ein Behavior-Tree diverse Vorteile, wie etwa die hierarchische Struktur oder die Modularität. Ebenso hebt er die grafische Darstellung hierbei hervor.

Bestandteile eines Behavior Trees

Bestandteil	Beschreibung
Node	Nodes oder auch Knoten stellen einen Status oder eine Aktion in einem Behavior-Tree dar
Branches	Sind die Verbindung zwischen den einzelnen Knoten
Root	Der Eingangs-Node. Dieser Node initialisiert den Behavior-Tree
Kinder	Ein Nachfolge-Node wird auch Kind genannt. So hat z.B. der Node Root 3 Kinder.
Decorator	Decorator werden dadurch definiert, dass sie nur ein Kind haben.
Composite	Composite sind Nodes, welche mehrere Kinder haben.
Selection	Bei einer Selection wird nach einem bestimmten Kriterium der nächste Node ausgewählt.
Sequence	Alle Kinder werden nach einer bestimmten Reihenfolge besucht.
Leafs	Leafs sind Nodes welche keine Kinder haben. Sie sind in einem Behavior-Tree Aktionen oder Condition/Bedingung

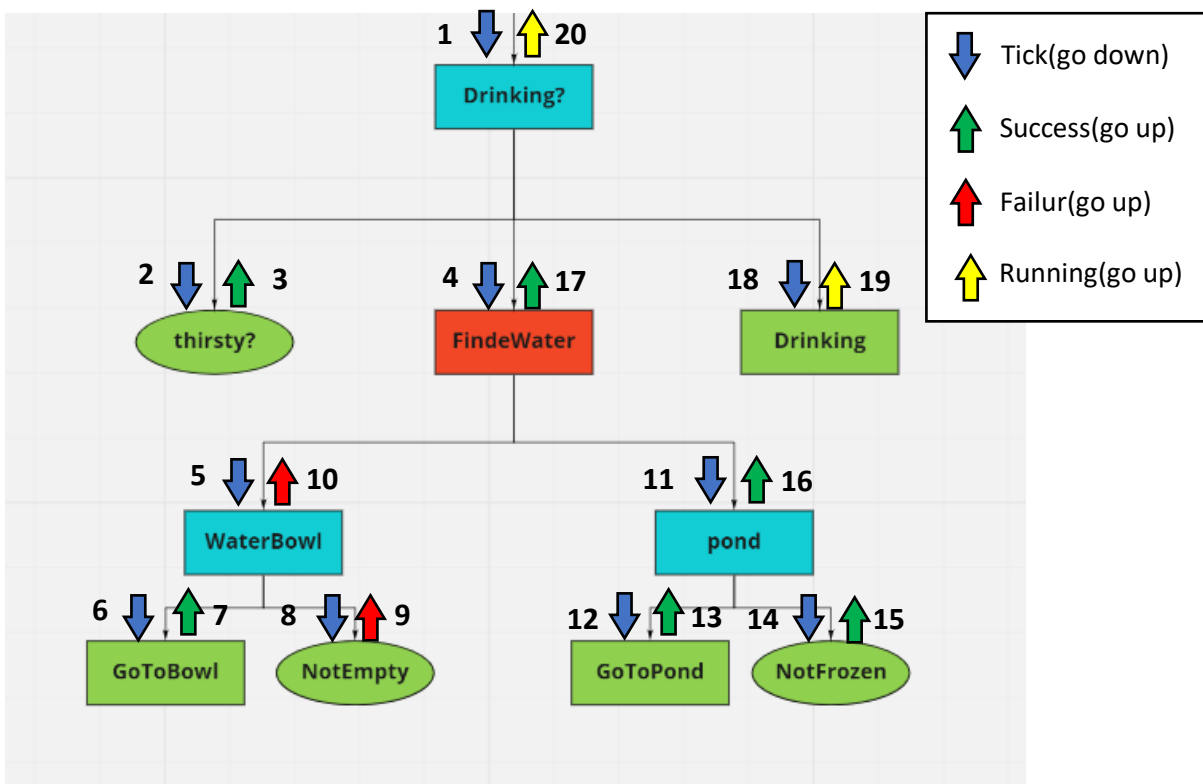
(vgl. Holistic3d, 2017) und (vgl. Petter Ögren, 2019a)



Wie funktioniert ein Behavior Tree?

Die Navigation in einem Behavior-Tree geschieht nach einem bestimmten Schema. Mit sogenannten Ticks bewegt man sich immer weiter nach unten. Bei einer Aktion oder Bedingung wird einer von drei Rückgabewerten zurückgegeben. Success, Failur und Running. Ist nun eine Aktion nicht korrekt Ausgeführt worden oder eine Bedingung falsch, gibt diese ein Failure an den Parent Node zurück. Je nach dem was das Parent ist wird anders verfahren. Sequence kann man mit einem «und» vergleichen, bei welchem alle Aktionen und Bedingungen ein Success ergeben müssen, um einen Success an den nächstoberen Node zurückzugeben. Bei einer Selection kann die Verbindung mit «oder» gemacht werden. Sobald eine Aktion oder Bedingung Success zurückgibt, wird auch die Selection ein Success zurückgeben. Gibt es jedoch kein Success, so gibt die Selection einen Failur zurück. So bewegt man sich mit einem Tick hinunter im Baum und mit einem Success, Failur odr Running wieder nach oben.

Unten sieht man anhand eines Hundes, der etwas Trinken will, wie solch ein Behavior-Tree funktionieren könnte und sollte.



1.	Tick. Der Hund geht was trinken	11	Tick. Teich prüfen
2.	Tick. Ist der Hund Durstig?	12	Tick. Zum Teich gehen
3.	Success. Der Hund ist durstig	13	Success. Beim Teich
4.	Tick. Wasser Suchen	14	Tick. Ist der Teich nicht gefroren?
5.	Tick. Napf prüfen.	15	Success. Der Teich ist nicht gefroren
6.	Tick. Zum Napf gehen.	16	Success. Wasser im Teich trinkbar
7.	Success. Beim Napf.	17	Success. Wasser gefunden
8.	Tick. Ist der Napf voll?	18	Tick. Trinken
9.	Failure. Napf ist Leer	19	Running. Trinken wird ausgeführt
10.	Failure. Kein Wasser im Napf.	20	Running. Der Hund trinkt

(vgl. Petter Ögren, 2019b)

Behavior-Tree eines Hundes

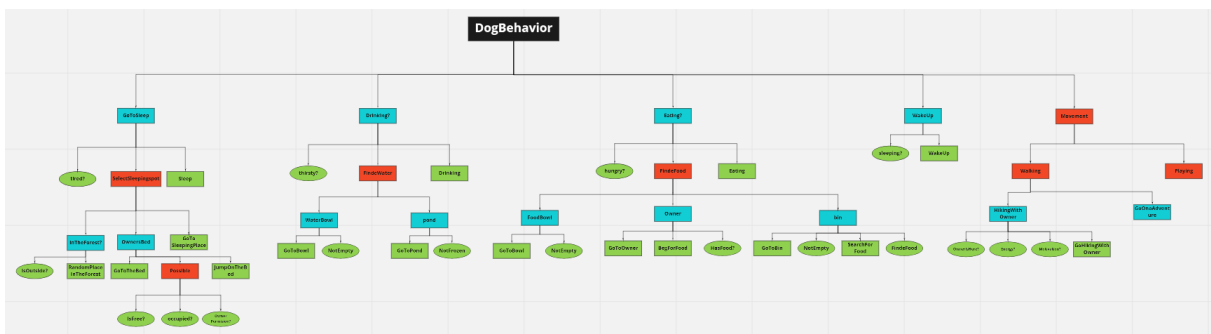
Nach dem ich mich mehr mit dem Behavior-Tree und dessen Funktionsweise auseinandergesetzt habe, fing ich an, den Behaviortree für einen Hund visuell darzustellen. Hierbei ist mir aufgefallen, dass man zuerst wissen muss, wie das Grundverhalten eines Hundes überhaupt ist. Für dies habe ich mir ein Paar NPC Hunde aus verschiedenen Videospiele angeschaut und analysiert.

Hunde in Videospielen

Für die Analyse habe ich den Hund von Fable 2, Dogmeat aus Fallout 4 und die Hunde aus the Legend of Zelda: Breath of the Wild genommen. Diese verhalten sich sehr unterschiedlich, zeigen aber gewisse Gemeinsamkeiten im Verhalten auf. So reagiert zum Beispiel der Hund aus Fabel 2 auf die Situation, wenn man eine Höhle betritt. Hierbei fängt er an zu wimmern und zieht den Schwanz ein, weil er Angst hat. Dieser Zustand bleibt so lange, bis man wieder aus der Höhle hinausgeht oder als Spieler den Hund wieder aufmuntert. Eine andere Situation ist, in der der Hund auf Objekte reagiert, wenn Gegner in der Nähe sind. Dann beginnt der Hund zu knurren. Ein weiteres Verhalten bei Hunden kann aufgrund eines Gegenstandes sein, den er gefunden hat. Dogmeat aus Fallout 4 bellt, wenn er einen Gegenstand an einem Ort gefunden hat, oder bringt diesen sogar zum Spieler. Während Dogmeat und der Hund aus Fabel nur zwei Begleiter sind, so sind die Hunde aus The Legend of Zelda: Breath of the Wild reine NPC, die ihren eignen Ablauf haben. So verhalten sie sich ähnlich wie ein Hund, welcher auf dem Bauernhof lebt. Ein Beispiel dafür ist, wenn Spielende in die Nähe des Hundes kommen, dieser dann neugierig auf einen zu kommt und ihnen folgt. Legen die Spielenden nun etwas Fleisch auf den Boden, so geht es einen kurzen Moment, bis der Hund schlussendlich das Fleisch frisst.

Backward-Chaining

Bei der Umsetzung des Behavior-Trees habe ich zuerst den Tree bis zu einem gewissen Grad visuell dargestellt. Dabei bin ich nach der Methode des Backward-Chaining vorgegangen. Bei dieser Methode wird versucht vom gewünschten Ziel, welches man erreichen will, rückwärts immer mehr aufgespalten in einzelne Teilschritte (vgl. Petter Ögren, 2019b).



Abschluss

Ein Behavior-Tree bietet die Möglichkeit das Verhalten eines NPC darzustellen und in einem Game zu implementieren. Anhand des Versuchs mit dem Verhalten eines Hundes, ist ersichtlich, dass je detaillierter das Verhalten in einem Behavior-Tree umgesetzt wird, dieser umso komplexer wird. So sollte man sich immer die Frage stellen, wie detailliert das Verhaltens eines NPC sein sollte und ob es so detailliert sein muss. Interessant wäre zu schauen, wie sich der Behavior-Tree verändern würde, wenn der Fokus nur auf ein spezifisches Verhalten reduziert würde.

Methode für die Planung und Umsetzung

Mit dem Backward-Chaining hat man nun eine Methode, wie man den Baum aufbauen und umsetzen kann. Dazu gibt es noch weitere Methoden, welche hier noch nicht angesprochen werden. Welche Methoden es noch gäbe, evtl. besser wären, und für welche Aspekte diese am ehesten gebraucht werden könnten, wäre ein Punkt, welcher noch genauer untersucht werden kann.

Einflüsse auf den Behavior-Tree

An dem Behavior-Tree des Hundes wurde auch ersichtlich, dass nicht nur der Spieler Einfluss auf das Verhalten eines NPCs hat. Auch die Umwelt und andere NPC beeinflussen das Verhalten. Unter anderem besteht die Vermutung, dass das Game-Genre schon eine gewisse Grundlage für das Verhalten eines NPCs erwartet und somit vorgibt, in welche Richtung der Behavior-Tree sich entwickeln muss. Eine offene Frage hierbei wäre sicherlich, inwiefern das Genre eines Videospieles das Verhalten eines NPCs vorgibt und somit auch das Design eines Behavior-Trees festlegt.

Funktionsweise

Eine Überraschung war sicher, die Art wie ein Behavior-Tree funktioniert. Dies hatte zur Folge, dass die ganze Herangehensweise überdenkt, werden musste und ein neuer Ansatz für den Behavior-Tree gefunden werden musste. Unter anderem war auch das Backward-Chaining etwas Neues und Ungewohntes. Nach ein paar Versuchen erkannte man recht schnell die Vorteile dieser Methode.

Literaturverzeichnis

Holistic3d. (2017, Oktober 16). *Introduction to Behaviour Trees*.

<https://www.youtube.com/watch?v=uq8hnnkAxsw>

Petter Ögren. (2019a, August 9). *What is a Behavior Tree and How do they work? (BT intro part 1)*.

<https://www.youtube.com/watch?v=DCZJUvTQV5Q>

Petter Ögren. (2019b, Oktober 28). *How to create Behavior Trees using Backward Chaining (BT intro*

part 2). <https://www.youtube.com/watch?v=dB7ZSz890cw>