A picture containing tool

Description automatically generatedSoenen Tristan

What is the impact of switching from a Finite State Machine to a Goal Oriented Action Planning (GOAP) in a simulation game?

Supervisor: Vandenberghe Pieter-Jan

Coach: Mores Alexandra

Graduation Work 2024-2025

Digital Arts and Entertainment

Howest.be

A close up of a card

Description automatically generated

Contents

[Abstract & Key words 3](#_Toc187009336)

[Preface 4](#_Toc187009337)

[List of Figures 5](#_Toc187009338)

[Introduction 6](#_Toc187009339)

[Literature Study / Theoretical Framework 7](#_Toc187009340)

[Finite State Machines 7](#_Toc187009341)

[What is a Finite State machine? 7](#_Toc187009342)

[How does a finite state machine work? 7](#_Toc187009343)

[Where are finite state machines used? 8](#_Toc187009344)

[STate machines are usefull when: 8](#_Toc187009345)

[Goal Oriented Action Planning (GOAP) 8](#_Toc187009346)

[What is Goal Oriented Action Planning (GOAP)? 8](#_Toc187009347)

[How does Goal Oriented Action Planning work? 8](#_Toc187009348)

[Where is Goal Oriented Action Planning used? 9](#_Toc187009349)

[GOAl oriented action planning is usefull when: 9](#_Toc187009350)

[The comparison finite state machines vs Goal Oriented Action Planning 10](#_Toc187009351)

[Differences 10](#_Toc187009352)

[Research 11](#_Toc187009353)

[Research Question: 11](#_Toc187009354)

[Hypothesises: 11](#_Toc187009355)

[Methodology: 12](#_Toc187009356)

[Experiment Set-Up: 12](#_Toc187009357)

[Agent Behaviors 12](#_Toc187009358)

[Hardware: 13](#_Toc187009359)

[Data Collection: 13](#_Toc187009360)

[Checking agent behavior: 14](#_Toc187009361)

[Experiment: 15](#_Toc187009362)

[Testing the level for performance: 15](#_Toc187009363)

[Data Collection: 15](#_Toc187009364)

[Prediction: 15](#_Toc187009365)

[Answer: 15](#_Toc187009366)

[Testing Agent Performance: 16](#_Toc187009367)

[Data Collection: 16](#_Toc187009368)

[Prediction: 16](#_Toc187009369)

[Result: 17](#_Toc187009370)

[Answer: 17](#_Toc187009371)

[Discussion 18](#_Toc187009372)

[Conclusion 19](#_Toc187009373)

[Future work 20](#_Toc187009374)

[Critical Reflection 21](#_Toc187009375)

[References 22](#_Toc187009376)

[Acknowledgements 23](#_Toc187009377)

[Appendices 24](#_Toc187009378)

# Abstract & Key words

**An abstract explains the outline of the paper concisely (the methods, results, etc.). Maximum length of 250 words, preferably both in English and Dutch.**

Key Words: Finite State machine, Goal Oriented Action Planning

# Preface

***A preface is a statement of the author's reasons for undertaking the work and may include personal comments that are not directly relevant to other sections of the thesis or dissertation.* No word count limit.**

# List of Figures

**The list of figures lists the figures in the order in which they appear throughout the thesis. They may be numbered sequentially, or be subdivided following the chapters in which they appear.**

Figure 1: A picture showing something

Figure 2: A graph showing another thing

Figure 3.1: A tabel showing yet another thing, that appears in chapter 3.

# Introduction

**In the introduction, you write the background of your topic and discuss the observation that spurred you on to do this research project. Explain the purpose of the paper and present your research question(s) and the hypothesis at the end of this section. This section is typically a couple of pages long.**

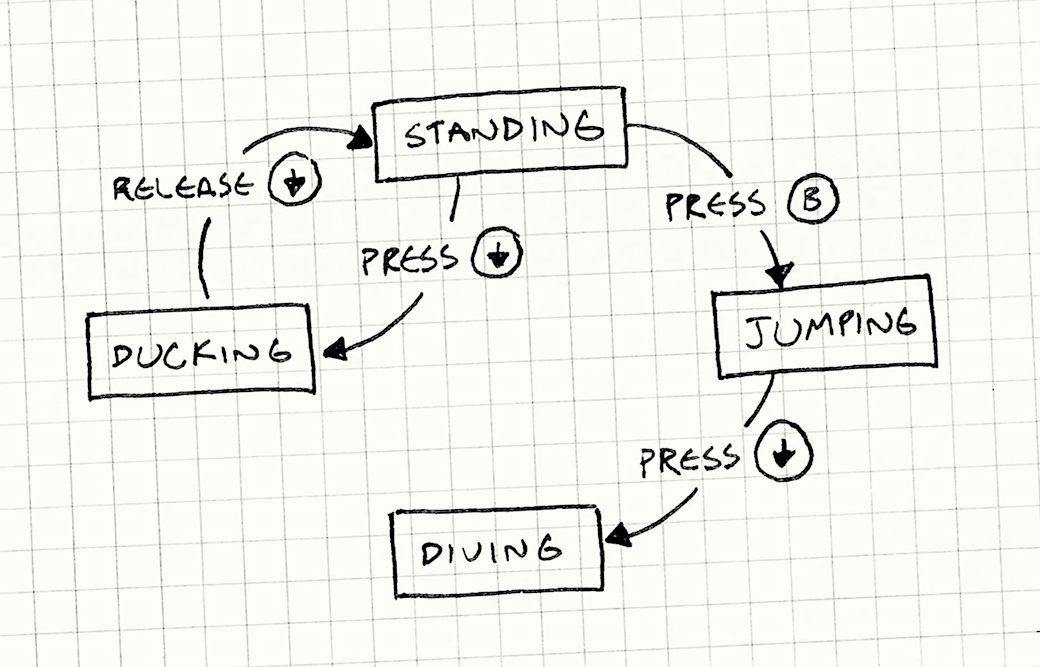
# Literature Study / Theoretical Framework

## Finite State Machines

### What is a Finite State machine?

A Finite State Machine is a model of computation based on a hypothetical machine made of one or more states. Only a single state can be active at the same time, so the machine must transition from one state to another in order to perform different actions. [1]

### How does a finite state machine work?



You have a fixed amount of states that the machine can be in. In the example above that is standing, ducking, jumping and diving. The Finite State Machine can only be in one state at a time. This means you can’t be standing and ducking at the same time. When down is pressed for example we transition from the standing state into the ducking state. When down is released we go back to the standing state. Each state has its transitions for example when we are in the ducking state and the input of B is pressed comes in we will not jump. In the ducking state we are only checking if down gets released to go back to the standing state. When we are in the standing state and B is pressed we transition into the jumping state. If we press down in the jumping state we then transition into the diving state. So per state we check an input and if that input matches what we are looking for we transition into the next state. These inputs can be from an controller or events from the world around the character. [2]

### Where are finite state machines used?

Finite State Machines are used when you need to react to a signal, event or input from the external world.

Examples:

|  |  |  |
| --- | --- | --- |
| * Vending machines | * Alarm Clock | * Video games |
| * Traffic lights | * Microwave |  |
| * Elevators | * Cash Registers |  |

[3]

### STate machines are usefull when:

* You need to respond to a series of inputs, events and signals over time. [2]
* You can split the work into multiple states. [2]
* The behaviour changes based on an internal state. [2]

## Goal Oriented Action Planning (GOAP)

### What is Goal Oriented Action Planning (GOAP)?

Goal Oriented Action Planning is an AI system that allows the Non playable Characters create their own plan with the given actions to satisfy their goal.[4]

### How does Goal Oriented Action Planning work?

Goal Oriented Action Planning has Actions. Action have preconditions and effects so first we need to check if we have met the precondition. Then we do the action and get the effect. Example our goal is to be hydrated these are the actions we have: Afbeelding met tekst, Lettertype, cirkel, diagram

Automatisch gegenereerde beschrijving

Based on the preconditions this is the plan that gets created.

Afbeelding met tekst, cirkel, lijn, diagram

Automatisch gegenereerde beschrijving

So we want to drink water first we search for a water bottle since it has no precondition. The effect we get is we have now a water bottle. So we go to the next action Open the water bottle. The precondition of that action is that we need a water bottle we got that from our last action so now we can start our new action. The effect we get is that the water bottle is now open. The next action is to drink water. We have the precondition so we drink water the effect is our goal to be hydrated. Good our plan is complete.

### Where is Goal Oriented Action Planning used?

In the game F.E.A.R the Non-Playable Characters use Goal Oriented Action Planning in combination with a small Finite State Machine for their AI.

### GOAl oriented action planning is usefull when:

* you want to decouple the actions from each other so you can focus on each action individually.[4]
* you want code that is easy to test and maintain.[4]

## The comparison finite state machines vs Goal Oriented Action Planning

### Differences

Finite State Machine (coupled):

Afbeelding met schermopname, cirkel, lijn, Lettertype

Automatisch gegenereerde beschrijving

[4]

Goal Oriented Action Planning (decoupled):

Afbeelding met schermopname, cirkel, Lettertype, tekst

Automatisch gegenereerde beschrijving

[4]

Finite State Machines tell the Non Playable Characters exactly how to behave and what to do next while the planner needs to create a plan with the given actions to satisfy a goal. So in short this would be the Finite State Machine knows what State comes after the next state but for Goal Oriented Action Planning the plan is created to satisfy what his goal is at that moment and then creates a new plan if that goal changes. [5]

# Research

## Research Question:

What is the impact of switching from a Finite State Machine to a Goal Oriented Action Planning (GOAP) in a simulation game?

## Hypothesises:

**H0**: After comparing the data graphs from both Finite State Machines and Goal Oriented Action Planning, there is no real difference in performance. Both take the same amount of time per frame or there is less than a 5% difference in average script execution time.

**H1**: The data graph of the Goal Oriented Action Planning (GOAP) shows a longer average script execution time per frame of more than 5% compared to the average execution time per frame of State Machines.

**H2**: After a 100 000 patients have passed through the hospital the Goal Oriented Action planning nurses take more coffee breaks.

**H3**: After a 100 000 patients have passed through the hospital the Goal Oriented Action Planning Patients leave the hospital more angry.

**H4**: After a 100 000 patients have passed through the hospital the Finite State machine has more Patients treated.

## Methodology:

### Experiment Set-Up:

There is one level where I just switch the Goal Oriented Action planning agents with the Finite State machine agents and build. It’s important that there are no meshes and other clutter so we can just see the performance of the scripts. The agents with the Goal Oriented Action Planning are created following a guide from unity learn with the author Penny The Byll[6]. For the agents of the Finite State Machine I will be recreating the agent behaviours of the Goal Oriented Action Planning making sure they behave in the same way.

### Agent Behaviors

**Patient:**

Afbeelding met tekst, diagram, schermopname, Rechthoek

Automatisch gegenereerde beschrijving

The Patient can do 6 things:

1. The patient spawns and goes to the hospital.
2. Once at the hospital the patient registers himself.
3. After registering the patient goes to the waiting room.
4. If the patient has to wait too long he gets angry and leaves the hospital.
5. The patients goes with an available nurse to a cubicle and gets treated.
6. The patient leaves the hospital and goes home

**Nurse:**

Afbeelding met tekst, lijn, diagram, schermopname

Automatisch gegenereerde beschrijving

The nurse can do 3 things:

1. Get a patient from the waiting room and see if a cubicle is available.
2. Treat the patient.
3. Coffee break.

### Hardware:

Information over the hardware that the build is tested on:

* Processor: AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz
* Ram: 16GB
* Type: system: x64-processor

### Data Collection:

To test the performance of each build I will use the Unity Profiler. I will be running the Unity Profiler with the target set to 60FPS on each build multiple times and take the average of five runs.

The metrics I will look at to formulate an answer:

* Average execution time spend on the scripts per frame.
* How many times he went over the given time of 60fps.

### Checking agent behavior:

Important data for the level set up:

* The are only 9 cubicles that can be used to treat patients.
* There are 9 nurses in the level.
* There is a patient spawner that spawns a 1000 agents every second until the total number of a 100 000 agents have been spawned.
* Once an agent can switches from one action to another it takes 0.01 seconds if there isn’t another condition that has to be met.
* After 0.25 seconds in the waiting room the patient can leave angry for both Finite State Machines and Goal Oriented Action Planning.
* After 0.25 seconds the nurse can take a coffee break for both Finite State Machines and Goal Oriented Action Planning.

This is a list of data that I will look at to see if the agents behave differently:

* The amount of patients that get treated.
* The amount of patients that left because they had to wait too long.
* The amount of coffee breaks taken by the nurses

All the data is kept in states where I just add to the state each time something happens.

After a 100 000 patients have gone home or left angry I will write all the data to a txt file and quit the application. I do this five times for Finite State machines and five time for Goal Oriented Action Planning then I take the averages and compare them.

## Experiment:

### Testing the level for performance:

I will compare the Finite State Machine with the Goal Oriented Action Planning build. Both builds spawn a 100 000 patients and after the last patient has been treated or left angry the application stops. I will use the Unity Profiler on each build to get the performance data from it.

### Data Collection:

I will run each level for a duration of 5 minutes while the Unity Profiler is collecting data. Once I have the data from a level I will only look at the script execution time, while the target frame rate(FPS) is set to 60 FPS. After completing the test I will extract the Unity Profiler data and import it in the Unity Profiler Analyzer for a more detailed analysis. I will run the level with the Unity Profiler for five times in total and compare their data against themselves and take the average of the five runs.

### Prediction:

My prediction is that the Finite State machine will better for the performance of the overall simulation. So **H0**  “After comparing the data graphs from both Finite State Machines and Goal Oriented Action Planning, there is no real difference in performance. Both take the same amount of time per frame or there is less than a 5% difference in average script execution time. “ should be true.

### Answer:

After I have compared the builds, I can **answer Hypothesis 0 and 1**.

### Testing Agent Performance:

I will create a build for both Finite State Machines and Goal Oriented Action Planning. I run each application five times. The applications close after a 100 000 agents have gone through the hospital and write a txt file with data. Then I take the averages of both their five runs and conclude which one is better at what.

### Data Collection:

This is how I collected my data for patients treated, patients that left angry and the amount of coffee breaks a nurse took.

Patients treated: Once a patient exits the get treated it counts as a treated patient.

Coffee breaks: each time a nurse takes a coffee break it counts as a coffee break.

Patients that left angry: once the counter in the Go To Waiting Room action hits 0.25 seconds for a patients he counts as a patient that left angry.

Each time I count something I keep track off it with a world state and after a 100 000 patients have been treated I write the states and their values to a txt file.

### Prediction:

I predict that the state machine will treat more patients and take more coffee breaks. This means that **H2**: “After a 100 000 patients have passed through the hospital the Goal Oriented Action planning nurses take more coffee breaks.” Is false.

**H3**: “After a 100 000 patients have passed through the hospital the Goal Oriented Action Planning Patients leave the hospital more angry.” Is true.

**H4**: “After a 100 000 patients have passed through the hospital the Finite State machine has more Patients treated.” Is true.

### Result:

Results for the simulations with the Finite State Machine as the agents behaviour:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | average |
| Patients Treated | 77 122 | 73 128 | 72 973 | 71 893 | 77 215 | 74 466.2 |
| Patients Angry | 22 887 | 26 872 | 27 027 | 28 107 | 22 785 | 25 535.6 |
| Coffee breaks | 909 | 900 | 900 | 900 | 900 | 901.8 |

Results for the simulations with Goal Oriented Action Planning as the agents behaviour:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | average |
| Patients Treated |  |  |  |  |  |  |
| Patients Angry |  |  |  |  |  |  |
| Coffee breaks |  |  |  |  |  |  |

### Answer:

# Discussion

**In this section, you offer an interpretation of the results you obtained and try to relate them to the theoretical framework you presented. This is typically not a very long section, but obviously one of the most important ones.**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ac quam nec arcu semper dignissim. Nulla quam magna, varius sit amet pharetra et, dictum quis elit. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Proin ullamcorper, ipsum sit amet scelerisque rhoncus, leo quam rhoncus elit, sit amet ullamcorper tellus nisi eget sapien. Suspendisse potenti. Ut non justo viverra, tempus felis vitae, elementum mi. Morbi at dui sed lacus fringilla condimentum. Duis non odio ac arcu volutpat vehicula eu et turpis.

Praesent vitae magna ante. Nulla in orci lacus. Donec quis vestibulum mi. Sed ipsum sapien, pretium maximus purus sed, bibendum consequat lectus. Aliquam porttitor dolor eu gravida vulputate. Vestibulum ut urna eget massa tincidunt ultricies. Morbi hendrerit sapien at diam tincidunt semper. Aliquam ut quam dictum quam maximus tempor sed at felis.

# Conclusion

**In this section, you ascertain the demonstrable outcomes of your study and outline the merits of the project for the academic field and the discourse community. This is typically not a very long section, but obviously also one of the more important ones.**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ac quam nec arcu semper dignissim. Nulla quam magna, varius sit amet pharetra et, dictum quis elit. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Proin ullamcorper, ipsum sit amet scelerisque rhoncus, leo quam rhoncus elit, sit amet ullamcorper tellus nisi eget sapien. Suspendisse potenti. Ut non justo viverra, tempus felis vitae, elementum mi. Morbi at dui sed lacus fringilla condimentum. Duis non odio ac arcu volutpat vehicula eu et turpis.

Praesent vitae magna ante. Nulla in orci lacus. Donec quis vestibulum mi. Sed ipsum sapien, pretium maximus purus sed, bibendum consequat lectus. Aliquam porttitor dolor eu gravida vulputate. Vestibulum ut urna eget massa tincidunt ultricies. Morbi hendrerit sapien at diam tincidunt semper. Aliquam ut quam dictum quam maximus tempor sed at felis.

# Future work

**This section is sometimes standalone, sometimes incorporated in the conclusion. It looks at the shortcomings of the study, alternative strategies, and what could be the next course of action in the research field. This is typically not a very long section.**

While this experiment was good to check the differences between Goal Oriented Action Planning and Finite State machines it still was limited in the amount of choice it gave to the agents. So when there is a limited amount of choice you can make I would say that gives a little bit of an unfair advantage to the Finite State Machines. So if I where to run these experiments again I would make them a lot bigger so there is an abundance of choice.

It would also be nice to look into something else than simulation since its then up to the player itself directly to see a difference or if they like one over the other.

Since the most used AI systems for non-playable characters are Behaviour Trees and Finite State Machines it would be nice to see a comparison between all of them.

Goal Oriented Action Planning is also sometimes used in a small finite state machine, making that and then comparing it again can give a different result but I would do it without the simulation then.

# Critical Reflection

**This section is typically associated with a bachelor paper, not other forms of serious writing. It allows the student to reflect on the learning outcomes, both academically and in terms of personal growth.**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ac quam nec arcu semper dignissim. Nulla quam magna, varius sit amet pharetra et, dictum quis elit. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Proin ullamcorper, ipsum sit amet scelerisque rhoncus, leo quam rhoncus elit, sit amet ullamcorper tellus nisi eget sapien. Suspendisse potenti. Ut non justo viverra, tempus felis vitae, elementum mi. Morbi at dui sed lacus fringilla condimentum. Duis non odio ac arcu volutpat vehicula eu et turpis.

Praesent vitae magna ante. Nulla in orci lacus. Donec quis vestibulum mi. Sed ipsum sapien, pretium maximus purus sed, bibendum consequat lectus. Aliquam porttitor dolor eu gravida vulputate. Vestibulum ut urna eget massa tincidunt ultricies. Morbi hendrerit sapien at diam tincidunt semper. Aliquam ut quam dictum quam maximus tempor sed at felis.

# References

**In this section, you list all the references you made in alphabetical order; consequently adhere to the referencing style you have chosen.**

Casey Raes (2014), Processing (second edition).

Saccade. (n.d.). In Wikpedia. Retrieved November 6 2016 from <https://en.wikipedia.org/wiki/Saccade>

Sarah Northway (2016) A year in VR Northway [Powerpoint slides] from <https://www.gdcvault.com/play/1024631/A-Year-in-VR-A>

[1] <https://code.tutsplus.com/finite-state-machines-theory-and-implementation--gamedev-11867tn>

[2] <https://gameprogrammingpatterns.com/state.html>

[3] <https://web.stanford.edu/class/cs123/lectures/CS123_lec07_Finite_State_Machine.pdf>

[4] https://gamedevelopment.tutsplus.com/goal-oriented-action-planning-for-a-smarter-ai--cms-20793t

[5] <https://www.gamedevs.org/uploads/three-states-plan-ai-of-fear.pdf>

[6] <https://learn.unity.com/project/goal-driven-behaviour>

# Acknowledgements

**In this section, you can thank people who contributed to your work in a meaningful way.**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ac quam nec arcu semper dignissim. Nulla quam magna, varius sit amet pharetra et, dictum quis elit. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin ullamcorper, ipsum sit amet scelerisque rhoncus, leo quam rhoncus elit, sit amet ullamcorper tellus nisi eget sapien. Suspendisse potenti. Ut non justo viverra, tempus felis vitae, elementum mi. Morbi at dui sed lacus fringilla condimentum. Duis non odio ac arcu volutpat vehicula eu et turpis. Praesent vitae magna ante. Nulla in orci lacus. Donec quis vestibulum mi. Sed ipsum sapien, pretium maximus purus sed, bibendum consequat lectus. Aliquam porttitor dolor eu gravida vulputate. Vestibulum ut urna eget massa tincidunt ultricies. Morbi hendrerit sapien at diam tincidunt semper. Aliquam ut quam dictum quam maximus tempor sed at felis.

# Appendices

**In many cases, there are items that were developed for a research paper that can’t go into the actual paper in full. Things suc as code, art pieces, output of statistical analysis, questionnaires, … In this section, you can present these elements; use the first page to list and number the items, then paste them sequentially. If some items are too large, you can store them online, and link to them. Common practice is to keep those links active at least one year after the publication of the thesis.**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ac quam nec arcu semper dignissim. Nulla quam magna, varius sit amet pharetra et, dictum quis elit. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin ullamcorper, ipsum sit amet scelerisque rhoncus, leo quam rhoncus elit, sit amet ullamcorper tellus nisi eget sapien. Suspendisse potenti. Ut non justo viverra, tempus felis vitae, elementum mi. Morbi at dui sed lacus fringilla condimentum. Duis non odio ac arcu volutpat vehicula eu et turpis. Praesent vitae magna ante. Nulla in orci lacus. Donec quis vestibulum mi. Sed ipsum sapien, pretium maximus purus sed, bibendum consequat lectus. Aliquam porttitor dolor eu gravida vulputate. Vestibulum ut urna eget massa tincidunt ultricies. Morbi hendrerit sapien at diam tincidunt semper. Aliquam ut quam dictum quam maximus tempor sed at felis.