Southampton Solent University

FACULTY OF MEDIA ARTS AND TECHNOLOGY

Computer Games (Indie)

2017/2018

Tayler-James Martin

“The design and Development of a 2D settler management game focusing on self-managing settlers with needs”

Supervisor : Mark Bennet

Date of presentation : February 20

Contents

[1. Aims and Objectives 4](#_Toc507156355)

[Aims 4](#_Toc507156356)

[Objectives 4](#_Toc507156357)

[2. Research findings 5](#_Toc507156358)

[*Villager behaviour implementation* 5](#_Toc507156359)

[*Goal Orientated Action planning1,2,3* 5](#_Toc507156360)

[*State machines4,5* 5](#_Toc507156361)

[*Agents6,7* 5](#_Toc507156362)

[**Conclusion** 6](#_Toc507156363)

[*Engine choice* 6](#_Toc507156364)

[*Unity Engine 2017* 6](#_Toc507156365)

[*Unreal Engine 48* 6](#_Toc507156366)

[Defold9 7](#_Toc507156367)

[**Conclusion** 7](#_Toc507156368)

[3. Reading List/References 8](#_Toc507156369)

[4. Project specification 9](#_Toc507156370)

[Game world 9](#_Toc507156371)

[Resources 9](#_Toc507156372)

[*Buildings* 10](#_Toc507156373)

[Villager Actions 13](#_Toc507156374)

[Villager Goals 15](#_Toc507156375)

[5. Task list 17](#_Toc507156376)

[6. Gantt Chart 18](#_Toc507156377)

[5. Risk analysis 19](#_Toc507156378)

[6. Flow charts 20](#_Toc507156379)

[Agent decision making 20](#_Toc507156380)

[“Move to” villager action 20](#_Toc507156381)

[“Pick up resource” villager action 21](#_Toc507156382)

[“Drop off resource” villager action 22](#_Toc507156383)

[“Eat food” villager action 22](#_Toc507156384)

[“Drink water” villager action 22](#_Toc507156385)

[“Use building” villager action 23](#_Toc507156386)

[“find job” villager action 23](#_Toc507156387)

[7. Pseudocode 24](#_Toc507156388)

# Aims and Objectives

## Aims

For this project, I intend to create an AI system, that allows agents to look after themselves based off needs.

The game surrounding this, will be a very basic settlement builder, where the player can line up jobs that they want doing, either building Houses, storerooms, and farms or collecting wood, stone, and water. The focus of the project will be the settlers, who will choose a job and then carry it out, while also taking care of their need to eat, drink and sleep.

What I want to achieve from this project is a way to demonstrate my ability to create a system that controls agents based off some input from a player to create seemingly realistic behaviour, as well as demonstrate this in a game-like environment, that I intend to further flesh out in the future.

The Aims of this project is to make an AI system that can account for the basic needs of its agents, while also carrying out a given task from an outside source (The player). I would also like to add some very basic settlement building mechanics to add gameplay and intractability

## Objectives

My first objective for this project was to research what the most appropriate algorithms for this project will be. I researched Goal orientated action planning, State machines, Behavioural agents and Decision trees.

Based off this research, and some guidance from my supervisor, I decided to go ahead with using Behavioural agents, specifically goal based agents, as they deal with taking a world state, knowing how their actions will affect a world state, and choosing an action to change the world state to satisfy a goal. This is the type of behaviour I want for my settlers, as I want them to keep track of their needs as if it was a world state, and as a need diminishes I want them to act to maintain it.

Additionally, I looked for game engines that would be suitable to develop this project in, as well as an appropriate format to develop in. Most games of this genre (such as the ones mentioned above) are top down 3D environments, however I feel that to keep the scope of this game tighter, and how I’m not focusing on environments, I intend to be top down 2D instead, reducing unnecessary workload.

From looking at what game engines that would seem appropriate for this type of game, Unity; an engine I am very familiar with, appeared very often on list of recommended engines. While it was never the best engine in these lists, I feel that my experience with it combined with its frequency as a recommended engine, would make it the most suitable engine of choice

Now that most of my research is complete the first thing I intend to implement is the needs system, as most of the project simply stems off it. Starting with making the “Villager” agents need water food and sleep to not “Die” and tweak how long they can survive while doing nothing. I will then implement the actions that the agents can perform to attend to those needs, going to a house to sleep, and a storeroom to eat and drink.

Once those two systems are in place, I will then be able to create the agent system that the villagers can use to keep themselves healthy. Finally, I can start working on the village aspect of the game, giving the player a way to interact with the agents, providing gameplay.

# 2. Research findings

## Villager behaviour implementation

### Goal Orientated Action planning1,2,3

**The idea behind goal orientated action planning is to provide an AI with a list of goals and several ways to achieve them to allow them to ‘decide’ how to achieve them. For example, an AI may need to get through a locked door, if it knows that a key will unlock the door, and that key is in an accessible area, it works backwards from ‘Open the door’ to ‘unlock the door’ and ‘get the key’. This could work well in my game as a villager will need to eat, but to do this it would need food, so with this system it can work backward from ‘Needs to eat’ to ‘Need to go to the storehouse to pick up food’.**

**Goal orientated action planning works best when actions will have prerequisites, as its goal is to create a plan of action, as the name suggests. However, my villagers will have very little prerequisites outside of being in the right place to eat, drink, sleep or work, so this algorithm may end up being overkill for my design**

### State machines4,5

**State machines, or finite state machines are an arrangement of states, each designed for performing one function (moving, picking something up, attacking, etc.), each with triggers for moving onto different states.**

**This would work well for my villagers as all the tasks I wish for them to perform fit well into different states, with needs falling under a certain threshold acting as triggers for moving onto different actions. This would work reasonably well, although state machines can get very complicated very quickly, and if I wish to build upon this project in the future, it could become overwhelmingly complex.**

### Agents6,7

**An Agent is an encapsulated system that takes information from ‘Sensors’ to understand a world state and knows what actions it can take and how they will affect that world state, going through each one to find the best action to take to alter the world state in the way that best suits its goals.**

**This works well for my villagers as each villager can act as one of these agents, with the world state being its current needs and tasks each one can know where to find food water and a bed, as well as how to perform the village building task. It can then have the goal of having optimal food water and sleep then take the action that achieves that the best. this also copes well with future improvements I might want to make to this project as I would simply need to add what actions an agent can take and what world state I wish to achieve with those actions.**

### Conclusion

**Having researched these three different algorithms, I have decided that AI agents is the best to be going forward with. It fits my project well with its ability to view and manipulate a world state based on actions given to it, as well as being the best at coping with scope increase. I also feel it is the most appropriate given how its design is based around multiple Agents functioning autonomously around each other, which is the behaviour I want from my villagers.**

## Engine choice

For engine choice I would prefer to stick with an engine I’ve had experience with previously. This said, I have only ever used two engines, Unity and Unreal engine, so I also decided to look into the use of Defold, an engine I had heard of in passing before designed for developing in 2D

### Unity Engine 2017

**Unity is by far the engine I have the most experience with, when originally deciding on what I was going to do for this project it was in unity I experimented in, it is also the engine I have done nearly all previous projects in, so was my first choice for this project. However, from my own experience it does have its drawbacks.**

**Unity is predominantly a 3D engine, although it does have 2D capabilities, and I’m very familiar with them, it does mean there will be a lot that Unity provides that I will not be using. Rendering a full 3D environment just to view 2D sprites isn’t entirely efficient, so I would have to make sure that my project is optimised well to make up for this.**

### Unreal Engine 48

Unreal Engines benefits over Unity are good tools for creating artwork and other aesthetic components. It also has a visual programming element, which is very useful for prototyping. Most of Unreal’s benefits seem to be for projects of a much larger scale and with a greater emphasis on cosmetics and artwork. Scripts in Unreal are also written in C++, a much more versatile language over C#, the language I would be writing scripts in if I were to use Unity.

Although C++ has more to offer in theory, I am much more experienced with writing in C#, Unreal also has much of its benefits firmly in the art side of projects, which I am not focusing on for this project, so does not feel like the right tool for the job. It also is a much more powerful engine, designed for larger projects.

### Defold9

Defold is a game engine designed for creating 2D games in, primarily for shipping to mobile markets. Unlike Unity and Unreal, Defold has a lot more focus on 2D over 3D, so will have much less overhead when it comes to rendering. Scripts in Defold are written in Lua, a language I am comfortable working in. Defold also provides a number of tutorials on how to use it.

The major issue with Defold, is that I would have to learn how to use an entirely new piece of software for this project, which I’m not certain I have time for.

### Conclusion

Although Defold would probably suit this project more, I feel my experience with Unity would be too useful to do when it comes to keeping this project on track. Unreal Engine 4 is just simply not suited to a project this small and technically based, there are just too many features of unreal engine I wouldn’t get use out of. Unity will be my engine of choice for this project, I simply feel my experience with Unity is not outweighed by the advantages of using a 2D focused engine

# 3. Reading List/References

1. OWENS B, 2014. Goal Oriented Action Planning for a Smarter AI [February 2018]. Available from:<https://gamedevelopment.tutsplus.com/tutorials/goal-oriented-action-planning-for-a-smarter-ai--cms-20793>
2. ANON, 2017. Learning and Using GOAP (Goal Oriented Action Planning) For Squeaky Wheel's Next Game [February 2018]. Available from:<http://www.squeakywheel.ph/blog/2017/2/6/goap-for-our-new-game>
3. THERAOT, 2017. What are the basics of implementing a Goal Oriented AI? [February 2018]. Available from:<https://gamedev.stackexchange.com/questions/136832/what-are-the-basics-of-implementing-a-goal-oriented-ai>
4. MARCO, No date given. Finite State Machine For Game Developers. [February 2081]. Available from: <http://gamedevelopertips.com/finite-state-machine-game-developers/>
5. BEVILACQUA F, October 2013. Finite-State Machines: Theory and Implementation. [February 2081]. Available from: <https://gamedevelopment.tutsplus.com/tutorials/finite-state-machines-theory-and-implementation--gamedev-11867>
6. GUPTA P, September 2017. Rational Agents for Artificial Intelligence [February 2018] Available from: <https://hackernoon.com/rational-agents-for-artificial-intelligence-caf94af2cec5>
7. ANON, No date given, AI - Agents & Environments [February 2018] Available from: <https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_agents_and_environments.htm>
8. CHOO J, March 2016, What are the main pros and cons of Unreal Engine 4 in general? [February 2018], Available from: <https://www.quora.com/What-are-the-main-pros-and-cons-of-Unreal-Engine-4-in-general>
9. ANON, No date given, Frequently asked questions [February 2018], Available from: <https://www.defold.com/faq/>

# 4. Project specification

This project will be a settlement management game, based around a small village. The player will have to collect wood and stone to build buildings, as well as collect food, and water to keep the villagers alive. Villagers will seek out food, water and shelter when needed, and will perform tasks from a task list populated by the player.

## Game world

The game world be a simple area with the starting village in the centre, giving the player room to expand. It will be limited in size for this project, as there is limited amounts the player could achieve with a larger amount of space. The world will have 4 major zones, Rocky, fields, forest and floodplains. Each with a benefit or a deficit to buildings built there, which will be described

## Resources

Resources will be a primary mechanic of the game, being what is used to create buildings as well as what is used to keep villagers alive.

|  |  |
| --- | --- |
| Water | |
| Purpose | Drunk by villagers to survive |
| Produced by | Well building |
| Description | One of the primary resources consumed by villagers to stay alive |

|  |  |
| --- | --- |
| Food | |
| Purpose | Eaten by villagers to survive |
| Produced by | Farm building |
| Description | One of the primary resources consumed by villagers to stay alive |

|  |  |
| --- | --- |
| Wood | |
| Purpose | **Used in construction of some buildings** |
| Produced by | Forrester building |
| Description | One of the resources used to produce buildings |

|  |  |
| --- | --- |
| Stone | |
| Purpose | **Used in construction of some buildings** |
| Produced by | Mine building |
| Description | One of the resources used to produce buildings |

|  |  |
| --- | --- |
| Energy | |
| Purpose | **Used by villagers to perform tasks** |
| Produced by | Villagers sleeping |
| Description | As villagers perform tasks, they will expend energy, they restore this energy by sleeping |

|  |  |
| --- | --- |
| Work hours | |
| Purpose | **What workers produce by performing tasks in building or using buildings** |
| Produced by | Villagers |
| Description | One of the resources used to produce buildings.  Work hours will be an “invisible” resource, not actually being shown to the player, instead driving the mechanics of the game in the back end. |

## Buildings

There will be 6 buildings, a mine for producing stone, a Forrester for producing wood, a farm for producing food and a well for producing water, a storehouse for storing wood, stone, food and water, and a house for villagers to sleep in.

|  |  |
| --- | --- |
| Mine | |
| Purpose | Produces stone |
| Requires | Wood to build, villager to operate |
| Spawns at start of the game? | No |
| Best zone | Rocky |
| Worst Zone | Floodplains |
| Description | One of 4 production buildings, designed to produce stone at the expense of villager time.  The mine will be one of the larger buildings in the game |

|  |  |
| --- | --- |
| Forrester | |
| Purpose | Produces Wood |
| Requires | Stone to build, villager to operate |
| Spawns at start of the game? | No |
| Best zone | Forrest |
| Worst Zone | Fields |
| Description | One of 4 production buildings, designed to produce wood at the expense of villager time. |

|  |  |
| --- | --- |
| Farm | |
| Purpose | Produces Food |
| Requires | Wood and stone to build, villager to operate |
| Spawns at start of the game? | Yes |
| Best zone | Fields |
| Worst Zone | Rocky |
| Description | One of 4 production buildings, designed to produce food at the expense of villager time. |

|  |  |
| --- | --- |
| Well | |
| Purpose | Produces Water |
| Requires | Stone to build, villager to operate |
| Spawns at start of the game? | Yes |
| Best zone | Floodplains |
| Worst Zone | Forest |
| Description | One of 4 production buildings, designed to produce water at the expense of villager time. |

|  |  |
| --- | --- |
| Storehouse | |
| Purpose | Stores water, food, stone and wood |
| Requires | Stone and wood to build |
| Spawns at start of the game? | Yes |
| Best zone | N/A |
| Worst Zone | N/A |
| Description | A storehouse is where villagers will store and pick up all resources. As villagers will need to go to storehouses to collect resources for building as well as eating and drinking, these will make up the core of villages |

|  |  |
| --- | --- |
| House | |
| Purpose | Give villagers a place to sleep |
| Requires | Stone and wood to build |
| Spawns at start of the game? | Yes |
| Best zone | N/A |
| Worst Zone | N/A |
| Description | The house will provide a villager a place to sleep, as well as increasing the maximum population size of the village |

|  |  |
| --- | --- |
| Construction site | |
| Purpose | Built into other buildings |
| Requires | Varies |
| Spawns at start of the game? | No |
| Best zone | N/A |
| Worst Zone | N/A |
| Description | When a building is placed down by player, a construction site appears, a villager must then bring the needed resources to it and build it. |

## Villager Actions

The Game will revolve around what actions a villager can do and what that action will achieve, these will mostly be eating, drinking as well as building and using buildings. But also, the other actions in between these that make them possible.

|  |  |
| --- | --- |
| Move to | |
| Prerequisites | Energy |
| Description | Allows to villager to reach a destination to perform other actions |

|  |  |
| --- | --- |
| Pick up resource | |
| Prerequisites | Energy, resource in range to pick up |
| Description | A resource in range is removed and then added to the villager’s inventory |

|  |  |
| --- | --- |
| Drop off resource | |
| Prerequisites | Energy, resource in villager’s inventory |
| Description | A resource is removed from villager inventory and added to building’s inventory in range |

|  |  |
| --- | --- |
| Eat food | |
| Prerequisites | Energy, Food in inventory |
| Description | Food is removed from villager inventory and villager hunger is decreased |

|  |  |
| --- | --- |
| Drink water | |
| Prerequisites | Energy, water in inventory |
| Description | Water is removed from player inventory and villager thirst is decreased |

|  |  |
| --- | --- |
| Sleep | |
| Prerequisites | House |
| Description | Villager stays in house for a period of time while energy is increased |

|  |  |
| --- | --- |
| Build | |
| Prerequisites | Energy, construction site with resources |
| Description | Removes resources from construction site and adds work hours onto construction, if work hours meet a threshold, where it then turns into the desired building. |

|  |  |
| --- | --- |
| Use building | |
| Prerequisites | Energy, Farm building |
| Description | Adds work hours to a building, that is then turned into resources at a ratio, based on the building. |

|  |  |
| --- | --- |
| Find job | |
| Prerequisites | Energy, Job list not empty |
| Description | Selects job from job list |

## Villager Goals

Villagers behaviour is based off goal achievement. They will always try to achieve the goal with the highest priority. As a goal goes unachieved its priority will increase, along with other factors based on that goal until it is the one with the highest priority.

|  |  |
| --- | --- |
| Eat food | |
| Priority increased by | Hunger increasing, having food in inventory |
| Goal achieved by | Hunger decreasing |
| Prerequisites to act on | Having food in inventory |

|  |  |
| --- | --- |
| Drink water | |
| Priority increased by | Thirst increasing, having water in inventory |
| Goal achieved by | Thirst decreasing |
| Prerequisites to act on | Having water in inventory |

|  |  |
| --- | --- |
| Get food | |
| Priority increased by | Hunger increasing, not having food in inventory |
| Goal achieved by | Having food in inventory |
| Prerequisites to act on | Being at storehouse |

|  |  |
| --- | --- |
| Get water | |
| Priority increased by | Thirst increasing, not having water in inventory |
| Goal achieved by | Having water in inventory |
| Prerequisites to act on | Being at storehouse |

|  |  |
| --- | --- |
| Complete Job | |
| Priority increased by | Having a job queued |
| Goal achieved by | Job being completed |
| Prerequisites to act on | Having a job queued |

|  |  |
| --- | --- |
| Get job | |
| Priority increased by | Not having a job queued |
| Goal achieved by | Gaining a job |
| Prerequisites to act on | Job in queue to be picked up |

|  |  |
| --- | --- |
| Get Stone | |
| Priority increased by | Having a job that requires stone |
| Goal achieved by | Having stone in inventory |
| Prerequisites to act on | Having a job that requires stone, having an empty inventory |

|  |  |
| --- | --- |
| Get Wood | |
| Priority increased by | Having a job that requires wood |
| Goal achieved by | Having wood in inventory |
| Prerequisites to act on | Having a job that requires wood, having an empty inventory |

|  |  |
| --- | --- |
| Empty inventory | |
| Priority increased by | Having something in your inventory, no job queued |
| Goal achieved by | Having an empty inventory |
| Prerequisites to act on | Being at storehouse |

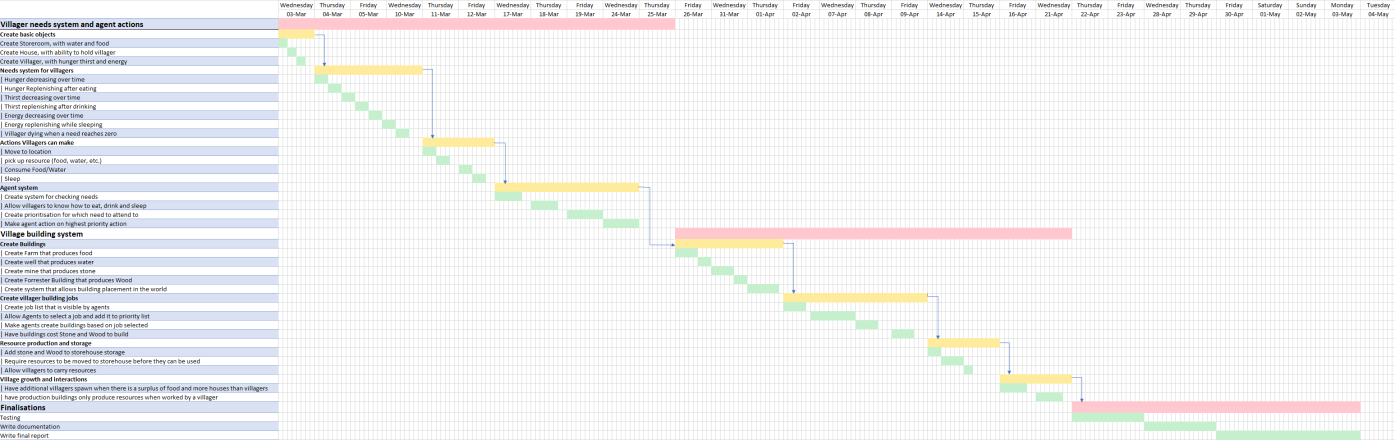
|  |  |
| --- | --- |
| Pickup resources from building | |
| Priority increased by | Having an empty inventory |
| Goal achieved by | Dropping off resource at storehouse |
| Prerequisites to act on | Being near resource building |

|  |  |
| --- | --- |
| Be at Building | |
| Priority increased by | Actions failing |
| Goal achieved by | Arriving at specified building |
| Prerequisites to act on | Energy |

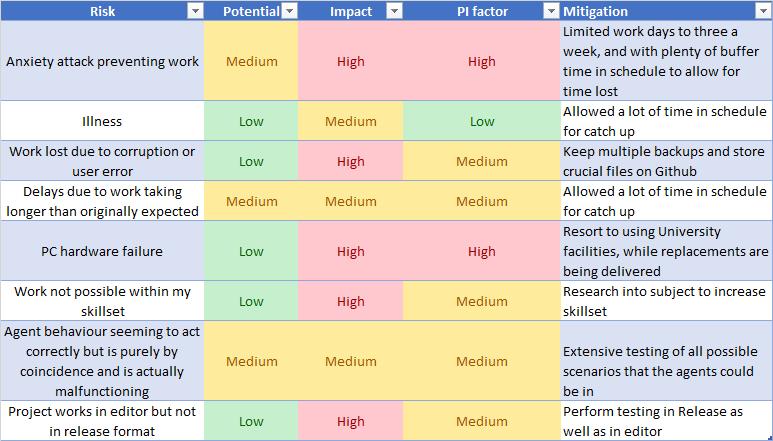
# 5. Task list



# 6. Gantt Chart

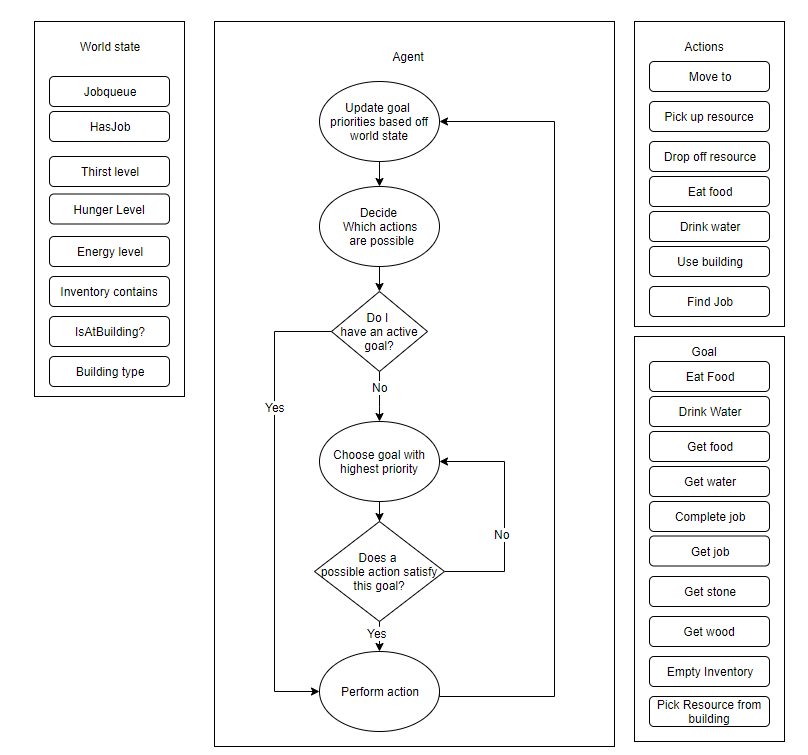


# 5. Risk analysis

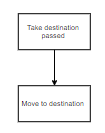


# 6. Flow charts

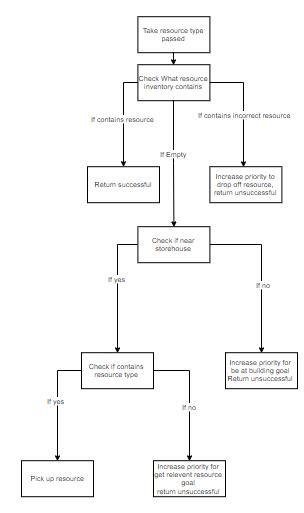
## Agent decision making



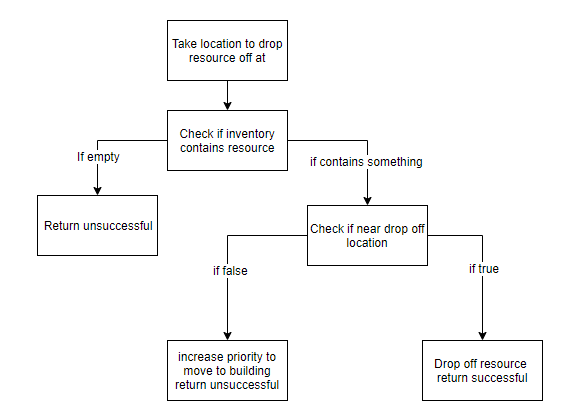
## “Move to” villager action



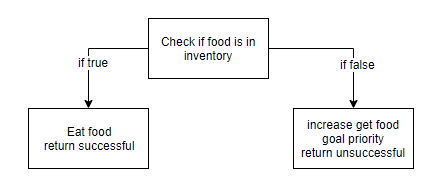
## “Pick up resource” villager action



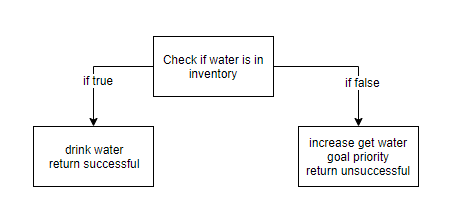
## “Drop off resource” villager action



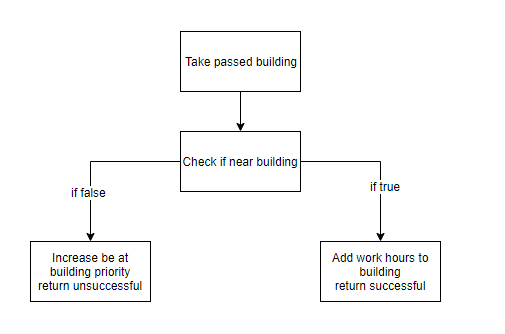
## “Eat food” villager action



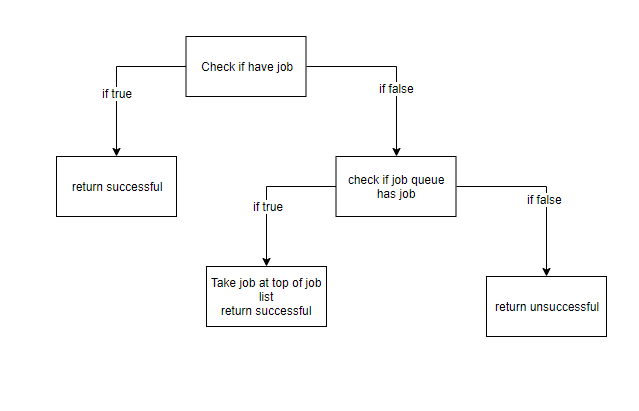
## “Drink water” villager action



## “Use building” villager action



## “find job” villager action



# 7. Pseudocode

