Inter-Agent Communication for Enhancing the World-State Knowledge of A.I. Characters

Matthew Holman - Department of Computer Science

ABSTRACT

***Abstract*—Inter-agent communication is potentially a useful technique to be used in creating AI characters for a video game. This paper explores the effect of implementing a method for two agents to communicate with one another on the time it takes those agents to complete a simple resource gathering task. This communication method is compared against identical AI agents who do not communicate, and agents that use a 'brute force' method of completing the same task. Analyzing the results of 24 experiments reveals that implementing inter-agent communication does indeed positively impact the time it takes for the agents to complete their tasks in a general sense.**

***Keywords*—AI, communication, finite state machine, FSM, video game**

INTRODUCTION

In his book, *AI for Games*, author Ian Millington, writes: "ask a gamer about game AI, and they think about decision making: the ability of a character to decide what to do" [1, p. 297]. To this end, the author provides a general organizational framework to contextualize the various AI algorithms he teaches in his book. This framework consists of three primary layers. These layers include a *movement* layer, a *decision making* layer, and a *strategy* layer [1, p. 10-12]. *AI for Games* displays these layers in a diagram, reproduced here in Figure 1 [1, p. 10].

Millington's *movement* layer "refers to algorithms that turn decisions into some kind of motion" [1, p. 11]. The situations described in *AI for Games* that would require movement-based AI seem to focus mainly on navigating an AI character to the location where a non-movement based action is to be performed [1, p. 11]. In order for an AI agent to behave reasonably in the context of a video game where the agent is required to move, such an agent would require adequate intelligence in order to do so. A robust path-finding algorithm, such as the A* algorithm, should be sufficient for the scenario under consideration in this paper.

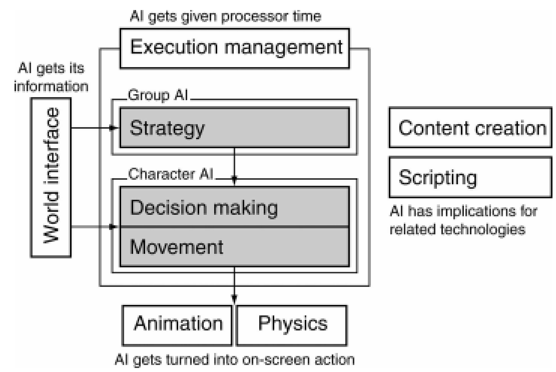

Figure 1: Millington's Three Layer AI Model

The *decision making* layer in Millington's AI model "involves a character working out what to do next" [1, p. 11]. The author describes AI characters as having "a range of different behaviors that they could choose to perform" [1, p. 11]. This layer of Millington's AI model is the one that decides which of these behaviors a character should perform in a given circumstance [1, p. 11]. A structure such as a finite state machine, which involves a distinct number of actions that a character can do as well as the logic that switches between these states, is perfectly applicable to the scenario tested herein.

The *strategy* layer described in *AI for Games* controls "an overall approach used by a group of characters" [1, p. 12]. The author goes on to state that while each character that is controlled by the *strategy* layer will have their own *movement* and *decision making* layers, "their decision making will be influenced by a group strategy" [1, p. 12].

Certainly, the intelligent application of existing AI algorithms which utilize Millington's three layered AI model could produce convincing game characters that satisfactorily complete a given task. However, while this approach may provide an efficient solution to a task, it may or may not provide the most realistic or natural-looking solution.

While groups of creatures in the natural world can both move to desired locations and make intelligentseeming decisions on their own, rarely, if ever, is there an overriding intelligence that dictates the strategy of an entire group. For example, the pheromone trails left by groups of ants can allow them to work as a coordinated team in the search for food. However, intuitively, these pheromones could only influence the ants that have access to that trail in the first place. The ants that did not have access to the pheromone trail could not be affected by it.

The previous example, while simplistic, is seemingly one of the closest ways that the group logic described in *AI for Games* can be found in nature. However, even this example falls short of the level of behavioral control that the author describes. Millington states that "in this category [the *strategy* layer] are AI algorithms that don't control just one character, but influence the behavior of a whole set of characters. Each character in the group may (and usually will) have their own decision making and movement algorithms, but overall their decision making will be influenced by a group strategy." [1, p. 12].

Implicit in the author's statement above, is that the strategic layer of AI would affect *all* the characters in the group *simultaneously*. Clearly, this is something that the pheromones in the ant colony example could not accomplish. Additionally, while the use of 'strategic layer' AI in controlling a group may be able to efficiently carry out a task, the way that task *appears* to be carried out to an attentive viewer may not look as if it were being performed by a creature found in nature.

It should be noted, however, that although something analogous to a strategic layer AI might not be immediately present in creatures from the natural world, it *can* be present in the *unnatural* one. For example, a military commander issuing orders to their troops on a battlefield through a radio receiver could be seen as a strategic layer AI of sorts. A contemporary example of such an AI in a game context is is AlphaStar - an AI which issues orders to groups of units in the game *Starcraft II* [2].

In order to create AI characters which are both efficient in their task, and which appear to be entities that arise from nature, this paper proposes doing away with a strategic layer AI algorithm. Instead, this layer could be replaced with an enhanced decision layer scheme. This new decision layer would emulate the effects of a strategic layer through the use of knowledge sharing between characters. Characters could gather information about the state of the world, and then propagate that information between themselves directly - not entirely unlike the pheromones in the ant colony example. This communication could potentially cause these AI characters to behave in a way that is both efficient to the assigned task, and which is believable to the viewer. To this end, this paper investigates the following questions:

1) What impact would the use of agent communication have on the time it takes AI characters to complete a simple given task, compared to identically programmed AI characters who do not communicate?
2) How would the time it takes for AI characters who communicate in the process of completing a given task compare to the time it takes for characters who operate in an unnatural 'brute force' manner to complete the same task?

SCENARIO

There are seemingly infinite situations in a video game context in which agent communication could be useful. These situations include sharing the contents of two agent's hands of cards when they are on the same team in a multi-player card game, deciding where soldiers should move to surround a player in an action game, or informing workers where a potential resource is in a strategy game, among others. This paper focuses on just one such situation. Obviously, performing tests on a single task precludes making broad conclusions about the usefulness of agent communication in general. However, a tight focus on a single idea will allow for more concrete statements about this *particular* communication-relevant scenario.

The experiments performed for this paper focused on agent communication in a resource gathering scenario. Each run of the simulation takes place on a two-dimensional square map, which is constructed as an $n \times n$ series of cells. Two 'space aliens' are tasked with collecting 20 different crystals which have been distributed on the map. Once an alien collects a crystal, it will then return it to a single drop-off point.

The 20 crystals were apportioned on the board in one of three ways The way the crystals were apportioned depended on the test that was being run. In the first apportionment method, as seen in 2, the crystals are distributed randomly across the board.

This random placement of crystals leads to a more or less even distribution over the map without any obvious patterns in their placement.

The second method of distribution, presented in Figure 3, places most of the crystals together in a large cluster which is placed randomly on the map.

The clustered placement of crystals is meant to represent a resource on the map that the aliens can return to multiple times. The purpose of this is to prevent the knowledge of the cluster's location from becoming obsolete once it is visited by an alien. For example, the knowledge of the location of a single crystal in isolation becomes meaningless once that crystal has been collected. However, even if a crystal is collected from the cluster, there are still multiple other crystals close to the

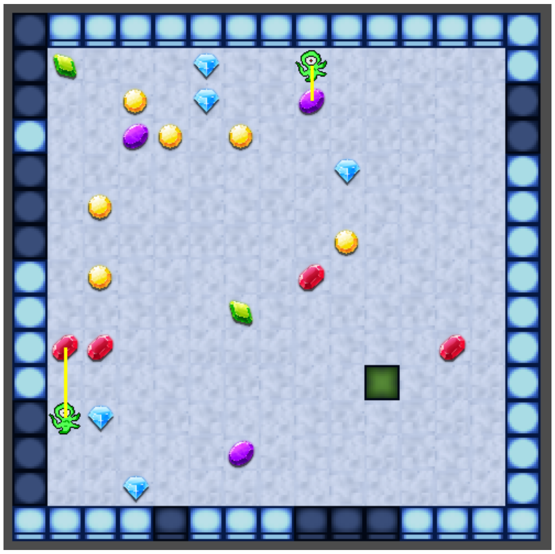

Figure 2: Random crystal Apportionment

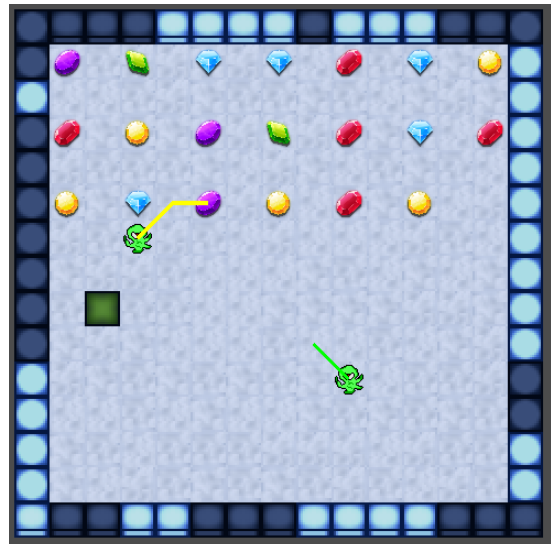

Figure 4: Spaced crystal Apportionment

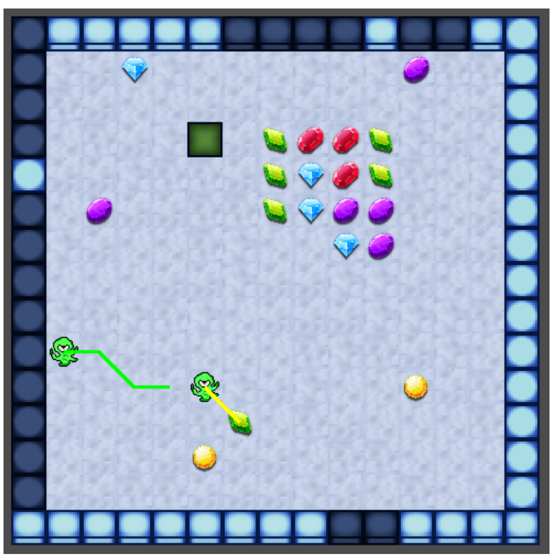

Figure 3: Clustered crystal Apportionment

one that was picked up. In this way, the knowledge of the initial crystals' location is still useful, even after it has been collected.

The final distribution method attempts to evenly space the 20 crystals at the top of the map, as seen in Figure 4.

The purpose of apportioning the crystals in this way is so that they appear consistently in one part of the map. Furthermore, the way the crystals are arranged in a regular pattern prevents too much clustering of resources which helps separate the tests using this configuration from the tests using the random or clustered configurations.

In the cases of the clustered distribution and the evenly spaced distribution, some of the crystals may be randomly placed elsewhere on the map. This random placement occurs if the crystal to be added to the map were otherwise going to spawn on the same location as the drop-off point, either of the aliens, or another crystal. The reason a crystal might spawn atop another crystal is if the cluster or configuration it would otherwise be a member of becomes too large as dictated by the algorithm that places them. This random placement is beneficial as it gives the aliens a reason to explore more of the map once the evenly spaced or clustered crystals are all discovered and returned to the drop-off point.

Depending on the setup of the test, the drop-off point for the crystals can be either placed in a random location on the map, or in a fixed location in the center of the play area. If the drop-off is placed in the center, the two aliens are also placed in the same position. If the drop-off is placed randomly, then the two aliens are placed in random starting locations on the board as well.

Obviously, the starting arrangement of the simulation will impact which crystals the aliens initially detect, and the order in which they go about collecting them. The starting arrangement will also impact when the aliens can initially communicate with one another. Starting positions that place the aliens far away from one another may prevent them from communicating with one another early in the simulation. Similarly, starting positions that are close to one another may allow them to share their information early on. However, for any helpful communication to occur in the first place, the aliens must be coded in a way that would allow that to happen.

Agent Construction

Theo Wadsley and Malcolm Ryan provide a succinct description of Michael Bratman's [3] "Belief-Desire-Intention (BDI) agent architecture" [4, p. 105]. Wadsley and Ryan describe each aspect of the BDI model thusly: "*Belief* models the agents subjective knowledge of the world. *Desire* models an agent's abstract goals, while *intention* models how those goals are translated into action" [4, p. 106]. In a loose sense, the aliens used in these experiments make use of the BDI model.

The *desire* aspect of the aliens BDI implementation is inherent to the simulation itself: the aliens desire to collect as many crystals as possible in as short a time as possible. Currently, the aliens can do nothing other than the crystal gathering task that is being tested. A more complex scenario might require more than just this one behavior, and thus would require a mechanism to determine different desires. However, in this case, no such mechanism is required.

Because the *desire* of the aliens is only to collect the crystals, the *intention* aspect of their BDI implementation need not be overly complicated. In an effort to more clearly highlight the effect of communication on the aliens crystal collection efforts, the aliens pursue their goal via the logic contained in a simple finite state machine.

The reason for foregoing more complicated AI schemes, such as decision trees, neural networks, etc., is because it was believed that whatever interesting behaviors the aliens performed should be clearly the result of their communication, or lack thereof. A more complicated underlying AI implementation would cloud those interesting behaviours by introducing additional variables. For example, if the aliens used a neural network instead of a finite state machine, then the way the neural network was trained might be the underlying reason why they collect crystals quickly, as opposed to their use of communication between one another. However, regardless of the underlying logic used to direct the aliens, they still require a way of knowing about their current world-state.

World State Data Structures

The aliens utilize several FIFO (first in, first out) queues and lists to represent their knowledge (*belief*) about the state of the world. Each list or queue contains a set of X/Y coordinates that represent different locations on the map.

The 'dropoff_locations' list holds the locations of all the drop-off points on the map. The experiments presented in this paper only use a single drop-off point. However, future experiments could be made to incorporate multiple drop-off points.

A 'crystal_locations' queue contains the map coordinates of all the crystals that the alien has 'seen.' This queue will grow when additional crystals are spotted. Each each new location the alien 'sees' is appended to the end of the queue. The vision range of each alien, shown in Figure 5, is represented by a circle. This range was chosen arbitrarily, and has a radius approximately $3\times$ the width of the alien itself. The reason for choosing a vision range of this size is because it is large enough to allow the aliens to see a decent number of crystals as they explore them map. However, it is not so large that the aliens could easily see all the crystals on the map without exploring.

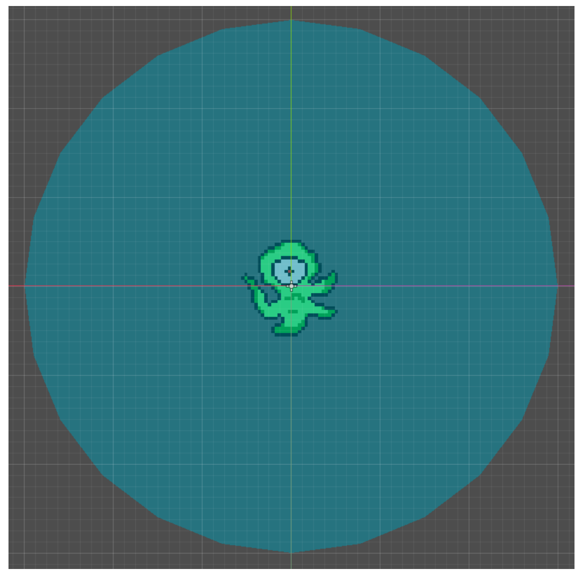

Figure 5: alien Vision Range

Any crystal that enters the vision range is counted as seen. If such a crystal is not already inside the crystal_locations queue, it will be added. The crystal_locations queue will shrink when the crystal at the front is designated as the aliens' movement destination.

The 'crystal_location_blacklist' contains the locations on the map from where crystals have been collected. This queue exists to prevent an alien from returning to a location where a crystal no longer resides.

The 'movement_location_blacklist' is similar to the crystal_location_blacklist. However, instead of holding empty crystal locations, it holds locations that an alien has visited when exploring the map, but which did not contain a crystal.

Finally, the 'movement_location_yellowlist' contains a list of map cells that surround one of the previously visited cells that appear in the movement_location_blacklist. This list is special in that an alien is not prevented from setting one of these locations as its destination. Instead, the alien only has a 25% chance of accepting one of these cells as its final destination. If one of these yellowlisted cells is not accepted, then a new destination is chosen instead.

The reasoning behind employing a yellowlist is for the aliens to more greatly explore the map. This exploration is accomplished by potentially preventing, but not completely prohibiting, an alien that has visited a certain cell to set an immediately adjacent cell as its next destination. For the experiments presented in this paper experiments, a yellowlist range of two has been added to each blacklisted cell. This means that all the cells that can be reached with any combination of two horizontal or vertical movements from the blacklisted cell are added to the yellowlist.

Underlying Finite State Machine

A representation of the finite state machine (FSM) that controls the actions of the aliens can be seen in Figure 6. This FSM contains ten weakly connected states. One state is considered the 'start state,' and represents an alien before it has been initialized at the start of the simulation. The FSM primarily contains three groups of states. Each group represents the three primary actions that an alien can perform. These actions include exploring the map (E), picking up a crystal (P), and dropping off a crystal to the drop-off point (D).

The FSM also contains one unconnected 'pseudo-state,' (C), which represents the aliens' use of inter-agent communication. This pseudo-state is only activated in the tests that utilize inter-agent communication, and will not take the alien out of any other state it was previously in. For example, if an alien is currently moving to pick up a crystal, and en route to the crystal it communicates with the other alien, neither alien will immediately change states as a result of the communication. Rather, only the aliens' world-state information will be updated for later use. However, once an aliens' world-state information is adjusted, its *future* actions could differ from what its actions *would have been* if it had *not* communicated.

The states available to the aliens are designated as:

- **S**: start state
- **E**: exploring_choose_destination
- **E'**: exploring_moving_to_destination
- **E"**: exploring_arrived_at_destination
- **P**: picking_up_crystal_choose_destination
- **P'**: picking_up_crystal_moving_to_destination
- **P"**: picking_up_crystal_arrived_at_destination
- **D**: dropping_off_crystal_choose_destination
- **D'**: dropping_off_crystal_moving_to_destination
- **D"**: dropping_off_crystal_arrived_at_destination
- **C**: communication 'pseudo-state'

Each state has a number of transitions to other states, which are explained as follows:

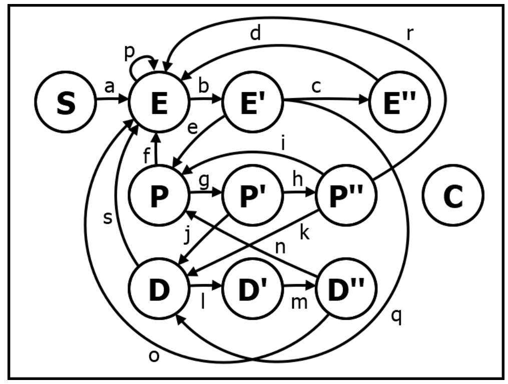

Figure 6: Underlying FSM

- **a**: The alien is initialized, and the simulation begins.
- **b**: A new randomly selected destination has been chosen, or a brute force destination has been decided upon.
- **c**: The alien has reached its randomly selected destination, and has no knowledge of crystal locations.
- **d**: The alien has no knowledge of crystal locations, and must choose a new destination.
- **e**: The alien has gained knowledge of at least one crystal location, but is not carrying a crystal.
- **f**: The alien could not select a valid crystal location from its crystal_locations queue to move to.
- **g**: The alien has acquired a valid crystal location to move to.
- **h**: The alien has reached its destination.
- **i**: The alien is not carrying a crystal despite having arrived at its intended destination, but it knows the location of at least one other crystal. This will happen if a second alien picks up a crystal intended for the first.
- **j**: The alien is now carrying a crystal, but did not reach the location of the one it intended to. This will happen if the alien collects a different crystal than the one it intended while en route to its previously selected destination.
- **k**: The alien has reached its destination, and is now carrying a crystal.
- **l**: The alien has acquired a valid drop-off location to move to.
- **m**: The alien has reached its destination.
- **n**: The alien has dropped off its crystal, and has knowledge of other crystal locations.
- **o**: The alien has dropped off its crystal, but does not have knowledge of other crystal locations.
- **p**: When the alien is using brute force movement, and cannot find a valid destination, disable brute force movement and choose a new random desti-

nation ala normal exploration.

- **q**: The alien is now carrying a crystal and needs to select a drop-off point destination (applicable to non-detecting brute force movement).
- **r**: The alien should begin exploring normally if any set of circumstances occur other than those presented in state. transitions **i** or **k**. This should be considered an 'emergency' situation, as it should not normally occur.
- **s**: The alien could not select a valid drop-off location from its dropoff_locations list to move to, and should begin normal map exploration. This should be considered an 'emergency' situation, as it should not normally occur.

Communication Methods

When researching agent communication, it became clear that there were two primary models that might be relevant to the scenario being tested. Nickles, Rovatsos, and Weiß cite Cohen and Levesque ([5] [6]) when they describe "the *mentalistic* approach" to agent communication [7, p. 94]. Nickles *et al.*then cite Singh ([8]) when describing the "*objectivist*" or "*social semantics*" strategy to communication [7, p. 94].

According to Nickles *et al.*, "the *mentalistic* approach specifies the meaning of utterances [communication] by means of a description of the mental states of the respective agents" [7, p. 94]. These mental states include "their beliefs and intentions, and thus indirectly their behavior" [7, p. 94]. A communication method such as this would seem to treat each alien in the simulation as a complete individual that could make its own decisions. Each alien could then act according to its current knowledge state, and any appearance of cooperation between aliens could be thought of as being voluntary in nature.

Nickles *et al.*describe the objectivist approach as one which attempts "to determine communication from an external point of view, focussing on public rules" [7, p. 94]. According to the authors, the benefit to the objectivist approach is in terms of verifiability of an agent's cognitive state [7, p. 94]. For example, the authors write that when using the mentalistic approach, one would have to make "simplifying but unrealistic assumptions to ensure mental homogeneity among the agents, for example that the interacting agents were benevolent and sincere, and it [the mentalistic approach] neglects the social context of utterances" [7, p. 94]. Nickles *et al.*assert that "objectivist semantics in contrast is fully verifiable, [as] it achieves a big deal of complexity reduction through limiting itself to a small set of normative rules, and has therefore been a significant step ahead" of the mentalistic approach [7, p. 94].

Chosen Method

The communication method used in the experiments presented in this paper is most closely related to the mentalistic approach described above. The reasons for choosing this method are twofold. Firstly, if implementing a mentalistic-type approach can yield interesting results, and an objectivist approach is in some way superior to a mentalistic one, then implementing an objectivist approach in future experiments might yield even better results than the current mentalistic model. However, in order to compare the two sets of results, tests centering on a mentalistic approach must be conducted regardless.

Ultimately, however, the experiments presented in this paper are meant to demonstrate a initial foray in using agent communication to accomplish a game-related task. As such, implementing both communication styles would be outside the scope of this paper. Secondly, implementing a social rule set as described above would have exceeded the spirit of simplicity that characterized the rest of the design for the aliens.

The communication used in the tests this paper presents is based on the proximity between one alien and the other. The communication aliens occurs when one alien enters the other's communication range. A relatively short range, depicted by the circle in Figure 7, was chosen and was kept constant throughout all the experiments that utilized the agent communication.

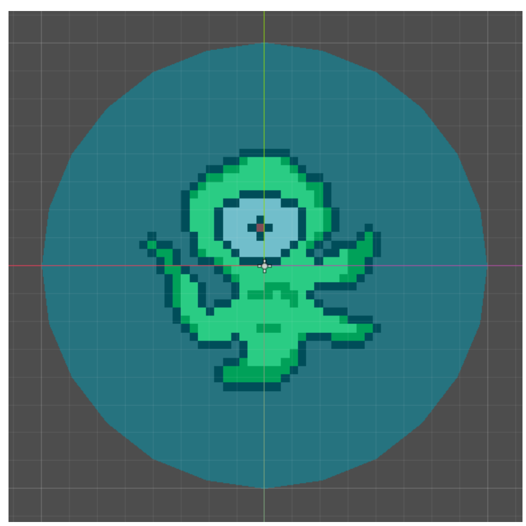

Figure 7: alien Communication Range

This range was chosen as it was allowed the aliens to communicate when they would either naturally collide or pass close by one another. This range was arbitrarily chosen as it appeared to be the range at which two beings, capable of speech, would have a natural conversation with one another without resorting to 'shouting' or anything similar.

When the two aliens communicate with one another, they perform set unions on their respective world-knowledge data structures. In the cases of the crystal_location_blacklist, movement_location_blacklist, and movement_location_yellowlist data structures, each alien has their versions of these lists combined together and then reassigned to both aliens. This means that after they communicate, both aliens will have the same crystal_location_blacklist, movement_location_blacklist, and movement_location_yellowlist data structures as per the equations below. Please note that several of the datastructure names have been modified for formatting purposes:

$$\begin{aligned} ACL_1 &= \{Alien\ 1\ crystal\_location\_blacklist\} \\ ACL_2 &= \{Alien\ 2\ crystal\_location\_blacklist\} \\ ACL_{new} &= ACL_1 \cup ACL_2 \\ ACL_1 &= ACL_{new} \\ ACL_2 &= ACL_{new} \end{aligned}$$

$$\begin{aligned} AML_1 &= \{Alien\ 1\ move\_location\_blacklist\} \\ AML_2 &= \{Alien\ 2\ move\_location\_blacklist\} \\ AML_{new} &= AML_1 \cup AML_2 \\ AML_1 &= AML_{new} \\ AML_2 &= AML_{new} \end{aligned}$$

$$\begin{aligned} AMY_1 &= \{Alien\ 1\ move\_location\_yellowlist\} \\ AMY_2 &= \{Alien\ 2\ move\_location\_yellowlist\} \\ AMY_{new} &= AMY_1 \cup AMY_2 \\ AMY_1 &= AMY_{new} \\ AMY_2 &= AMY_{new} \end{aligned}$$

However, the aliens crystal_locations queues are handled in a slightly different manner. First, the crystal_locations are unioned together, as above. Then, the new list of crystal locations is sorted by the distance from the crystal location to the one drop-off point used by the tests presented in this paper. Once the crystals are in their distance-sorted order, the list is 'cut in half.' One half is assigned to one alien, and the other half assigned to the other alien. Because the crystals are sorted by the distance to the drop-off point, assigning the two halves of the unioned list to the aliens in this way has the effect of assigning one alien to pickup a set of crystals which are close to the drop-off point. The other alien is assigned to pick up crystals which are, ostensibly, further away from the drop-off than those that the first alien is to collect. In this way, the aliens indirectly act in a way that is complementary to one another.

Alternative Methods

In a previous iteration of the experiments presented in this paper, the crystal_locations were equally apportioned to each of the aliens. Once the list was unioned, the crystals were *not* sorted as they are in the method actually used. Once the unioned list was created, every other entry in that list was assigned to one of the aliens. This had the effect of giving each alien an equal share of the crystals in the order they were seen.

The above method was not chosen for the final tests presented in this paper, as the pseudo-teamwork that resulted from the method actually employed was seen as more desirable behavior. This small change in the way the crystals were divided up further differentiated the behavior of the aliens between the tests where they communicated, and the tests where they did not.

Experimental Methodology

Technical Information

Technical Information

All the experiments performed for this paper were conducted using a simulation. This simulation is described in the Scenario section of this paper, and was created using the *Godot* game engine, version 3.2.1, a link to which is available in Appendix A - Links. The path-finding the aliens used in the simulation was handled by the A* algorithm. The code which was used to set up the underlying directed graph used by A*, and which executes path searches in this simulation, is a heavily modified version of MIT-licensed code released by GDQuest [9]. For a complete list of system specifications for the computer the simulation was run on, please see the *system.txt* file available at the root directory of the GitHub repository, a link to which is available in Appendix A - Links.

Each run of the simulation was executed by an automated testing script. this script was responsible for starting each trial and collecting its data which was collated and combined as appropriate for later analysis. The source code for both the simulation and the testing script are available via the links in Appendix A - Links.

Experimental Design

The experiments performed for this paper were principally classified by two parameters. The first parameter included which of the three ways the crystals could be apportioned on the map. The second experimental parameter was whether the drop-off and initial alien locations were set to be a fixed position in the middle of the map, or if they were placed in random map locations. This means that there were a total of six experiment types ($3 \times 2 = 6$). These experiment types included:

- **Clustered Crystals, Fixed Start**
- **Clustered Crystals, Random Start**
- **Random Crystals, Fixed Start**
- **Random Crystals, Random Start**
- **Spaced Crystals, Fixed Start**
- **Spaced Crystals, Random Start**

Each experiment type was tested using one of four different crystal collection techniques for the aliens. These techniques included:

- **Communication**: The aliens search the map for crystals, and share their world-state information with one another when they enter communication range.
- **Non-Communication**: The way the aliens behave is *identical* to the communication technique, but the aliens *do not* share their world-state information.
- **Brute Force Detecting**: The aliens move to opposite corners of the map, and mechanically search across each map row in a manner similar to a typewriter. When a crystal enters an alien's vision range, the alien will move to collect it, and will store the locations of other crystals it finds in its crystal_locations queue.
- **Brute Force Non-Detecting**: The aliens move to opposite corners of the map in the same way the 'detecting' version does. However, the aliens have no vision to detect crystals. The only way for an alien to collect a crystal using this technique is for the alien to directly collide with it in the process of traversing the map.

Because of the three crystal apportionment methods, the two starting conditions, and the four crystal collection techniques, there were a total of 24 experiments ($3 \times 2 \times 4 = 24$) performed. Each of the 24 experiments used the size of the map as the independent variable. The map size was increased incrementally from a size of 15×15 to 25×25. This means there was a total of 11 iterations per experiment. Each iteration was performed five times each. As such, there were 1,320 trials ($24 \times 11 \times 5 = 1320$) conducted in total for the experiments presented in this paper.

Each trial was given a strict time limit of 60 seconds, during which the aliens were tasked with collecting as many of the 20 crystals on the map as possible before time ran out. Each 60 second trial involved collecting several data values. These data values included:

- **Communications**: The number of times the aliens communicated during the trial.
- **Crystals Collected**: The total number of crystals the aliens picked up and successfully dropped off during the trial. If an alien collected a crystal, but was unable to return to the drop-off point before the timer expired, that crystal did not count towards the total.
- **Non-Productive Time**: The amount of time that *neither* alien was holding a crystal during the trial.
- **Time**: The total amount of time it took the aliens to collect *all* the crystals during the trial. If the aliens were *unable* to collect all the crystals, this value was equal to the total time of the trial: 60 seconds.

The mean and standard deviation for all four data values was calculated from the five trials done on each iteration of each experiment. Once all 11 iterations of the experiment were complete, there were eight series of data, the means and standard deviations of the data values above, each of which contained 11 data-points corresponding to the 11 map sizes that were tested (15×15, 16×16, ..., 25×25).

While there were only 1,320 trials timed to sixty seconds each ($1320 \times 60 = 79200$ seconds), it took additional time to start and end the simulation program beyond the 60 seconds the simulation was programmed to run. According to the final output from the testing script, it took exactly 287656.61658906937 seconds to run all the tests presented in this paper. This equates to approximately 3.3293 days to gather the data presented here. Previous data collection efforts, available for viewing via the link in Appendix A, took similar amounts of time.

PROBLEMS

Path-Finding

As the A* algorithm was used to handle the aliens' path-finding, each alien would calculate very similar-looking paths from one location to another. While these paths are efficient to guide the aliens to their destinations, a creature found in the natural world would not necessarily move in such a path.

Furthermore, oftentimes the paths the aliens found would contain jagged 'zig-zagging' portions. This is because each path that this implementation of the A* algorithm calculates consists of a list of sequential coordinates for *every* map cell belonging to that path between a given starting point and the path's destination. The implementation of a *path smoothing* technique would have alleviated this issue. Such a technique would have connected non-sequential path coordinates when an unobstructed straight line existed between them. By using such a technique, the number of points in a given path could have been reduced, and the 'zig-zagging' nature of some of the A* path solutions could have been eliminated.

Finally, according to the documentation for the *Godot* engine, the default implementation of the A* algorithm uses the *Euclidean distance* as its heuristic in computing the shortest path from one point to another [10]. This heuristic may or may not have been the best solution for generating the paths the aliens' would take. Although the *Godot* engine allows for other heuristics to be used [10], the experiments presented in this paper used the default.

Obstacles

There were no obstacles or walls used in any of the tests presented in this paper. However, previous iterations of the simulation *did* use walls and obstacles, created by way of a simplex noise generator. However, given the relatively few number of trials used per experimental iteration, it was believed that including walls and obstacles on the map would have introduced too much randomness into the experimental results. While performing more trials per experimental iteration would have likely 'smoothed out' the resulting data, such trials were not practical to perform given the time that data collection would take.

Randomization

In the cases of randomized crystal placement and clustered crystal placement, both of these methods of apportionment can produce crystal placements that differ greatly between trials. Furthermore, placing the drop-off point and the two aliens in different random initial locations can vary from trial to trial as well. Because of this randomization, the outcomes of the trials that use these parameters will not necessarily be consistent from trial to trial. Taking the mean of five trials per experiment iteration is meant to compensate for this inconsistency. However, only performing five trials per iteration may be insufficient for accomplishing this goal.

Trial Run Time

The time for each trial being limited to 60 seconds was an arbitrary decision. It is possible that either a longer or shorter time would produce results that differ in some way from those which are presented in the paper.

A previous iteration of the simulation program used the non-productive time as a means of knowing when to end the trial. When neither alien was carrying a crystal, a ten second countdown timer would begin. If an alien picked up a crystal during that time, the timer would stop and reset itself.

The non-productive timing method was abandoned because when it was employed the total time for each trial could vary wildly from one trial to the next. For example, if the aliens continually found crystals from the beginning of the trial to the end, they could conceivably collect all 20 in 200 seconds ($20\ crystals \times 10\ seconds\ each = 200\ seconds\ total$). Alternatively, if the aliens were unlucky and did not encounter any crystals at all in the first ten seconds of the trial, the simulation would immediately conclude.

Selected Results

The original data collected by the automated testing script is available to view via a link in Appendix A - Links. This data was plotted according to the crystal apportionment and alien/ drop-off starting method the test called for. The data points on the graphs came from the four crystal collection methods the experiments employed. These plots were grouped together by the eight data values that the testing script calculated.

The data values the testing script calculated included the means and standard deviations for four metrics. These metrics include the number of inter-agent communications, the number of crystals the aliens collected, the amount of non-productive time the aliens spent, and the total time it took the aliens to complete each map size iteration of the experiment. This means that there are a total of six graphs for each data value the testing script calculated, for a total of 48 graphs ($3 \times 2 \times 8 = 48$).

It should be noted that the explanatory clarity of some of the original graphs is somewhat lacking. For example, the graphs for the standard deviations of the four data values can vary wildly, as seen in Figure 8.

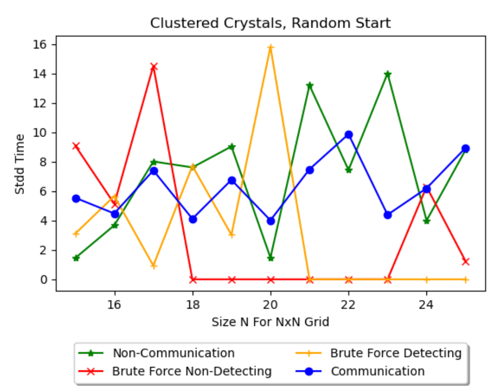

Figure 8: Time to Complete Trial, Std. Dev.

Given the limited number of trials conducted for each iteration of the experiment, and the randomness involved in how the crystals, aliens, and the drop-off point used in the trial are set up, it is reasonable for the standard deviations of the four crystal collection methods used in the trials to appear 'jumbled' when juxtaposed with one another. However, the visualizations of the means of the data values are much more clear.

In general, the means for the number of communications shows a general downward trend as exemplified in

Figure 9. The only experiment who's graph clearly shows an opposite trend is that of the randomly apportioned crystals with a random starting location as seen in Figure 10.

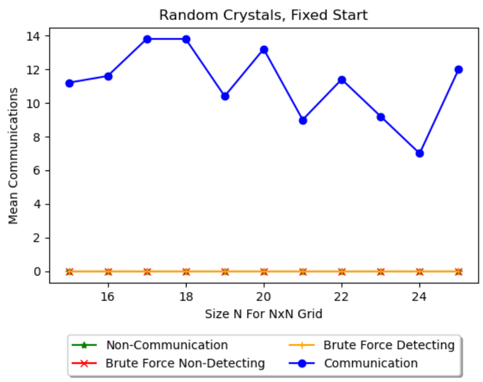

Figure 9: Number of Communications, Mean

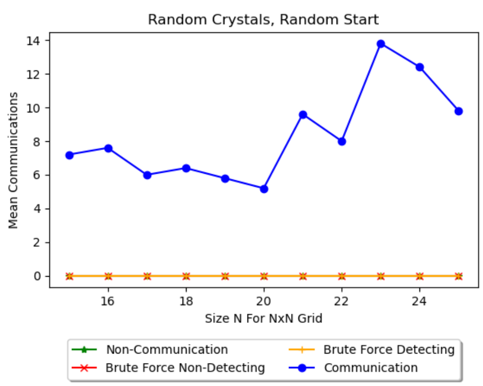

Figure 10: Number of Communications, Mean

Arguably, the most useful data values for measuring the effectiveness of the four crystal collection techniques are: the means of the metrics for the number of crystals collected, the amount of non-productive time, and the total time to complete each map size iteration. For a metric such as the number of crystals collected, a high value indicates better performance as it is desirable for the aliens to collect as many crystals as they can. By contrast, the values for the non-productive time and total time should be as low as possible to indicate desirable performance. This is because a lower non-productive time value indicates that the aliens are carrying crystals more than they would be if that value was higher. Further, a low total time value indicates that the aliens are collecting all the crystals on the map more quickly than if the value was high. Both of these time values being low indicates that the aliens are more efficiently collecting they crystals and returning them to the drop-off point than if their values were higher. Although graphs that show the actual data values collected during the experiments are somewhat clear in what they represent (a link to such graphs can be found in Appendix A - Links), additional clarity can be added to the 'story' the collected data tells.

For each of the three previously mentioned 'useful data values,' the data that was collected was reinterpreted to reflect the *relative rankings* of all four crystal collection techniques across all six experiment types. Because there were four crystal collection techniques, each technique could be rated as being the 'best,' 'second best,' 'third best,' or 'worst' for each iteration of an experiment.

The quantities of each ranking for each collection type were used to create several new series of data. This data included the means, variances, and standard deviations for each of the quantities of relative rankings for all four crystal collection techniques. For examples of this data, please see Appendix B - Example Data.

The means of the rankings for all four collection types were ultimately compared against one another in three different graphs - one for each of the aforementioned 'useful data values.' The added clarity that these new graphs bring to the original data allows for broad statements about the effectiveness of the four crystal collection techniques to be made in the final analysis. These graphs can be seen in Figures 11, 12, and 13.

Please note that when ranking the different techniques against one another it would sometimes be possible to have multiple techniques with identical underlying scores. In these cases, techniques with the same values were ranked equally to one another, and were given the highest possible relative ranking. This means, for example, that it is possible to have multiple 'best' rankings, but no 'third best' or 'worst' rankings of the four collection techniques for a given test. The results of this ranking strategy can be seen in Figure 11. This figure indicates that at least three of the four collection techniques were frequently collecting all the crystals on the map, and thus all received 'best' rankings for various iterations of the experiments.

ANALYSIS

Of all the information presented in Figures 11, 12, and 13, the results shown in Figure 11 require the most explanation. Unlike Figures 12 and 13, Figure 11 does not have as many clearly defined maximums for each of the four relative rankings that correspond to one of the four collection techniques.

Figure 11 initially appears to indicate that all four crystal collection techniques are at least some-

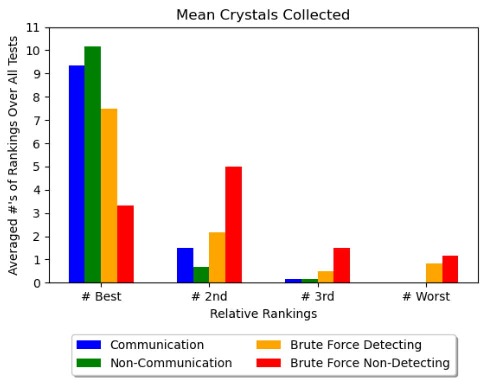

Figure 11: Number of Crystals Collected

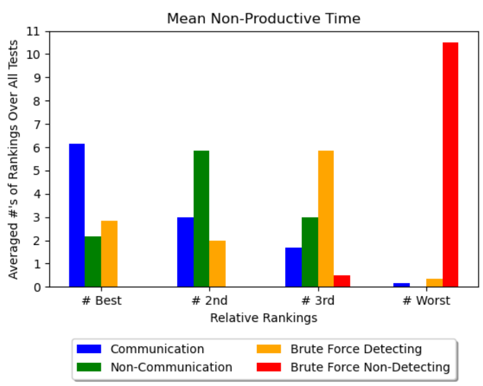

Figure 12: Non-Productive Time

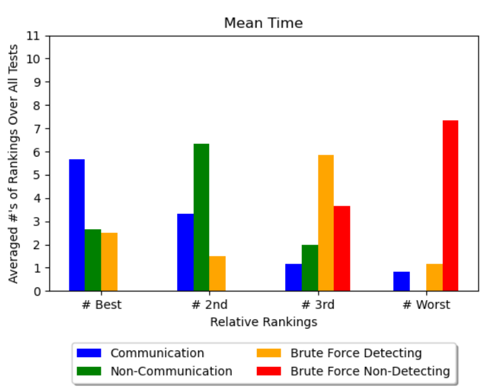

Figure 13: Time to Complete Iteration

what useful for collecting crystals. However, the Non-Communication, Communication, and Brute Force Detecting techniques are clearly all superior to the Brute Force Non-Detecting method as their numbers of 'best' collections over all the experiments far exceed those of Brute Force Non-Detecting. While the numbers of best collections for the top three methods are very similar to one another, the Communication and Non-Communication methods both do better than the Brute Force Detecting method. Although the Non-Communication method technically has the highest number of best crystal collections, the number of best collections the Communication method produced is very close to this value. Further, the Communication method had slightly more *second best* crystal collections than did the Non-Communication method. These considerations suggest that the Communication and Non-Communication methods are approximately equivalent to one another in their ability to collect the crystals from the map.

By contrast, the Brute Force Detecting and Brute Force Non-Detecting methods had far more 'second best,' 'third best,' and 'worst' collection rates than either of the Communication or Non-Communication methods. This clearly establishes both Brute Force techniques as being inferior to the two communication-related techniques in terms of collecting crystals. However, the number of crystals each method collected is not the only consideration in determining which crystal collection technique is the most efficient.

Figures 12 and 13 both have maximum values for the numbers of best, second best, third best, and worst relative rankings that each clearly correspond to one of the four crystal collection techniques. In both figures, the Communication technique has the largest number of best times. The Non-Communication technique has the largest number of second best times. The Brute Force Detecting method had the highest number of third best times. Finally, the Brute Force Non-Detecting technique had the highest numbers of worst times.

Conclusions

To directly answer the questions posed in the introduction to this paper, the impact of using agent communication to complete the given task of collecting crystals from the map, when compared to the time an identical non-communicating agent took to complete the same task, is that the communicating agents were generally able to complete their job more efficiently than the non-communicating ones. Further, the communicating aliens were, in general, able to complete their crystal gathering task more quickly then when the aliens used non-communicating brute force techniques.

Overall, even though the Communication and Non-Communication methods of crystal collection were relatively equal in their ability to collect crystals, the Communication method was clearly superior in the *time* it took to complete the task. As such, of all four crystal collection methods considered in this paper, the method that employed agent communication was the best of the four.

Future Work

There are a number of changes to the simulation that would be valuable to test the effects of. For example, when splitting up the crystal_locations list between the two aliens, it would be interesting to see the effect of reversing the sub-list that contains the crystals that lay further away from the drop-off point. Reversing this list in such a way would cause the alien that receives it to first move towards the crystal furthest away from the drop-off point. The alien would then continue collecting crystals that are progressively closer to it.

Additionally, if more than one drop-off point is used, the crystals could be ordered by the distance to the drop-off point nearest to where the communication took place. Alternatively, the crystal locations could be ordered into several different groups which correspond to the drop-off point closest to where that particular crystal is found.

Regardless of whatever is chosen for further study, it is clear that inter-agent communication is potentially a powerful tool to be used when designing an AI character's behavior.

Appendix A - Links

Data

https://github.com/mholman5721/Agent_Communication/tree/master/Data/Graphs_And_Collated_Data

GitHub Repository Main Page

https://github.com/mholman5721/Agent_Communication

Godot Engine

https://godotengine.org/

Simulation Source Code

https://github.com/mholman5721/Agent_Communication/tree/master/Source_Code

System Information

https://github.com/mholman5721/Agent_Communication/blob/master/system.txt

Testing Script

https://github.com/mholman5721/Agent_Communication/blob/master/Source_Code/runTests.py

Appendix B - Example Data

Number of Crystal Collected - Communication Technique

Test / Metric	# Best	# 2nd	# 3rd	# Worst	# Ties / 33
CC - FS	10	1	0	0	23
CC - RS	11	0	0	0	16
RC - FS	9	2	0	0	19
RC - RS	4	6	1	0	5
SC - FS	11	0	0	0	26
SC - RS	11	0	0	0	16
MEAN:	9.333333333	1.5	0.166666667	0	17.5
VARI:	6.222222222	4.583333333	0.138888889	0	44.25
STDD:	2.494438258	2.140872096	0.372677996	0	6.652067348

Number of Crystal Collected - Non-Communication Technique

Test / Metric	# Best	# 2nd	# 3rd	# Worst	# Ties / 33
CC - FS	10	1	0	0	23
CC - RS	9	2	0	0	16
RC - FS	10	0	1	0	21
RC - RS	10	1	0	0	13
SC - FS	11	0	0	0	26
SC - RS	11	0	0	0	16
MEAN:	10.16666667	0.666666667	0.166666667	0	19.16666667
VARI:	0.472222222	0.555555556	0.138888889	0	20.47222222
STDD:	0.687184271	0.745355992	0.372677996	0	4.524623987

Number of Crystal Collected - Brute Force Detecting Technique

Test / Metric	# Best	# 2nd	# 3rd	# Worst	# Ties / 33
CC - FS	10	1	0	0	21
CC - RS	5	2	2	2	12
RC - FS	8	3	0	0	15
RC - RS	7	3	1	0	10
SC - FS	10	1	0	0	24
SC - RS	5	3	0	3	10
MEAN:	7.5	2.166666667	0.5	0.833333333	15.33333333
VARI:	4.25	0.805555556	0.583333333	1.472222223	29.22222222
STDD:	2.061552813	0.897527468	0.763762616	1.213351648	5.405758247

Number of Crystal Collected - Brute Force Non-Detecting Technique

Test / Metric	# Best	# 2nd	# 3rd	# Worst	# Ties / 33
CC - FS	4	6	1	0	11
CC - RS	2	5	0	4	6
RC - FS	5	4	2	0	12
RC - RS	4	1	4	2	7
SC - FS	5	6	0	0	14
SC - RS	0	8	2	1	0
MEAN:	3.333333333	5	1.5	1.166666667	8.333333333
VARI:	3.222222222	4.666666667	1.916666667	2.138888889	21.55555556
STDD:	1.795054936	2.160246899	1.38443731	1.462494065	4.642796092

References

[1] I. Millington, *AI for games*. Taylor et Francis, a CRC title, part of the Taylor et Francis imprint, a member of the Taylor et Francis Group, the academic division of TetF Informa, plc, 2019.

[2] The AlphaStar Team, "AlphaStar: Mastering the Real-Time Strategy Game StarCraft II." https://deepmind.com/blog/article/alphastar-mastering-real-time-strategy-game-starcraft-ii. Accessed: 2020-05-07.

[3] M. Bratman, *Intention, plans, and practical reason*. CSLI, 1999. https://web.stanford.edu/group/cslipublications/cslipublications/site/1575861925.shtml.

[4] T. Wadsley and M. Ryan, "A belief-desire-intention model for narrative generation," in *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, pp. 105–108, 2013.

[5] P. R. Cohen and H. J. Levesque, "Performatives in a rationally based speech act theory," in *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, pp. 79–88, Association for Computational Linguistics, 1990.

[6] P. R. Cohen and H. J. Levesque, "Communicative actions for artificial agents.," in *ICMAS*, vol. 95, pp. 65–72, Citeseer, 1995.

[7] M. Nickles, M. Rovatsos, and G. Weis, "Empirical-rational semantics of agent communication," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '04, (USA), p. 94–101, IEEE Computer Society, 2004.

[8] M. P. Singh, "A social semantics for agent communication languages," in *Issues in agent communication*, pp. 31–45, Springer, 2000.

[9] N. Lovato and A. Franke, "pathfind_astar.gd." https://github.com/GDQuest/godot-demos/blob/master/2018/03-30-astar-pathfinding/pathfind_astar.gd. Accessed: 2020-05-13.

[10] J. Linietsky, A. Manzur, and the Godot Engine community, "AStar." https://docs.godotengine.org/en/3.2/classes/class_astar.html. Accessed: 2020-05-13.