Collaborative Agents solving Two-Player Puzzle Game

Final Report

Manuel Carvalho
Group 50
Instituto Superior Técnico
Lisbon, Lisbon, Portugal
manuel.g.carvalho@tecnico.ulisboa.pt

Miguel Coelho
Group 50
Instituto Superior Técnico
Lisbon, Lisbon, Portugal
miguel.g.coelho@tecnico.ulisboa.pt

Afonso Gomes
Group 50
Instituto Superior Técnico
Lisbon, Lisbon, Portugal
afonso.gomes@tecnico.ulisboa.pt

ABSTRACT

The problem we will tackle in our project is that of a two-player puzzle game. This game consists of a series of levels with puzzles that require collaboration between two autonomous agents in order to complete them. Higher stage levels will be more complex and thus harder, as they have multiple puzzles dependent on each other and therefore require better coordination between the agents to be cleared.

Our objective is thus to build two Autonomous Agents that can cooperate with each other so they can solve these puzzles, by being able to support each other's actions as needed. This multi-agent system was developed, implemented and tested with the software Unity, while the code was primarily written in the C# language.

INTRODUCTION

How fast can a Multiagent System solve a puzzle? Who solves it faster? Two autonomous agents or two humans?

We're living in times where AI is surpassing humans in almost everything. This project will show how two autonomous agents collaborate and coordinate so they can solve a problem and how much time they take to complete a variety of puzzles.

Our main goal is to develop a Multiagent System composed of two Autonomous Agents that will be able to identify each other's current objectives, act conforming their senses and collaborate with each other when one of them is in need of help in order to solve a puzzle.

The game's puzzles include buttons that open doors, boxes that can be grabbed or dropped, obstacles, such as walls and doors, that require the players to physically interact with in order to surpass them, by jumping over them, for example. Finally, any level in the game has a flag, and reaching it is considered the main objective of the agents when it comes to clearing the game. So in order to complete the puzzle, they will have to find some way to reach it eventually.

The collaboration requirements of the challenges range from non-timing sensitive collaboration, such as dropping off a box on a button, to precisely timed collaboration, like remaining still on top of a button to force a door to stay open for as much time as needed.

RELATED WORK

Our first idea was to implement a single autonomous agent to collaborate with a human player, adapt to him and be proactive. We focused on one of the given examples in the project statement: **FireBoy and WaterGirl**. [2]

Figure 1: **Example level of the FireBoy and WaterGirl game [1].**

This is a puzzle game that can be played by one or two users. The goal of **FireBoy & WaterGirl** is to bring both characters to the end of each level, involving buttons and switches that need to be used by both players in order to get through, while collecting as many gems as possible.

After playing this game, we decided we wanted to create a puzzle game and an autonomous agent that could help a user and collaborate with him to complete it. However, due to the high complexity of the collaboration requirements between an agent and a human, such as perceiving intentions and the (lack of) information exchange, we then changed our project from one autonomous agent collaborating with a player, to two autonomous agents collaborating with each other in order to solve the puzzles.

We also checked the **Geometry Friends** Project [3]. It is a two-player co-op puzzle platformer game with the objective of collecting a set of diamonds in the least amount of time.

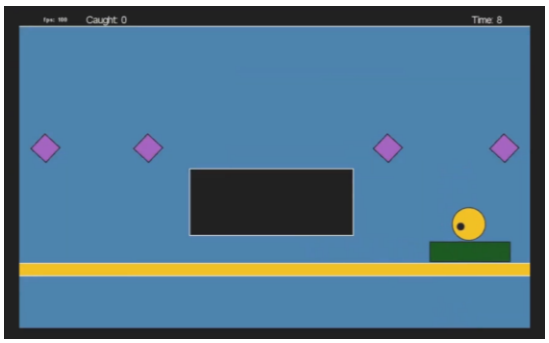

Figure 2: Example of a level of the Geometry Friends game [3].

The game has two characters, a green rectangle and a yellow circle. The rectangle can change its shape to a horizontal or vertical rectangle (with the same area), while the circle character is capable of jumping. The diamonds, which are the goal of these characters, appear in places that require cooperation and coordination of both of them.

We wanted to do something like this, but with a different goal. Our goal was to have both agents reach the flag at the same time, meaning we would still need the cooperation and coordination between the agents to surpass all the obstacles in between them and the flag.

APPROACH

To make it so the developed agents make relevant contributions towards solving the puzzles, we began by carefully observing how us, players, would approach each of these levels and then translate each approach into a stack of temporary objectives that the agents should try to clear in order to make progress.

Each agent should thus have their list of objectives, which starts off as independent from one another, but upon observing another agent's objective, they might update this list with a new priority objective aimed at supporting the partner agent in completing his current objective.

In general, we felt that this approach of having the agents observe the partner agent and then making choices based on those observations was a good idea, since the main focus of the game is collaboration. However, the agents will also be able to take initiative and explore in default mode, that is, if no relevant objective is being observed from the other agent.

We will follow this logic for each type of puzzle at a time at first, until the agents are able to solve levels with at least one instance of each type of puzzle. We ended up naming these levels "basic" levels, while "intermediate" levels focus on different combinations of the puzzle mechanisms present in the "basic" levels. Finally, "advanced" levels were mainly aimed at solving some more specific situations, such as a "deadlock" type of scenario where both agents are trying to get through a door to help the other.

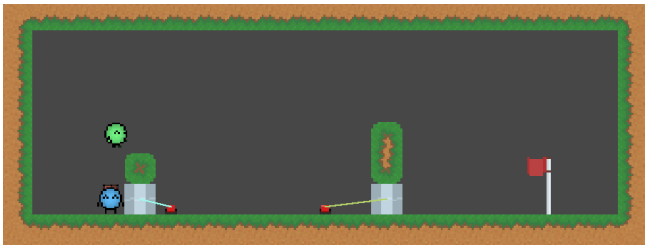

Figure 3: Green agent is jumping over the wall with the help of blue agent that has a box.

An example of how the level shown in **Figure 3** would be solved would be by identifying that one of the agents, that is walking towards the flag, ran into a door, meaning that they would need to press the button in order to open that door. Since there are no buttons on the side of the door the agents are on, the only way to open up that door would be by pressing the button that is on the other side. And in this case, this can only be done by grabbing a box and then helping the other agent jump over the wall.

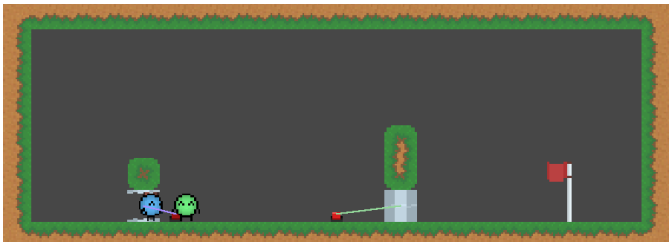

Figure 4: Green agent is pressing the button, allowing the blue agent to get through the door.

Afterwards, when the agent jumped over, he would press the button, like we can see in **Figure 4**, so that the other agent could go through that door and then drop the box at the second button to open the second door and allow both of them to later reach the flag (**Figure 5**).

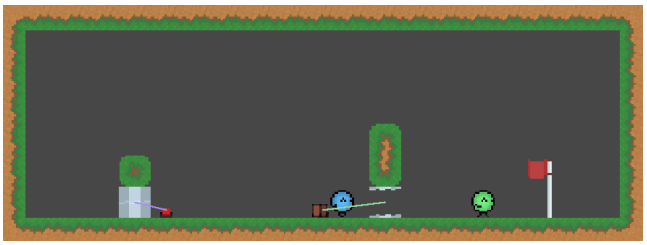

Figure 5: Blue agent drops the box on the button, allowing both agents to get through the last door.

CONCEPTUAL MODEL OF THE MULTIAGENT SYSTEM

The end goal of the players in this game is to reach the flag from whatever position they start off. However, if only one player reaches the flag that is not enough to clear the game. Likewise, if a player makes it to the flag, then leaves, and then a different player reaches the flag this is also not enough to consider a level as completed.

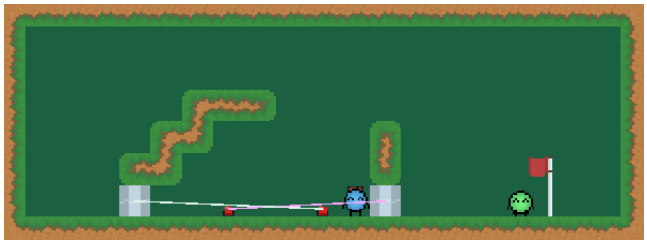

Figure 6: Game has yet to be completed even though one of agents already made it to the flag.

Therefore, the game is only considered cleared when both agents are by the flag at the same time. The levels in the game may include two main types of obstacles that can prevent the players from simply walking directly over to the flag to clear it: ground tiles and doors.

Doors may exist throughout any level and the agents have no way of knowing where these are until they effectively run into one. Any given door should have at least one button associated with it, which, when pressed, opens the door for as long as that button remains pressed. So if whatever is keeping a button pressed walks away from it or is removed from it, the door associated with it will close back, stopping players from getting past it.

To simplify the overall solution a bit, one specification we added regarding this puzzle mechanism is that all information regarding the buttons associated with a given door can be accessed as long as the agent has perceived that door and decided he wants to get past it. Therefore, when an agent finds a door, he can immediately know the location of any buttons that should be pressed to open it, as well as perceive if they are being pressed when attempting to open up the respective door.

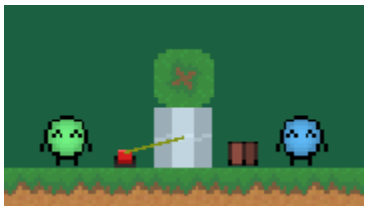

Figure 7: A closer look at every type of obstacle and puzzle mechanism in the game. The line drawn from the button to the door serves as an indicator of which buttons open up which doors.

We could have had the agents searching around the map randomly until they found a button, much like with doors, and then have them check if pressing that button opened up the respective door, but this just makes the solution unnecessarily complex as it does not add much to the main theme of the project, which is the collaboration between the two agents.

The other main obstacle to the flag are ground tiles, and unlike doors, these cannot be rendered ineffective in any way, so as will see later, agents can only bypass either one unit of ground by executing a simple jump action, or two units through a collaborative mechanic: stackable boxes.

Boxes may exist throughout levels and can have a variety of different effects when interacting with any of the previously mentioned puzzle mechanisms, obstacles or the agents themselves.

For starters, they can be picked up and moved around on an agent's back, and later dropped off by them in the position that agent is currently standing on. When an agent is holding a box, this makes it possible for the other agent to jump on top of him, allowing the jumping agent to reach past an obstacle that may have otherwise been unreachable, by executing a second jump action while on top of the other agent. Additionally, when boxes are dropped off on buttons, the buttons will remain pressed, leaving the door associated with that button open for as long as that box does not get picked up and removed from the button by an agent later on.

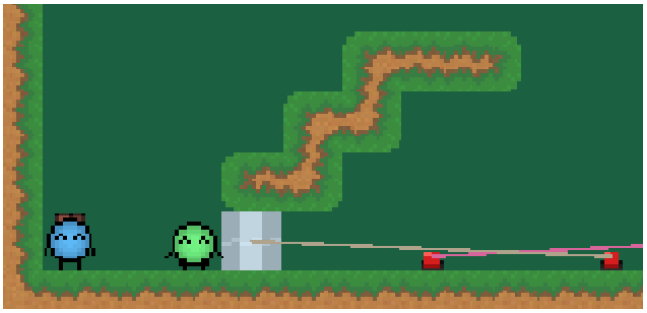

Figure 8: Green agent decided that he wants to jump over the door and ground tiles in order to go press a button on the other side. Blue agent grabs a box that was back at the start of the level upon perceiving this objective.

Much like with the buttons, we have made it so agents can immediately know if any boxes exist in a level, as well as know their locations at any given time, in case they have decided they want to use a box for some end goal. Once again, we could have had agents searching for boxes, but this just strays from their collaborative aspect.

One last thing that is worth mentioning regarding the overall movement of the agents is that we made a slight simplification when it comes to circumventing obstacles that are in between the agents and the flag. Most of our level layouts mainly focus on horizontal layouts rather than vertical layouts, like the

following image where the flag is placed below the agents but slightly to the right.

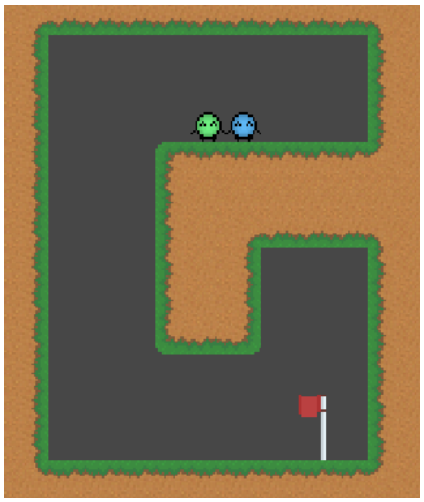

Figure 9: Basic level that would require the agents to walk to the left first and only towards the flag after dropping down, instead of simply rushing to the right from the start.

We made this choice because, once again, this does not have much to do with the collaboration of the agents so we felt that it was fine to focus on levels where agents simply have to walk around the x axis towards their objective, aside from the occasional jumps needed to get past obstacles directly in front of them.

Finally, the following graph gives a general overview of how an agent's objective can evolve as they progress through a level.

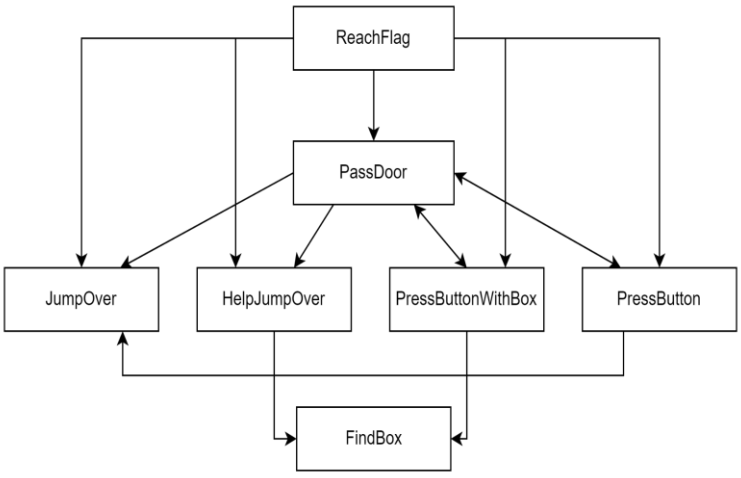

Figure 10: Graph showing how each possible agent objective may lead to other objectives that should be cleared first.

This evolution of state is better explained later on as we go through each level scenario that was explored to build the agents' final behavior.

THE AGENTS AND THE ENVIRONMENTS

The agents have a set of actions that they can choose so they can complete the puzzles. These actions are:

WALK RIGHT: Moves the agent's position one unit to the right in the X-Axis (x+1).

WALK LEFT: Moves the agent's position one unit to the left in the X-Axis (x-1).

JUMP RIGHT: Moves the agent's position one unit up in the Y-Axis and one unit to the right in the X-Axis (x+1, y+1).

JUMP LEFT: Moves the agent's position one unit up in the Y-Axis and one unit to the left in the X-Axis (x-1, y+1).

STAY: Stays in the same position.

GRAB/DROP BOX: Agent grabs a box close to him or if he already has a box, drops it at his current position.

Once a level starts, the agents are given the default objective of the puzzle which is to reach the flag. However, during the run of the level, they are going to encounter several obstacles and will thus decide to take on other intermediate objectives in order to surpass those obstacles. Needless to say, there will be levels where both of them will have to collaborate, coordinate and support each other, so they can find a way to both reach the flag.

The first few designed levels do not have many obstacles and are fairly easier than others to clear even by a simple agent that simply chooses to walk towards the flag. Later levels become harder though, as these have more combinations of obstacles and puzzle mechanisms (doors, buttons and boxes) in the environment for them to sense and interact with.

Basic Level 1

This level is the simplest one we designed and the agent that was developed from it served as a base for all future agents. The starting objective of both agents is to reach the flag, which is easily achievable by simply walking in the correct direction, and jumping instead whenever there is a ground tile acting as an obstacle. This does not require collaboration between them whatsoever, but it helped us establish some basic movement, actions and perceptions.

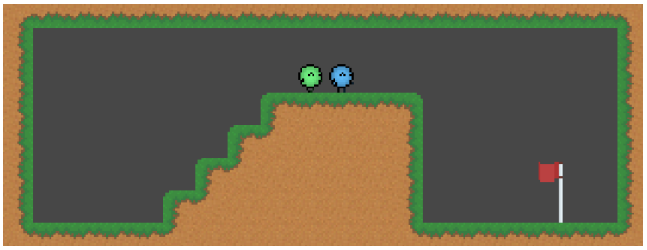

Figure 11: Agents need to jump over the ground tiles to reach the flag.

Basic Level 2

In this level, there is a door and two buttons connected to that door, one on each side of it. The approach to clear this is quite intuitive: one agent has to press the button on the left so that the other agent can first get through, and then this other agent should stay on the button on the right until the first agent has made it past the door.

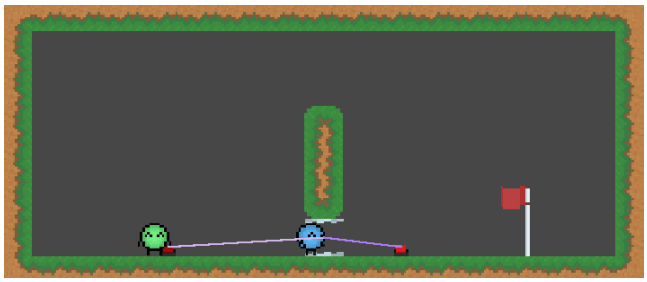

Figure 12: Agents need to cooperate with each other to reach the flag.

The initial agents that cleared the previous level failed to complete this one as they simply kept running into the door which was in the way of the flag. So in order to have them collaborate to open the door one at a time and get through, we implemented two objectives: "PassDoorObjective" and "PressButtonObjective".

An agent will push a PassDoorObjective to his stack of objectives whenever he perceives a door that is in the way of his current objective, and the other agent is not already trying to pass through that same door. This objective is considered cleared as soon as the agent has made it past that door.

A PressButtonObjective will be acquired if an agent perceives that the other agent's current objective is to pass through a door. Upon pushing this objective, he will try to move towards each button associated with that door one at a time until he either:

1. Runs into a wall or another door, at which point he chooses to go look for a different button.
2. Finds the exact button he was looking for, remaining pressed on it until the other agent passes through the door.

So as soon as one of the agents senses the door on this level by running into it, the other one will go look for a button that opens it, and remain on it until the original agent has made it past the door.

As we can see in **Figure 12**, the green agent is pressing the button to open the door for the blue agent to pass. Once he passes, the green agent walks off the button and collides with the door, alerting the blue agent to help him pass through the door, by eventually pressing the button that is on the side as the flag.

Basic Level 3

This one has another new mechanism in the environment, which is the box. These previously mentioned boxes that can be grabbed and dropped, allow the agents to get through the door without the need of one of them to remain on a button. This is especially required for this level, as there is only one button present on the left side of the door.

We decided to implement two new objectives that the agents can acquire, both aimed at clearing this level: "FindBoxObjective" as well as a specification of the PressButtonObjective developed for the previous agent, which we cleverly named "PressButtonWithBoxObjective".

FindBoxObjective has the agent check if any boxes exist on the current level, and if any are found, they will then look for them one by one much like PressButtonObjective has them looking for the buttons. It is considered completed when the agent has grabbed a box, which he will do as soon as he gets close enough to one.

PressButtonWithBoxObjective works pretty much like the base objective to press a button, but first requires the agent executing it to check if they are holding a box, and if they are not, they will push a FindBoxObjective and only proceed to the button(s) once they have grabbed a box. Upon finding a button, they will drop the box they are holding in order to force the button to stay pressed even after they leave it.

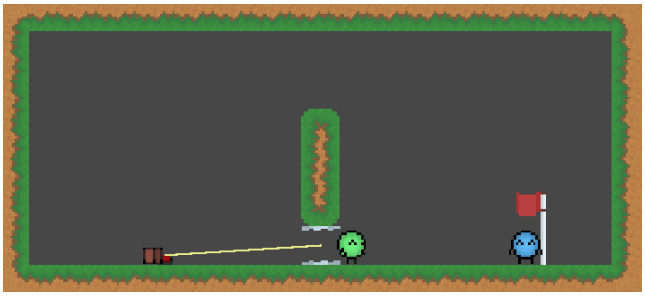

Figure 13: Green agent dropped a box on the button.

As we can see in **Figure 13**, one of the agents left the box on the button, permitting both of them to pass through the door, even though there is only one button.

Intermediate Level 1

Reaching the intermediate levels, it becomes more complicated for the agents to get to the flag. We realized this around here, as there were some problems and simplifications with our previously developed agents, most specifically with the logic behind their objective choices.

Therefore we built the agents aimed at clearing the intermediate levels from the ones developed during the basic levels 2 and 3, but with some optimizations to the previous objectives before adding any new ones.

For example, in most situations it does not make much sense to have an agent try to press a button or look for a box that is on the opposite side of the door the agents are trying to overcome. So we decided to make it so the PressButtonObjective and FindBoxObjective only look for "reachable" buttons and boxes, respectively. This optimization was quite convenient when establishing a priority level for each type of objectives that should be attempted at a given situation.

So for the first intermediate level there is now an additional conflict to overcome in order to reach the flag: the only button that opens up the rightmost door is on the side that is opposite to the agents.

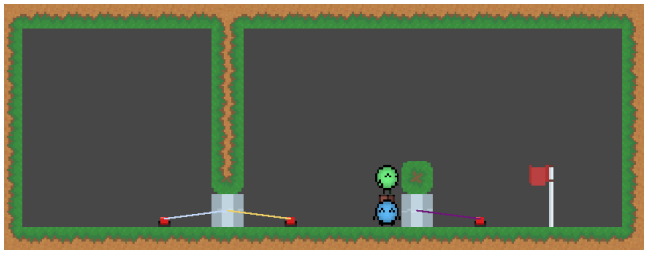

Figure 14: Agents need to collaborate and jump over a door to reach the flag.

In this situation, after the agents have successfully gotten past the first door by cooperating much like on basic level 2 (**Figure 12**)**,** one of them needs to now grab a box in order for the other to jump on top of him and get past the door.

To this end, we designed two objectives that complement each other, appropriately named "JumpOverObjective" and "HelpJumpOverObjective". The first one causes an agent (green in **Figure 14**) to await on the tile where the two consecutive jumps are to be executed until the other agent has noticed that this agent's current objective is to jump over the door, and is ready to assist him to that end.

Upon the agent (blue, who was trying to get past the door), becomes aware that his partner is trying to jump over that very same door, he pushes the HelpJumpOverObjective, which causes him to first look for a box, much like how PressButtonWithBox works, and then await on the correct position until his partner has made it past the door by jumping twice.

Afterwards, the green agent presses the button on the other side of the door so the partner can get past the door and reach the flag as well.

Intermediate Level 2

The intermediate level 2 has already been described in the **Approach** section, as we can see in **Figures 3**,**4** and **5**. No new objectives were added to the agents' behavior from the previous level, however a case was added allowing an agent who currently has a PassDoorObjective and is holding a box to go drop the box off on the button instead of requesting the partner to go press the button.

What's worth noting here perhaps is that an agent dropping off a box on a button just so he alone can get through that door, usually because there is only one reachable button that opens up that door, is not really a collaborative puzzle **by itself** (see advanced level 3, **Figure 18**).

For this reason, we decided not to make it so agents could just go and prioritize opening up doors for themselves, while potentially ignoring their partner.

Intermediate Level 3

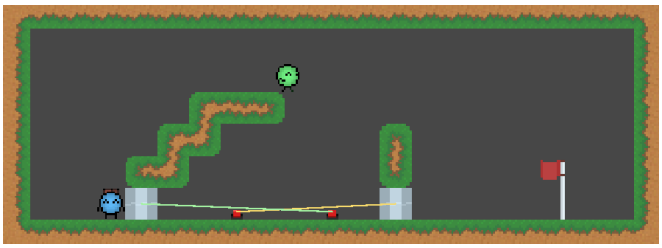

Figure 15: **Agents need to collaborate and jump over a door and then press the buttons to reach the flag.**

As we can see in **Figure 15**, the agents have successfully cooperated so the green agent could jump past the door to go press the button on the right, which opens the first door and lets the blue agent pass through it. Afterwards, to open the second door, the blue agent eventually goes back and drops the box on the leftmost button in a similar fashion as the one described in the previous level.

Advanced Level 1

The advanced levels focus on tackling some rather complicated scenarios, rather than combining a bunch of puzzle mechanisms in a linear fashion, as we saw on the intermediate levels. The complexity of some of these may not be apparent at first, but due to how the agents' behavior was developed incrementally, that is, for one puzzle or level at a time, while also ensuring each new behavior did not break older behavior, it's very easy to see why the agents developed up until this point would fail some or all of the following three levels.

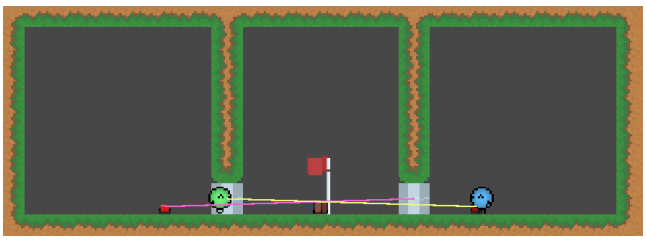

Figure 16: Deadlock scenario where agents need to work in a team properly, so they can both make it to the flag.

In this environment, each one of the agents is locked in a side of the map. The blue agent is on the right side and there's a button that opens up the door on the left side, while the green agent is on the left side with a button that opens the door blocking the blue agent.

In **Figure 16** we can notice that the blue agent is pressing the button to let the other agent go through the door, allowing him to grab the box at the center, bring it to his side and leave it on top of the button on his side. Thus, the door on the right side will remain open, because the green agent dropped the box on the button. After leaving the box the blue agent still needs to press the button on his side to let the green agent pass the door again and reach the flag.

Advanced Level 2

For this level we tried something different. Just like previous levels, this one requires the agents to cooperate so one can jump over a wall measuring up to a maximum of two units. The difference here is that while in the previous ones, we had a door on the ground and a ground tile on top of it, on this one, we have a ground tile and then a door on top of it.

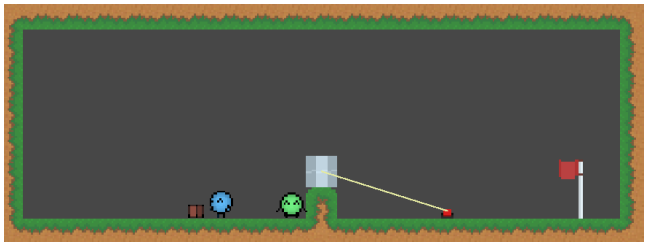

Figure 17: Agent needs to jump over a door that is on top of the ground tile, and later open it so the other agent can jump above the ground tile.

We immediately realized how this scenario made perceiving the door rather awkward, as all the agent had to do before was check on the cell that is in the same direction as the flag, which would be right in this case.

After making it so the agents can also perceive doors in this type of scenario, the blue agent now decides to go get the box to help the green agent jump over the door, as we can see in **Figure 17**.

After that, the green agent just needs to press the button much like before in order to open the door, so the blue agent can jump and go through it and towards the flag.

Advanced Level 3

Finally, we have the last level of our game in which the solution requires the button on the left to be pressed by a box, so both agents can get through the door that is blocking them from reaching the flag. But before this, they need to execute a coordinated jump by making use of a second box that is easily reachable on their left.

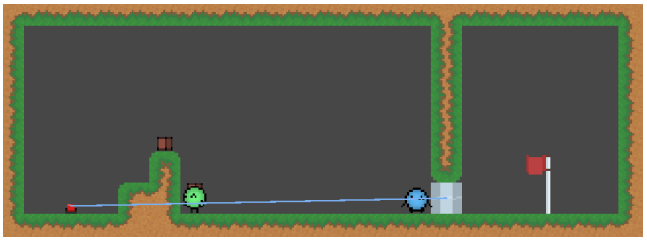

Figure 18: Agents need to drop box on button to reach the flag

Surprisingly, this ended up being one of the most complicated levels to solve, mostly due to the implemented behavior for the agents that solved the previous levels being slightly incomplete for scenarios such as this one.

We eventually did manage to make it so the green agent could identify that he should jump over to reach this button, and from that point on the, blue agent pushed the HelpJumpOverObjective as previously defined. Upon successfully jumping over, the green agent grabs the box sitting on top of the leftmost ground tiles and places it on the box, forcing the door to stay open.

EMPIRICAL EVALUATION

To evaluate the behavior of the developed agents we will be looking into three main metrics. The first is whether the agents manage to solve the puzzles successfully or not.

Another metric we will be considering is the number of actions that were needed to finish the game, which helps us check for efficient solutions. A few examples of actions include taking a step in a given direction, performing a jump, grabbing a box, dropping the box, staying in the same place, etc.

The final main metric we will also look into is the amount of time that was required to solve the puzzles, in order to test the overall performance of the agents.

Aside from these three main straight-forward metrics we will also look into how collaborative or initiative the developed agent's behavior is as well as the quality of the communication between them, when modeling their objectives.

Back when we intended the project to be one where an agent would collaborate with a human to solve the game, we also intended to analyze the different types of communication the human and agent could use to reach a consensus on how to solve the puzzle together.

As we switched to two agents, we will not evaluate this. There is mostly one main type of "communication" used between the agents which was effectively implemented by having each agent "spy" on the other agent's current objective and deciding afterwards on a new objective based on that observation.

COMPARATIVE ANALYSIS

The first idea and suggestion we had to compare different agents was to implement an agent that chooses actions randomly and then see how each newly developed agent compares to that and any of the previous ones.

We quickly realized however, that for our project specifications and the puzzle mechanisms we developed, this is pretty much infeasible, as random agents would likely take hundreds of actions and possibly hours trying to solve even a basic level, let alone a more complex one.

Therefore, we decided to mainly compare our developed agents' evaluation to scripted runs, where the agents knew the exact best set of actions needed to complete that level in the most efficient way possible, as defined by us.

One final thing that is worth mentioning about one of our three metrics to measure the agents' performance, is that, when it comes to agent actions, the action **STAY** where the agent awaits in place for something to happen was not counted towards the total number of actions.

The reason for this is that for some levels where agents had to continuously wait for a relatively long period of time, the update function would be called very frequently, effectively spamming that action up to hundreds of times. We felt that these hundreds of **STAY** actions were irrelevant and made visualizing the evaluation results of the agents a little harder, which is why we did not count them.

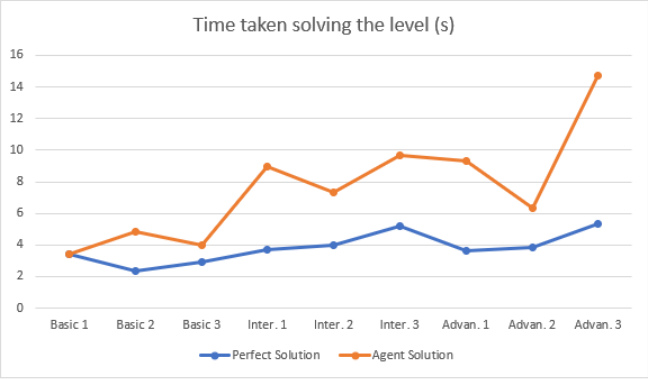

Figure 19: Graph comparing the times taken to solve each level between our agents' solution and the perfect solution.

Observing the graph of times, we notice that our agents' solutions take consistently more time which is obviously expected. An exception to this is the first basic level, which the Agents are able to solve perfectly, due to its simplicity and the fact that it does not require collaboration. We also verify that the time required for the agents to complete a level is on average double of the time required by the perfect solution.

However, we believe this to be due to the levels included in our analysis being quite simple. Due to the agents' need to try out different objectives to find the ones that help them make progress, we believe the times required by the agents to complete a level grow exponentially with the complexity of the levels, and more specifically, with the number of collaborative tasks required to solve the level. This can be observed by the third advanced level, which the agents' take triple the time to complete.

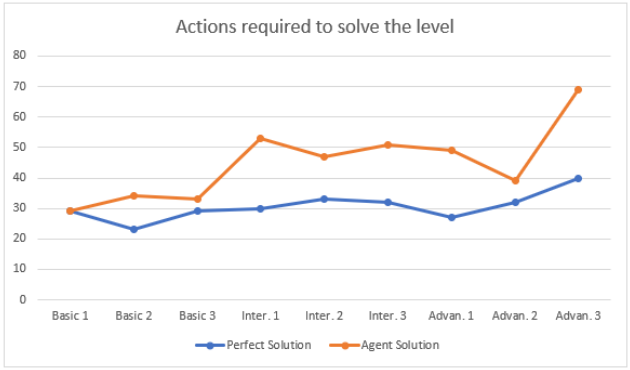

Figure 20: Graph comparing the actions taken to solve each level between our agents' solution and the perfect solution.

Regarding the number of actions required to complete each level, interestingly, the agents require on average 50% more actions than the perfect solutions. We believe this to be due to our decision to not include the special action **STAY** in the action counter. Aside from that, the number of actions required is noticeably proportionate to the time taken.

CONCLUSION

To conclude, even though our agents have a lot of imperfections and would probably end up stuck in some more complex levels that could be created, we are still quite satisfied with how they turned out, being able to complete a lot of levels with quite good efficiency.

REFERENCES

[1] Pomu, n.d. Fireboy And Watergirl In The Forest Temple. [image] Available at: <https://www.pomu.com/game/fireboy-and-watergirl-in-the-forest-temple.html> [Accessed 14 June 2022].

[2] Fireboynwatergirl.com. n.d. Fireboy and Watergirl - Play Now!. [online] Available at: <https://www.fireboynwatergirl.com/> [Accessed 14 June 2022].

[3] Gaips.inesc-id.pt. n.d. About The Game. [online] Available at: <https://gaips.inesc-id.pt/geometryfriends/?page_id=6> [Accessed 14 June 2022].