**Polycraft AI Lab (PAL)**

**DARPA SAIL-ON Program**

**POGO Task Specifications**

Table of Contents

[POGO Task Visualization: 2](#_Toc81317085)

[POGO Novelties: 3](#_Toc81317086)

[Shared Novelties: 3](#_Toc81317087)

[1-1-1 Objects – Type: Axe 3](#_Toc81317088)

[1-2-1 Objects – Type: Fence 3](#_Toc81317089)

[2-1-1 Attributes – Type: Tree Type 4](#_Toc81317090)

[2-2-2 Attributes – Type: Log Production Increase 5](#_Toc81317091)

[2-3-1 Attributes – Type: Arena block type 5](#_Toc81317092)

[3-1-1 Representations – Type: Incorrectly Reported Recipes 5](#_Toc81317093)

[4-2-2 Agents – Type: Thief 6](#_Toc81317094)

[5-5-1 Actions – Type: Pogoist Trades 7](#_Toc81317095)

[6-3-1 Relations – Type: Saplings 8](#_Toc81317096)

[7-2-1 Interactions – Type: Traders Won’t Interact 9](#_Toc81317097)

[POGO Task Design: 10](#_Toc81317098)

[Arena Layout: 10](#_Toc81317099)

[Example steps to complete the POGO Task: 10](#_Toc81317100)

[Recipes: 10](#_Toc81317101)

[Objects: 11](#_Toc81317102)

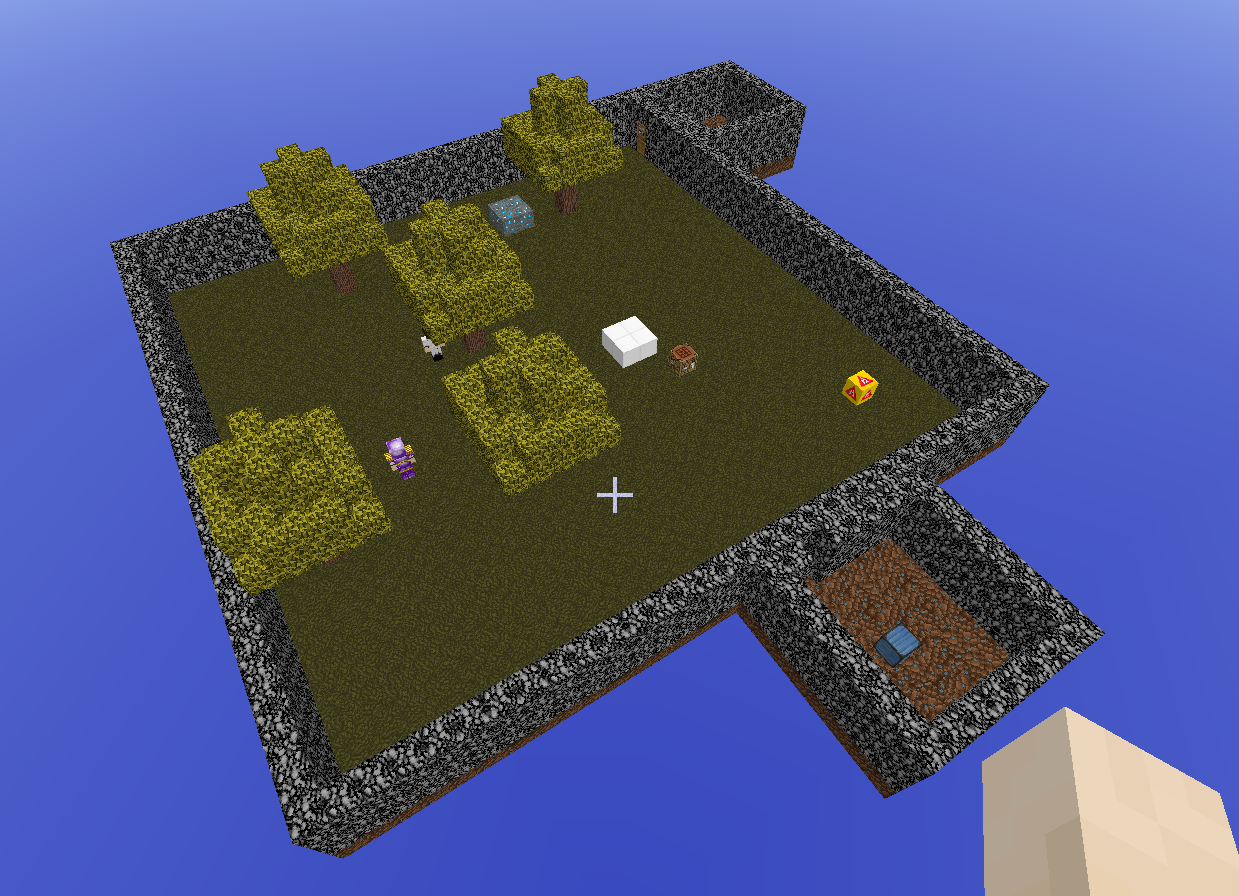
[Agents: 12](#_Toc81317103)

[API: 12](#_Toc81317104)

[Appendix A: Change Log 14](#_Toc81317105)

[2021.08.31 updates: 14](#_Toc81317106)

# POGO Task Visualization:



*Aerial view of Phase 2 POGO Task Arena*

# POGO Novelties:

## Shared Novelties:

### 1-1-1 Objects – Type: Axe

*Phase 2 Updates:*

1. The Axes should only appear in Room 1.
2. The Pogoist should avoid (i.e., walk around and not pickup) the axes on the ground.

*Phase 1 Specifications:*

1. A recipe for a wooden axe is added to the recipe list.
2. The CRAFT: Wooden Axe command works on the crafting table and, if the materials are present, returns SUCCESS and gives the player a wooden axe in their inventory.
3. If a player has a wooden axe in their inventory, SELECT\_ITEM: Wooden Axe returns SUCCESS and their future actions (until they select a different item) are performed as if done with the wooden axe.
4. When an oak tree is cut down by BREAK\_BLOCK after the agent has completed SELECT\_ITEM: Wooden Axe, the stepCost is decreased (~1/2 default) relative to using your empty hand.
5. The agent is able to use TP\_TO to teleport to a floating wooden axe or its coordinates. Once it does this, it picks up the wooden axe by the same mechanism that it picks up floating logs after breaking a tree.
6. If an agent uses a MOVE command to a floating wooden axe, it picks it up by the same mechanism that it picks up floating logs.
7. Axes that are floating or in inventory should be visible with both SENSE\_ALL and SENSE\_SCREEN.

|  |  |
| --- | --- |
| Easy | 1. Wooden axe recipe is always available.  2. Player starts with 1 wooden axe in their inventory hot bar.  3. 0 Wooden axes lying on the ground.  \* This is a static novelty distribution (only 1 available novelty variation). |
| Medium | 1. Wooden axe recipe is always available.  2. Player starts with 0 wooden axes in their inventory hot bar.  3. 1-20 (rng) Wooden axes lying on the ground. Axes can be in any free location (rng). |
| Hard | 1. Wooden axe recipe is always available.  2. Player starts with 0 wooden axe in their inventory hot bar.  3. 0 Wooden axes lying on the ground. |

### 1-2-1 Objects – Type: Fence

*Phase 2 Updates:*

1. The Fences should only appear in Room 1, but this will occur by default since Trees only spawn in Room 1.
2. The Pogoist should move randomly and otherwise do absolutely nothing. We don’t want the Pogoist cutting down the un-fenced trees and impacting performance significantly from Phase 1.
3. Make sure that the spawning of the fences is not compromised by other new objects such as the Pogoist, TVs, Chest, and Ore.

*Phase 1 Specifications:*

1. Fence should completely surround all 8 locations around a tree when present (Cardinal directions plus diagonals).
2. There must be an option for the fence to be 2 layers deep. The player would have to move to the outer fence, break it, move to the inner fence, break it, move to the tree, break it.
3. If player attempts to teleport to a tree surrounded by a fence, the action should fail and they should either stay where they are or teleport to outside the fence.
4. If player attempts to move to a fence, the action should fail and they should stay where they are.
5. Wooden fence should be breakable with a single BREAK\_BLOCK. stepCost should be somewhat less than for breaking a tree.
6. If a player is facing 90 degrees and using our API, they must be able to move through a broken piece of fence without ‘snagging’ on adjacent, unbroken pieces of fence.
7. Confirm that Level 0 rng with trees close to each other does not create problems with fence placement/generation.
8. Fence blocks should be visible with both SENSE\_ALL and SENSE\_SCREEN.

|  |  |
| --- | --- |
| Easy | 1-2 trees (rng) are surrounded by fence. |
| Medium | 3-5 trees (rng) are surrounded by fence. |
| Hard | All 5 trees (rng) are surrounded by fence and the fences are 2 layers deep. |

### 2-1-1 Attributes – Type: Tree Type

*Phase 2 Updates:*

1. The Pogoist should move randomly and otherwise do absolutely nothing. We don’t want it chopping down trees and making this novelty harder than it was in Phase 1.
2. Saplings should be deactivated and not fall when a Tree is chopped down to mimic Phase 1 balancing.

*Phase 1 Specifications:*

1. A variable number of trees (1-5) must be able to be spawned as Jungle Trees.
2. When logs from a Jungle Tree are used to CRAFT: Wooden Planks, it should create Wooden Planks of the ‘Oak’ variant so as not to impact further steps.
3. When COLLECT\_RUBBER is performed on a Tree Tap that is on a Jungle Tree, the stepCost for successful collections should be significantly reduced (at least ~1/2 default).
4. Using BREAK\_BLOCK on a Jungle Tree should produce the same amount of wood as an Oak tree.

|  |  |
| --- | --- |
| Easy | 4 trees are Jungle Trees  \* This is a static novelty distribution (only 1 available novelty variation). |
| Medium | 2-3 trees are Jungle Trees |
| Hard | 1 tree is a Jungle Tree  \* This is a static novelty distribution (only 1 available novelty variation). |

### 2-2-2 Attributes – Type: Log Production Increase

*Phase 2 Updates:*

1. No changes needed relative to the Phase 1 version.

*Phase 1 Specifications:*

1. The number of logs produced from BREAK\_BLOCK on a tree should be determined by a rng as described below.

|  |  |
| --- | --- |
| Easy | 5-6x default logs determined independently per tree. |
| Medium | 3-4x default logs determined independently per tree. |
| Hard | 1-2x default logs determined independently per tree. |

### 2-3-1 Attributes – Type: Arena block type

*Phase 2 Updates:*

1. The walls in the additional Rooms 2-3 should change in the same manner as the Room 1 walls.

*Phase 1 Specifications:*

1. The outer wall block type should be changeable for 0-100% of the blocks in the wall.
2. For all novelty variations, the outer wall must remain unbreakable.
3. The block types that the outer wall changes to must not include trees, logs, crafting tables, or tree taps.
4. There should be a whitelist of available blocks that the walls and floor can change to.
5. Within an instance, all of the blocks that change type should change to the same type.
6. Between instances, the type of block should change to different versions. Currently all new blocks are the same color.

|  |  |
| --- | --- |
| Easy | 1-50% of all outer wall blocks change type. |
| Medium | 51-100% of all outer wall blocks change type. |
| Hard | 100% of all outer wall blocks change to grass blocks. (\*static distribution) |

### 3-1-1 Representations – Type: Incorrectly Reported Recipes

*Phase 2 Updates:*

1. The new Pogo Stick recipe should be treated correctly by this novelty.

*Phase 1 Specifications:*

1. In this version, all of the original recipes still work, but one or more recipes are incorrectly reported in a new orientation.
2. Recipes must be reported inaccurately via SENSE\_RECIPES and SENSE\_ALL.
3. The incorrect recipe(s) reported should be randomly rotated either 90 degrees, 180 degrees, or 270 degrees.
4. The original recipes from the default game should all still be correct and function as before (i.e., if the agent memorized the recipes from the default game, they would experience no change).
5. Make sure that stepCost for FAIL CRAFT commands is less than the stepCost for SUCCESS CRAFT commands.

|  |  |
| --- | --- |
| Easy | 1 recipe is reported incorrectly (rng for which recipe and orientation). |
| Medium | 2 recipes are reported incorrectly (rng for which recipes and orientations). |
| Hard | All recipes are reported incorrectly (rng for orientations). |

### 4-2-2 Agents – Type: Thief

*Novelty characteristics:*

1. Class, attribute, or representation change? New Class of agent.
2. Bonus, neutral, or hazard (maximum available score)? Hazard or Neutral.
3. Ignorable or avoidable while still completing the task? Not typically.
4. Requires accommodation? Yes, in most cases. Not strictly accommodation.
5. Paired novelty where a pre-novelty element is broken? No.

*Value as a shared novelty:*

1. This novelty is not a typical ‘Agent’ category novelty for multiple reasons:
   1. Most new agent class novelties add Agents that perform subsets of the actions seen in pre-novelty. This agent performs multiple new actions including STEAL, STORE, and LOCK.
   2. This novelty is predicted to be very difficult compared to a typical Agent novelty.
   3. This agent illustrates actions that move items from perceptible spaces (i.e., the players inventory) to imperceptible spaces.
   4. This agent has a clearly adversarial element.
2. The agent does not strictly require accommodations and could be ignored either by luck or by randomly going to the safe at an opportune time, but often will not be ignorable and will require accommodation.
3. Collectively, the above characteristics provide numerous opportunities for discussing topics important in novelty generation.
   1. Combinations of multiple novelty levels in single novelties (new agent, new actions, and new interactions are all present in this novelty).
   2. Provides an example of what it looks like when a new agent class is present in post-novelty.
   3. Provides examples of what it looks like when new actions are present in post-novelty.
   4. Provides opportunities to encounter and address engineering or other concerns arising from state changes that appear ‘magical’ because some pre- or post-conditions are not perceptible.
4. It is important to note that the Thief novelty is not representative of the difficulty or complexity of other unshared novelties. The expected complexity of this novelty is in fact a primary reason why this novelty was chosen as a ‘shared novelty’ so that many discussions would be facilitated.

*Specifications:*

1. A new Actor Type called a ‘Thief’ is added.
2. The Thief spawns randomly in the main room (Room 1).
3. The Thief can have the appearance of a Patrol wearing Black Armor.
4. The Thief can MOVE or TP\_TO, just like the Pogoist. What form of movement the Thief uses is based on the movement used by the TA2 Agent.
5. The Thief will move at the same speed as the TA2 agent. It will NOT move at ½ speed like the Pogoist. The Thief will take 1 action for every action the TA2 takes.
6. The Thief’s actions will be as follows:
   1. Move towards TA2.
   2. Whenever the Thief starts its turn within 1 block of the TA2 agent, including diagonals (this will likely be whenever the TA2 does an action that is not a move, such as breaking, crafting, collecting, etc.) the Thief will perform the action ‘STEAL’.
7. STEAL will take items from the TA2’s inventory and put them in the Thief’s inventory, with the items taken based on the difficulties below.
8. Immediately after STEAL, the Thief will move to the Safe and put the stolen items in the safe. The safe should be locked again after each time that the Thief deposits items.
9. It is important that the Thief perform a series of realistic actions to get the items in the safe, such as:
   1. Move towards door.
   2. Use to open the door.
   3. Move towards safe.
   4. Select Key. (It can be assumed that the Thief already has any needed keys for anything)
      1. If unlocked, simply STORE, and then lock.
   5. Use to unlock the safe.
   6. STORE items.
   7. LOCK on the safe to re-lock the safe.
10. After re-locking the safe, the Thief will move to the farthest Trading Villager and remain near the TV for a period of time based on difficulty.
11. After a set amount of time, the Thief begins again and moves back towards the TA2 and steals again, repeating this cycle until the task ends.

|  |  |
| --- | --- |
| Easy | 1. Thief STEALs all of 1 resource from the TA2 agent. Does not steal the pickaxe or tree tap since those are tools.  2. Thief waits at the Trading Villagers for 8 turns before starting to steal again. |
| Medium | 1. Thief STEALs all of 2 resources from the TA2 agent. Does not steal the pickaxe or tree tap since those are tools.  2. Thief waits at the Trading Villagers for 4 turns before starting to steal again. |
| Hard | 1. Thief STEALs all of 3 resources from the TA2 agent. Does not steal the pickaxe or tree tap since those are tools.  2. Thief waits at the Trading Villagers for 2 turns before starting to steal again. |

### 5-5-1 Actions – Type: Pogoist Trades

*Novelty characteristics:*

1. Class, attribute, or representation change? New Attribute of an existing action.
2. Bonus, neutral, or hazard (maximum available score)? Neutral.
3. Ignorable or avoidable while still completing the task? No.
4. Requires accommodation? Yes.
5. Paired novelty where a pre-novelty element is broken? Yes.

*Value as a shared novelty:*

1. This novelty illustrates what an attribute change novelty can look like in a Phase 2 novelty category.
2. This novelty illustrates a strict accommodation novelty.
3. This novelty illustrates a ‘Paired novelty’. The pre-novelty task is specifically broken where the Trading Villagers no longer have trades to offer.
4. Though this novelty does not illustrate the addition of a novel class of actions, examples of new classes of actions can be seen in the Thief shared novelty in the Agents category.

*Specifications:*

1. The Pogoist should now offer all 4 of the trades usually offered across the two TVs.
2. Interacting with the Pogoist in post-novelty should be essentially identical to interacting with TVs in pre-novelty.
3. (Paired novelty to force accommodation) The TVs should no longer have any trades to offer. When the TA2 interacts with the TVs, the interaction should be a SUCCESS, and the TVs should send a short message that says something to the effect of, “No trades available”. Interactions with TVs in post-novelty should be identical to interactions with the Pogoist in pre-novelty, except with a slightly different verbal message being returned.
4. The Pogoist will change its behavior based on difficulty level as described below.
5. The Pogoist should still appear exactly like the Pogoist in pre-novelty in terms of its name, appearance, etc.

|  |  |
| --- | --- |
| Easy | 1. The Pogoist spawns near the TVs. The Pogoist does not move and behaves like a TV. |
| Medium | 1. The Pogoist spawns normally, but immediately walks to the TVs.  2. The Pogoist walks back and forth between a TV and the crafting table, stopping for 4 actions at each place. |
| Hard | 1. The Pogoist behaves exactly as the Pogoist does in pre-novelty. |

### 6-3-1 Relations – Type: Saplings

*Novelty characteristics:*

1. Class, attribute, or representation change? New Attribute of an existing relation.
2. Bonus, neutral, or hazard (maximum available score)? Neutral.
3. Ignorable or avoidable while still completing the task? No.
4. Requires accommodation? Yes.
5. Paired novelty where a pre-novelty element is broken? Yes.

*Value as a shared novelty:*

1. This novelty illustrates what an attribute change novelty can look like in a Phase 2 novelty category.
2. This novelty illustrates a strict accommodation novelty.
3. This novelty illustrates a ‘Paired novelty’. The pre-novelty task is specifically broken where trees do not spawn and must be planted.
4. This novelty illustrates a change in the spatial relationship of tree and saplings.

*Specifications:*

1. The TA2 has to plant any trees that it wants to use by placing saplings.
2. (Paired novelty to force accommodation) Zero Trees spawn on all difficulties.
3. The Pogoist should move randomly throughout the entire instance for all difficulties. It does not need to go after Trees that form from saplings planted by the TA2.
4. For all difficulties the Trees that form from the Saplings must also produce another Sapling when they are chopped down.

|  |  |
| --- | --- |
| Easy | The TA2 starts with 5-10 saplings in their inventory. There are 0 saplings laying on the ground. |
| Medium | The TA2 starts with 0 saplings in their inventory. There are 20-30 saplings laying on the ground. |
| Hard | The TA2 starts with 1 sapling in their inventory. There are 0 saplings laying on the ground. |

### 7-2-1 Interactions – Type: Traders Won’t Interact

*Novelty characteristics:*

1. Class, attribute, or representation change? New Class of interaction.
2. Bonus, neutral, or hazard (maximum available score)? Neutral.
3. Ignorable or avoidable while still completing the task? Sometimes, but not often.
4. Requires accommodation? Yes, most of the time.
5. Paired novelty where a pre-novelty element is broken? No. There is only one novelty, which impacts the task directly on its own.

*Value as a shared novelty:*

1. This novelty illustrates how a novelty that could be very challenging to overcome can still have minimal impact on stepCost when completed optimally.
2. This novelty illustrates an accommodation novelty that does not include a separate paired novelty.
3. This novelty illustrates a new class of interactions where the TVs can now be busy.

*Specifications:*

1. Traders sometimes are unavailable to INTERACT or TRADE with, simulating if they’re busy, helping another customer, on a break, etc.
2. When a TV is busy, the trades that they offer should are simply unavailable, they are not available through the other TV.
3. The result of an INTERACT with a busy TV should be a SUCCESS with a message such as “I’m sorry, but I can’t talk with you at the moment.” This should be similar to the interaction result with the Pogoist in pre-novelty.
4. The result of a TRADE with a busy TV should be a FAIL with the same busy message and the stepCost for the action should be reduced to the same stepCost as the INTERACT command.
5. INTERACT and TRADE attempts with the Pogoist should be unaffected and should always be the same as pre-novelty.

|  |  |
| --- | --- |
| Easy | The TVs should be busy on 50% of INTERACT commands. |
| Medium | The TVs should be busy on 50% of INTERACT commands and 50% of TRADE commands. |
| Hard | The TVs should be busy on 75% of INTERACT commands and 75% of TRADE commands. |

# POGO Task Design:

## Arena Layout:

1. Main room is still 32x32 blocks (including walls).
2. 1-2 Additional rooms of variable size (but significantly smaller than the main room) are generated connected to main room.
3. All objects except the safe are in the primary room. One of the additional rooms will contain the safe object.

## Example steps to complete the POGO Task:

1. **Trees mini-game:** Obtain logs, planks, sticks, a tree tap, and a sack of polyisoprene as in Phase 1.
2. **Safe mini-game:** Obtain Diamond blocks either from the safe or from ore.
   1. Ore: Equip the iron pickaxe and break the diamond ore. Craft diamond blocks from diamonds.
   2. Safe: Obtain the key using collect on the plastic chest. Open the door to adjacent rooms and find the safe. Equip the key and use on the safe to unlock. Collect from the safe to obtain diamonds. Craft diamond blocks from diamonds.
3. **Trading mini-game:** Obtain Titanium blocks by trading platinum, diamonds, or logs.
   1. Platinum: Equip the iron pickaxe and break the platinum ore. Trade directly for titanium.
   2. Diamond: Collect 36 extra diamonds. Trade diamonds for platinum. Trade platinum for titanium.
   3. Logs: Plant new trees with Saplings and break them to collect logs. Trade logs for titanium.
4. **Craft Pogo Stick:** Use wooden sticks, polyisoprene, diamond blocks, and titanium blocks to craft the Pogo Stick.

## Recipes:

1. Pogo stick recipe includes diamond and titanium blocks.
   1. This change requires 2 blocks of Diamond and 2 blocks of Titanium to successfully complete the task.
   2. New Recipe:
      1. [stick, titanium block, stick]
      2. [diamond block, titanium block, diamond block]
      3. [0, sack of polyisoprene, 0]
2. Add Recipe for making diamond blocks.
   * 1. [diamond, diamond, diamond]
     2. [diamond, diamond, diamond]
     3. [diamond, diamond, diamond]
3. All other recipes unchanged from Phase 1.

## Objects:

1. Saplings:
   1. Saplings allow you to grow new trees. They are a mechanism to prevent terminal cases where the agent could run out of trees.
   2. After a tree is chopped down, either by TA2 Agent or External Actor, a sapling will appear in the following 4-7 turns.
   3. Saplings can be planted using “PLACE minecraft:sapling” facing an empty space and a tree will spawn right in front of the player.
2. Doors:
   1. There is a door between the main room and connecting rooms.
   2. The Door must be opened with the USE command.
3. Plastic Chest:
   1. A single plastic chest will spawn in the main room.
   2. The plastic chest contains a “blue key” object that can be obtained by using the COLLECT command when facing the chest.
4. Blue Key:
   1. Obtainable through the plastic chest.
   2. Used to access the safe. When you get to the safe, first “select\_item polycraft:key metadata” then send the USE command. The metadata is optional, but can be used to select a key with a specific color if you have multiple keys.
   3. SENSE\_ALL when the key is in inventory includes its color attribute.
5. Safe:
   1. Spawns in one of the additional rooms.
   2. Initially the safe is in a locked state. It can only be unlocked by using the blue key on it. Once the safe is unlocked, TA2 can use the COLLECT command to get 18 diamonds.
6. Diamond Ore:
   1. Diamond ore will spawn in a cluster of 2-4 blocks in the main room.
   2. Each diamond ore block will drop 9 diamonds.
   3. Requires the iron pickaxe to be selected first. Otherwise the BREAK\_BLOCK action will fail.
7. Platinum Blocks:
   1. Platinum blocks will spawn in a cluster of 2-4 blocks in the main room.
   2. Each platinum block will drop 1 platinum block.
   3. Requires the iron pickaxe to be selected first. Otherwise the BREAK\_BLOCK action will fail.
8. Iron Pickaxe:
   1. The Agent starts with an iron pickaxe in their inventory that is required to be selected to mine the diamond ore and platinum blocks.

## Actors:

1. Trading Villagers:
   1. Two traders will spawn near each other. They are identifiable through different id’s and armor.
   2. These agents don’t move.
   3. INTERACT [id] to obtain the trades available through each agent. There are 4 trades and each agent randomly gets two. The response is the same format as sense\_recipes.
   4. TRADE [id] [item] [item\_count] to perform a trade action through a trader.
   5. The following ‘Trades’ are available:
      1. *TA2 receives:* 1 Titanium block – *TA2 gives:* 1 Platinum block.
      2. *TA2 receives:* 1 Titanium block – *TA2 gives:* 10 logs.
      3. *TA2 receives:* 9 Diamonds – *TA2 gives:* 2 Platinum blocks.
      4. *TA2 receives:* 1 Platinum block – *TA2 gives:* 18 Diamonds.
2. Pogoist:
   1. In each pre-novelty instance, 1 Pogoist spawns in the main room.
   2. It has purple armor equipped and is visually distinct from other actors.
   3. The Pogoist takes one action for every 2 actions the TA2 Agent takes.
   4. Many actions do not advance time and do not trigger the Pogoist to take an action. These include the following: SENSE commands, CHECK\_COST commands, LOOK commands, TURN, and REPORT\_NOVELTY.
3. The Pogoist will use the movement type that the TA2 Agent uses, either MOVE or TP\_TO.
4. Goals: The main goal for the Pogoist is to craft a Pogo Stick. Every time it chops down a tree, it goes to the crafting table and crafts planks, sticks, or a tree tap. It always attempts to acquire resources in a specific order: wood, rubber, diamond, platinum, titanium.

## API:

1. NOP
   1. The command NOP can be sent to advance time by one turn without taking an action for the TA2 agent. This could be useful if the Agent wants to wait for a specific future state without changing location. NOP does not have a stepCost.
2. DELETE [block/item name] [integer/how many to delete]
   1. This is to prevent edge cases where an agent may fill their inventory with seeds, logs, etc. and be unable to proceed.
   2. API Command: DELETE [block/item name]
   3. StepCost: Free
3. SELECT\_ITEM [block/item name]
   1. You can now pass no parameters to deselect the current item
4. USE
   1. Previously USE\_HAND
   2. This will perform the “USE” action with the currently selected item as a parameter.
5. PLACE [block/item name]
   1. Previously “PLACE\_BLOCK”
   2. Used to attempt to place a block or item
6. COLLECT
   1. Replaces “EXTRACT\_RUBBER”
      1. Step cost for extracting rubber will remain the same
   2. Collecting from other objects has a much lower step cost of 1200
7. TRADE [entityID] [item 1] [qty] [item 2] [qty] [item 3] [qty] [item 4] [qty] [item 5] [qty]
   1. Similar to craft but used for trading. Trades have up to 5 slots in and 5 slots out. But we are only using one slot in and one slot out currently.
8. INTERACT [entityID]
   1. This command is similar to SENSE\_RECIPES in that it returns the list of available trades with the Trading Villager.
9. Partial Observability with SENSE commands:
   1. SENSE commands only sense objects in the same room as the player
      1. This includes the edge wall blocks of the room, but not adjacent rooms.

# Appendix A: Change Log

- An up-to-date change log for Polycraft AI Lab can be found on GitHub in the **ChangeLog.txt** file.