Gold Miner

——A Simulation World in NetLogo

CONTENTS

2 Research Question

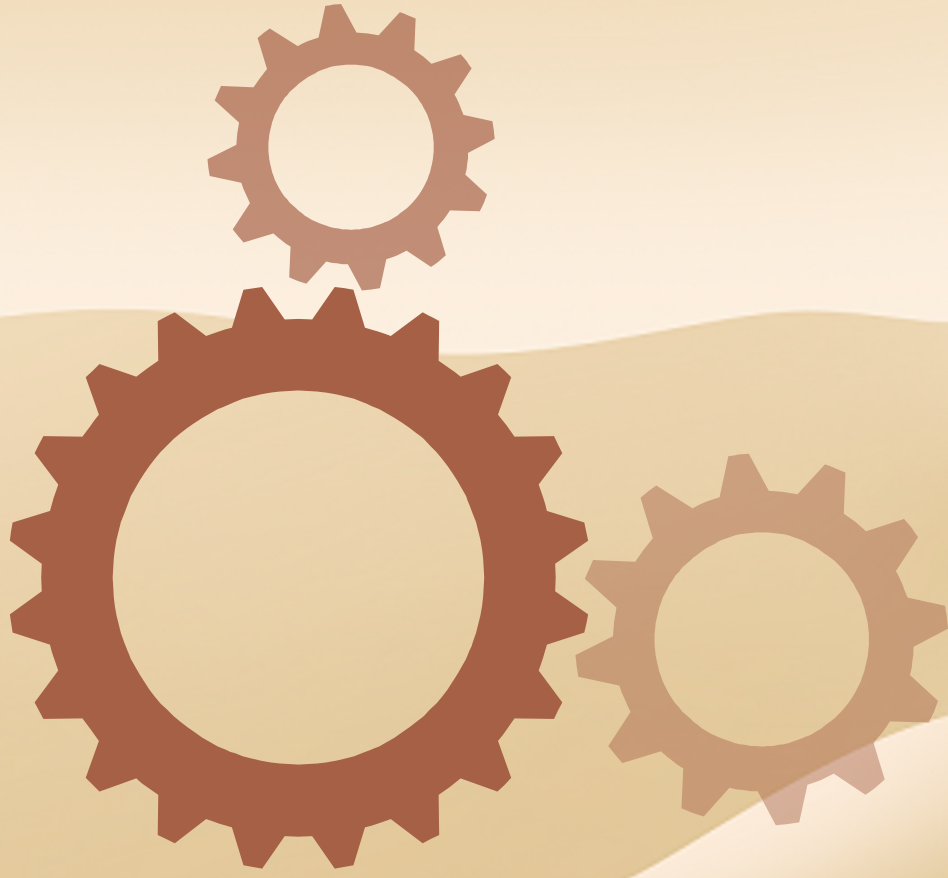
01 Research Question

Question:

What’s the best strategy for the miner to make the most money with the least payout with limited energy?

Purpose:

Build a simulation environment for players and provide some visual data to help them find the best way of combining two mining strategies.

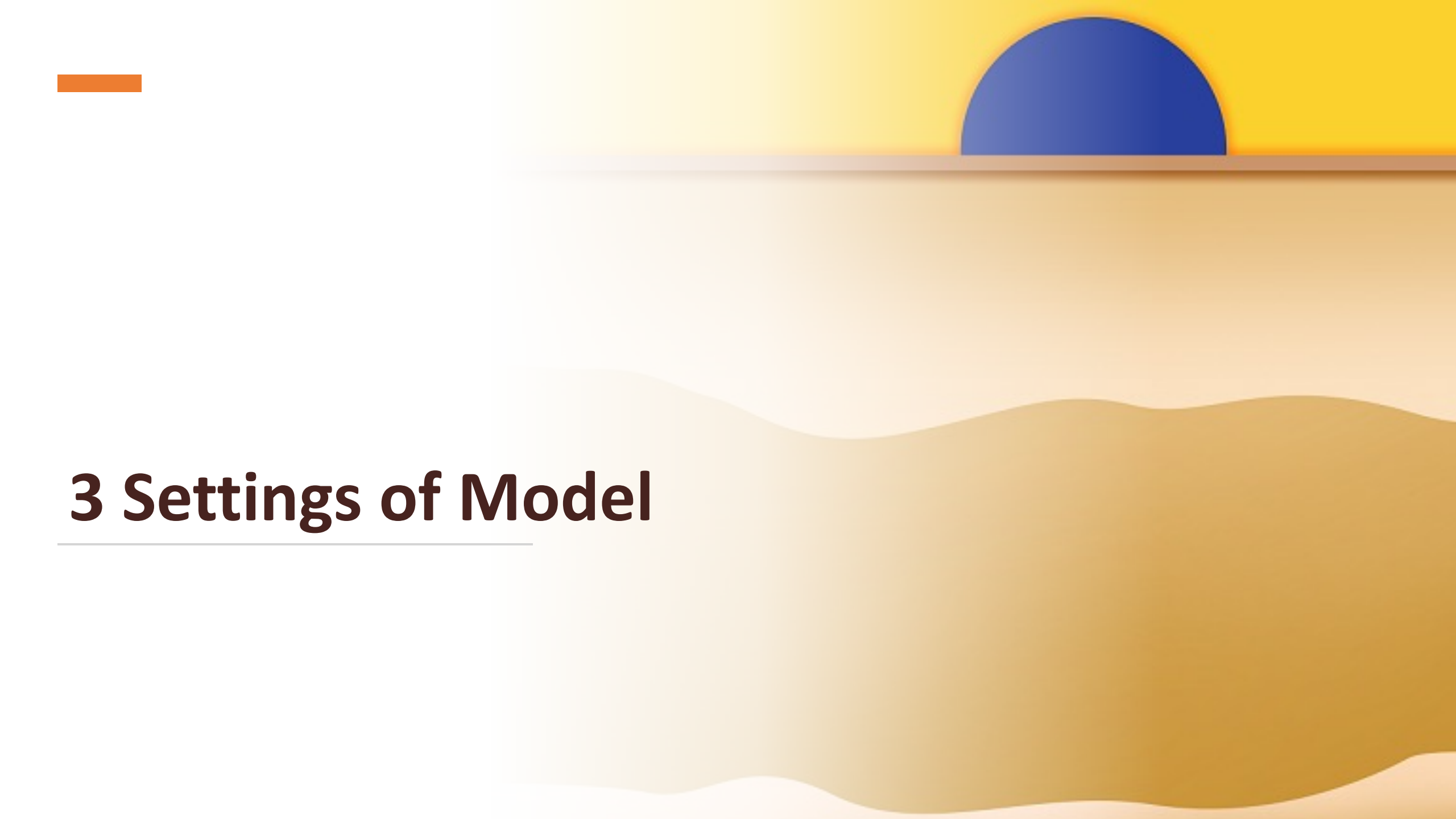


2 Simulation Method

02 Simulation Method

Agent-based Modeling --

NetLogo



3 Settings of Model

3.1 Breed

```
breed [miners miner]
breed [diamonds diamond]
breed [stones stone]
breed [golds gold]
```

Four types of turtles:

Miner

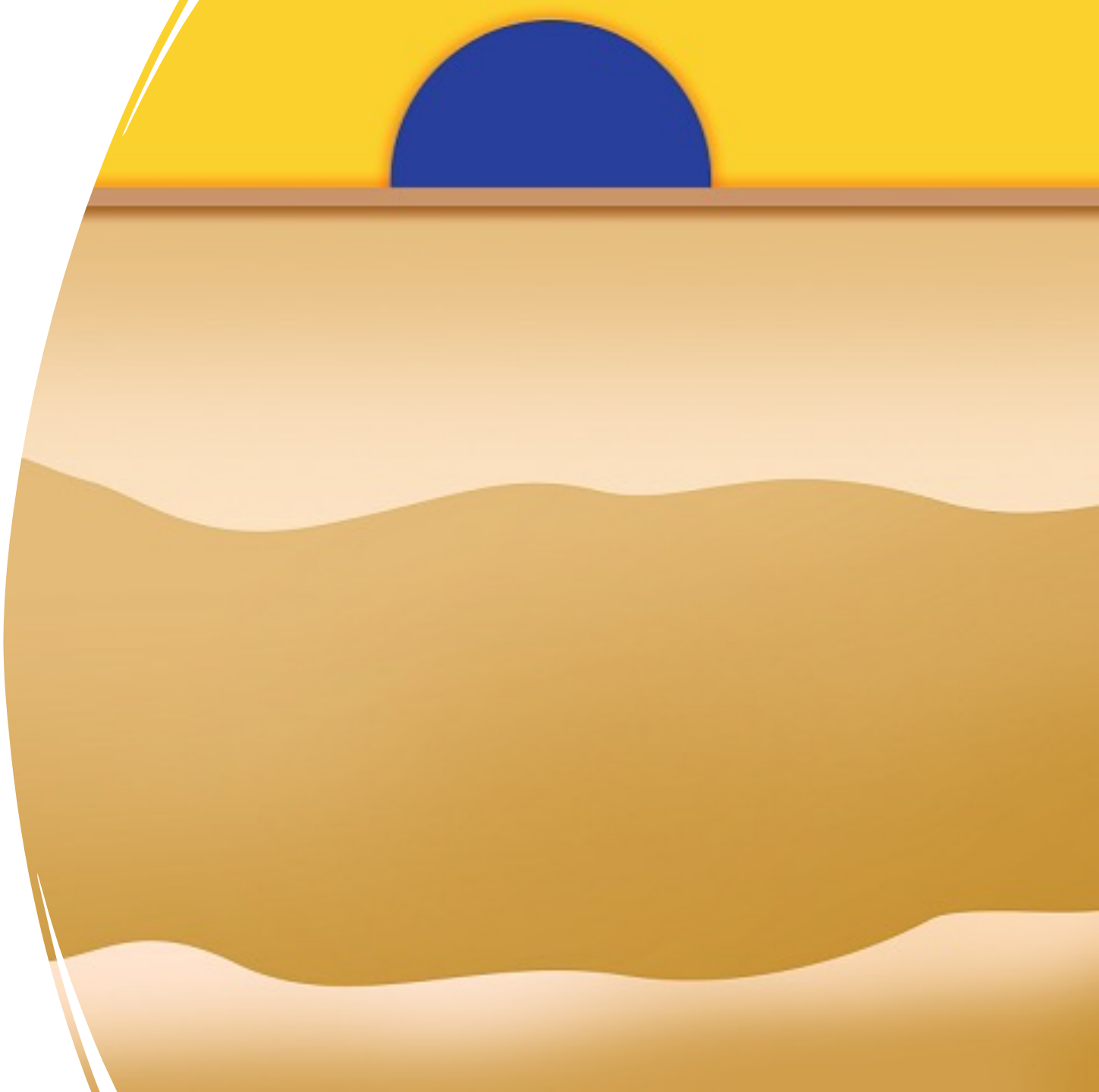
Diamond

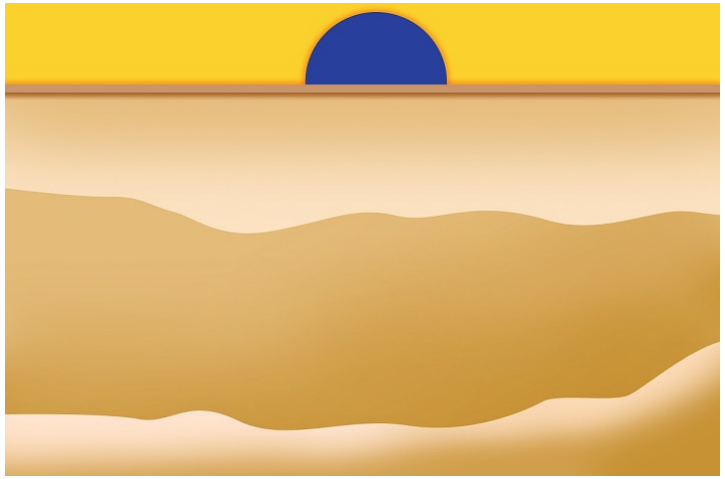
Gold

Stone

3.2 Globals

- **target-treasure:** the collective name for "diamond" "gold" "stone" as the miner's target
- **step:** the step of the miner
- **total_step:** the total step of the miner
- **a:** the xcor of the miner
- **b:** the ycor of the miner
- **Y1:** the marginal gain in plot
- **Y2 :** the total gain in plot





名称	修改日期	类型	大小
bkg	1/5/2022 16:54	PNG 文件	263 KB
GOLDMINER	10/5/2022 19:41	NetLogo Model	25 KB

3.3 to setup

Import the background picture：

- **Place both picture and Netlogo file in the same folder**
- **import-drawing "\\bkg.png"**

3.4 to setup

Three subprograms:

1

add- miner:
set miner to find treasure

2

add_treasure:
set target-treasure including diamond, stone, golds

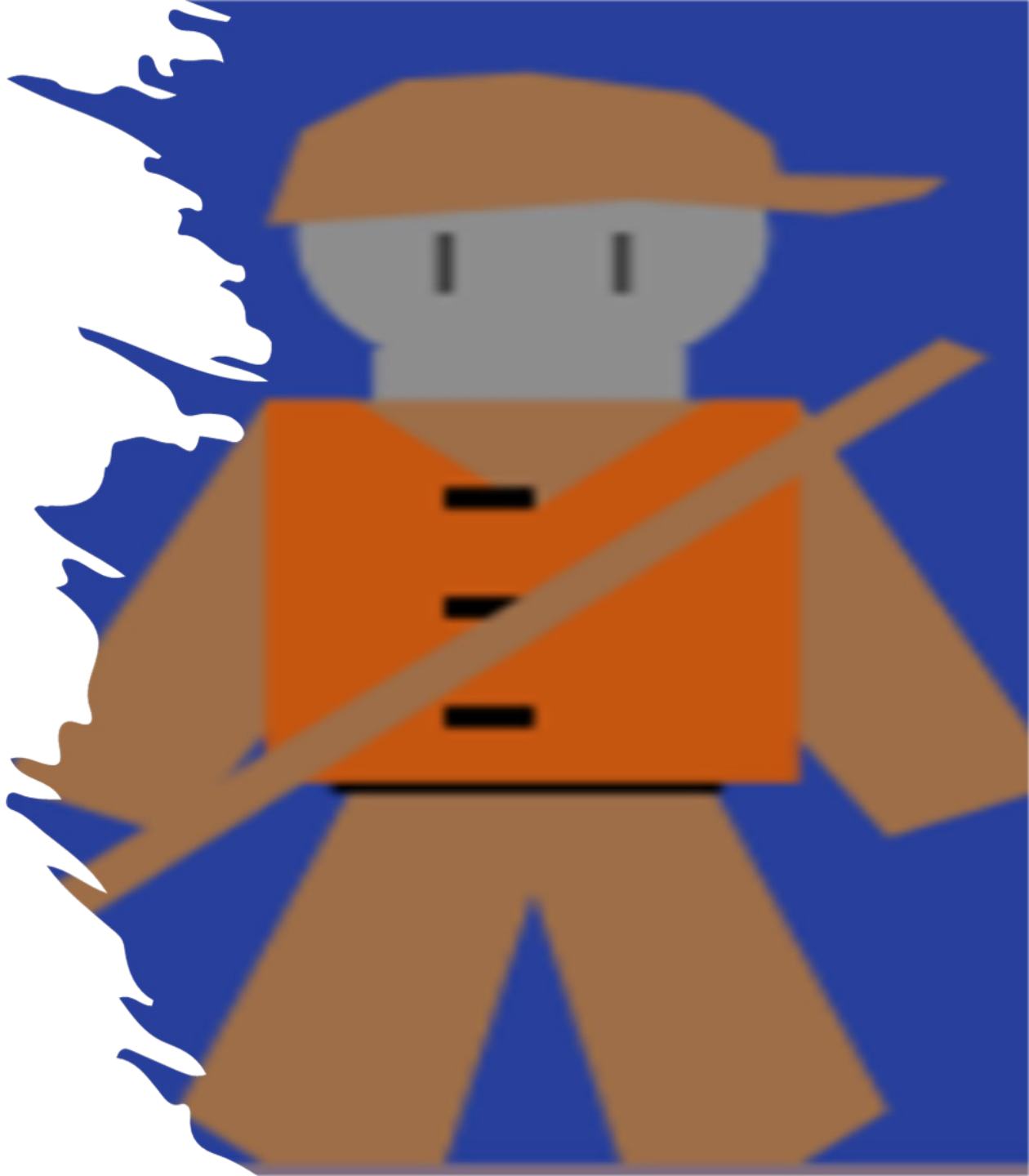
3

buy_entrance_ticket :

set the entrance fee and corresponding energy & the initial treasure setting

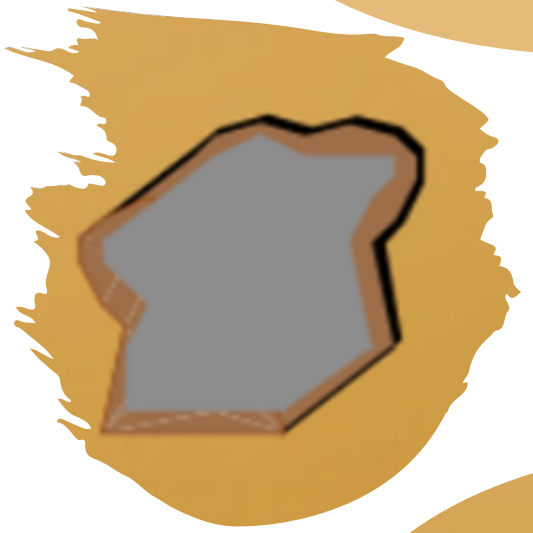
3.4.1 add_miner

```
to add_miner
  create-miners 1[
    set shape "miner"
    set color 24
    set size 5
    setxy 31 36
    set energy energy
    set label energy
    set entrance_fee entrance_fee
  ]
end
```



3.4.2 add_treasure

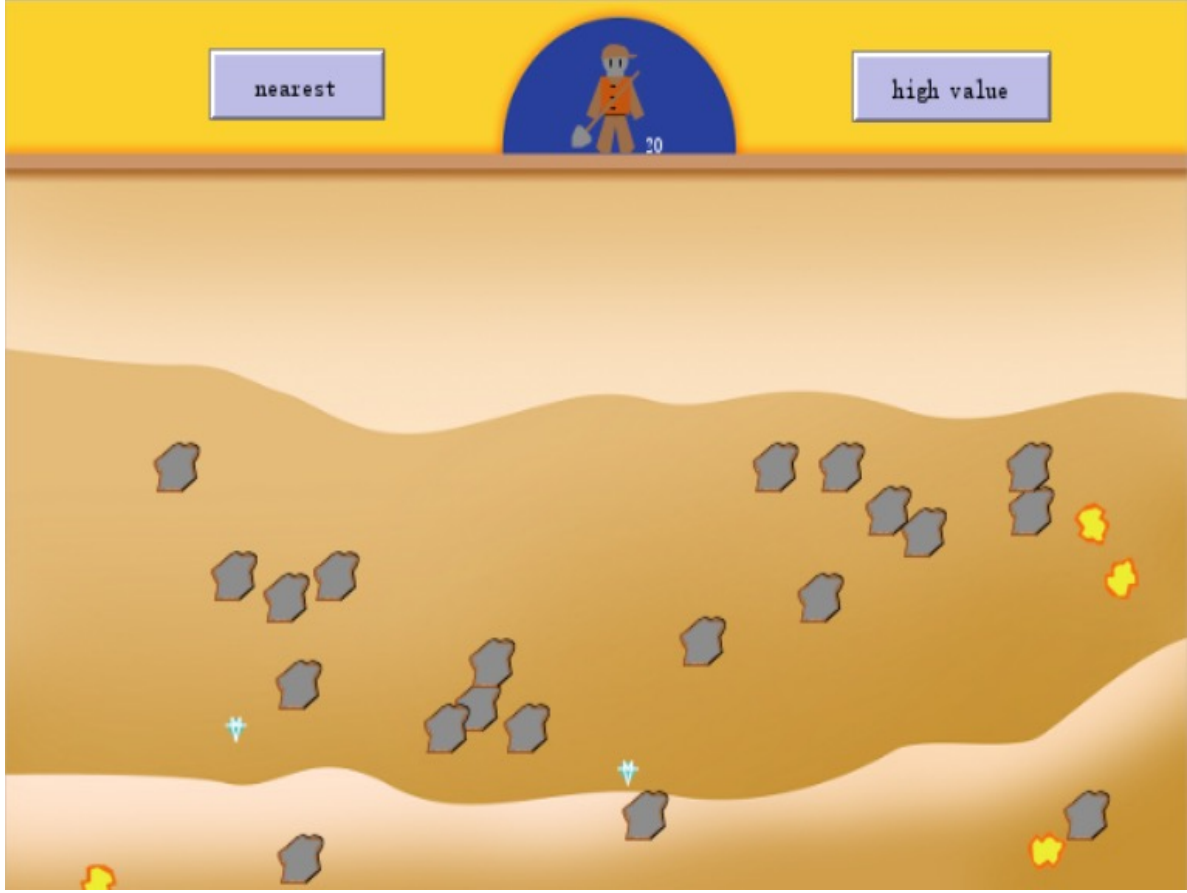
```
to add_treasure ;; add miner, diamonds, golds, stones in treasure field
  create-diamonds num_diamond[
    set shape "diamond"
    set size 2
    setxy random-xcor random 20 ;; only appear in middle and lower layers
    set heading 0
    set value 200
  ]
  create-golds num_gold[
    set shape "gold"
    set size 5
    setxy random-xcor random 20 ;; only appear in middle and lower layers
    set value 100
  ]
  create-stones num_stone[
    set shape "stone"
    set color 23
    set size 3
    setxy random-xcor random 20 ;; only appear in middle and lower layers
    set value 50
  ]
end
```

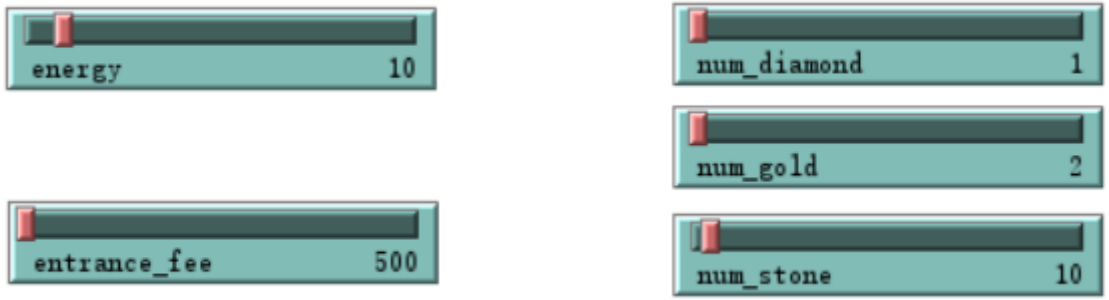


3.4 .3 buy_entrance_ticket

```
to buy_entrance_ticket
  if entrance_fee = entrance_fee [
    set energy entrance_fee / 50
    set num_diamond entrance_fee / 500
    set num_gold entrance_fee / 250
    set num_stone entrance_fee / 50
  ]
end
```

- Adjust the ENTRANCE_FEE slider to buy entrance ticket
- $500 = 10 energy = 1 diamond + 2 gold + 10 stones
- The number of ENERGY, NUM_DIAMONG, NUM_GOLD and NUM_STONE will change as ENTRANCE_FEE change





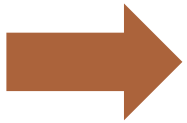
4 Data generated

We set 2 strategies for player to choose

4.1.1 strategy_1

Strategy_1 : The miner go to the **nearest** target treasure

```
to strategy_1
  ask miners [
    pd
    set a xcor
    set b ycor

    set target-treasure min-one-of other turtles [distance myself]
    ask target-treasure [
      set step (distancexy a b)
      set total_step total_step + step
    ]

    move-to target-treasure
    ask other turtles-here [die]
```



Step:

1. Locate the miner
2. Find the nearest targets
3. Calculate the Euclidean Distance (named **step)**
4. Calculate the **total step** taken by the miner
5. Miner **move to the nearest target-treasure**
6. **Mining**

4.1.2 strategy_1

Strategy_1 : The miner go to the **nearest** target treasure

```
    set energy energy - 1
    set label energy
    if energy < 0 [die]

    if energy = 0 [stop]
  ]

  set Y1 ((num_diamond * 200 + num_gold * 100 + num_stone * 50)
    - count turtles with [shape = "diamond"] * 200
    - count turtles with [shape = "gold"] * 100
    - count turtles with [shape = "stone"] * 50) / total_step
  set Y2 ((num_diamond * 200 + num_gold * 100 + num_stone * 50)
    - count turtles with [shape = "diamond"] * 200
    - count turtles with [shape = "gold"] * 100
    - count turtles with [shape = "stone"] * 50
    - entrance_fee) / total_step

  tick
end
```

Step:

7. Set the energy, each mining cost 1 energy
8. If energy exhausted, then exit the game
9. Stop the whole process

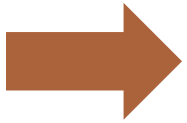
10. Calculate marginal gain & total gain

4.2.1 strategy_2

Strategy_2 : The miner go first to the **highest** value of target treasure

diamond > gold > stone

```
to strategy_2

  ask miners [
    pd
    set a xcor
    set b ycor

    set target-treasure max-one-of turtles with [breed != miners] [value]
    ask target-treasure [
      set step (distancexy a b)
      set total_step total_step + step
    ]

    move-to max-one-of turtles with [breed != miners] [value]

    set energy energy - 1
    set label energy
    if energy < 0 [die]

    ask diamonds-here [die]
```



Step: (repeat some strategy_1 settings)

1. Locate the miner
2. Find the highest value targets-random
3. Calculate the Euclidean distance (named **step)**
4. Calculate the **total step** taken by the miner
5. Miner **move to the highest target-treasure**
6. Mining

4.2.2 strategy_2

Strategy_2 : The miner go first to the **highest** value of target treasure

diamond > gold > stone

```
    if count turtles with [shape = "diamond"] = 0 [
      ask golds-here [die]
      if count turtles with [shape = "gold"] = 0 [
        ask stones-here [die]]
    ]

    if energy = 0 [stop]
  ]

  set Y1 ((num_diamond * 200 + num_gold * 100 + num_stone * 50)
    - count turtles with [shape = "diamond"] * 200
    - count turtles with [shape = "gold"] * 100
    - count turtles with [shape = "stone"] * 50) / total_step
  set Y2 ((num_diamond * 200 + num_gold * 100 + num_stone * 50)
    - count turtles with [shape = "diamond"] * 200
    - count turtles with [shape = "gold"] * 100
    - count turtles with [shape = "stone"] * 50
    - entrance_fee) / total_step

  tick
end
```

Step: (repeat some strategy_1 settings)

7. When there are no diamond, mine gold
8. When there are no gold, mine stone

10. Stop the whole process
10. Calculate marginal gain & total gain

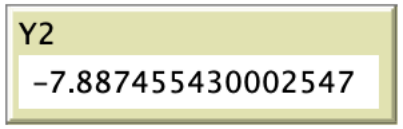
5 Finding and conclusion

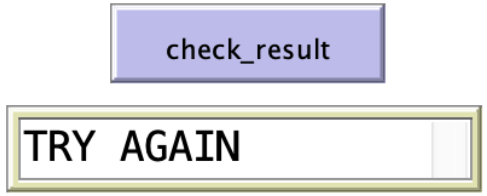
5.1 check_result

```
to check_result
  ifelse Y2 * total_step > 0
  [output-print (word "CONGRATULATIONS")]
  [output-print (word "TRY AGAIN")]

end
```

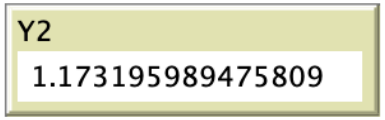
Using **Y2** as an indicator to check the result:

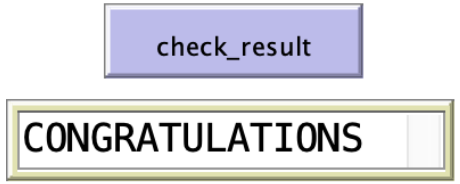
if Y2 is a negative value → don't make the right strategy & show TRY AGAIN





If Y2 is a positive value → the total value is positive, then CONGRATULATIONS

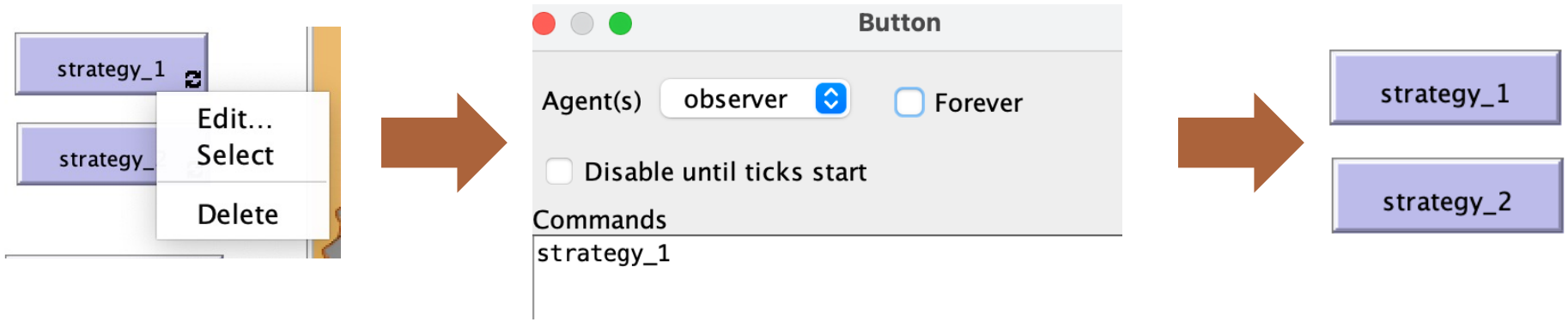




5.1 other findings

We can also use this mini-game to test which strategy is better between players:

1. **Removing forever** in strategy_1 & strategy_2



2. Players can **switch strategies** between Strategy_1 and Strategy_2 **at every step** due to the miner's behavior and the layout of Treasure on the interface

3. To check if they make the successful strategy