Beyond Winning: Code Coverage-Guided Neuroevolution for Robust Game Testing

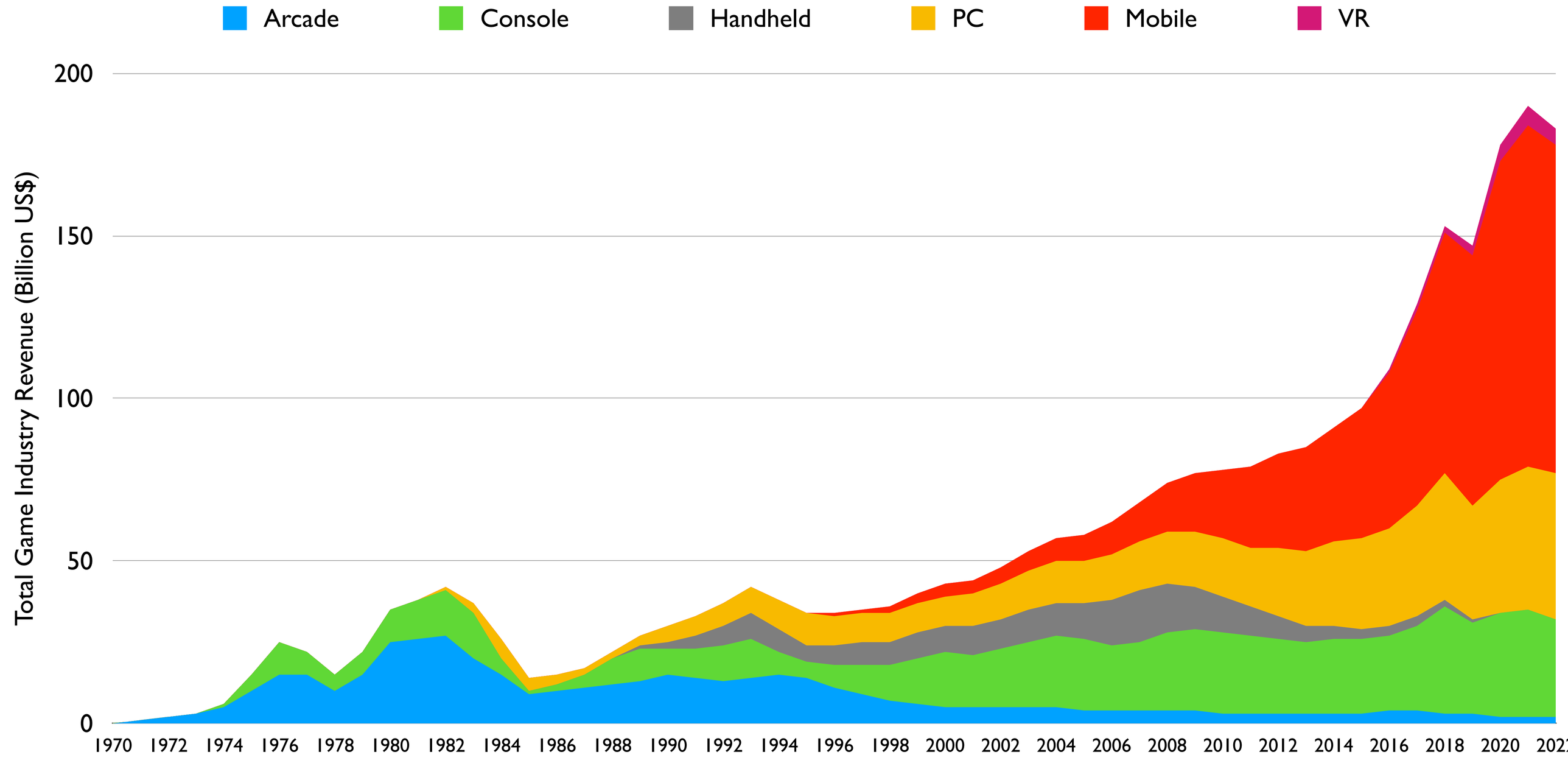
Gordon Fraser
University of Passau


1UP
4420
HIGH SCORE
20000
STAGE 0

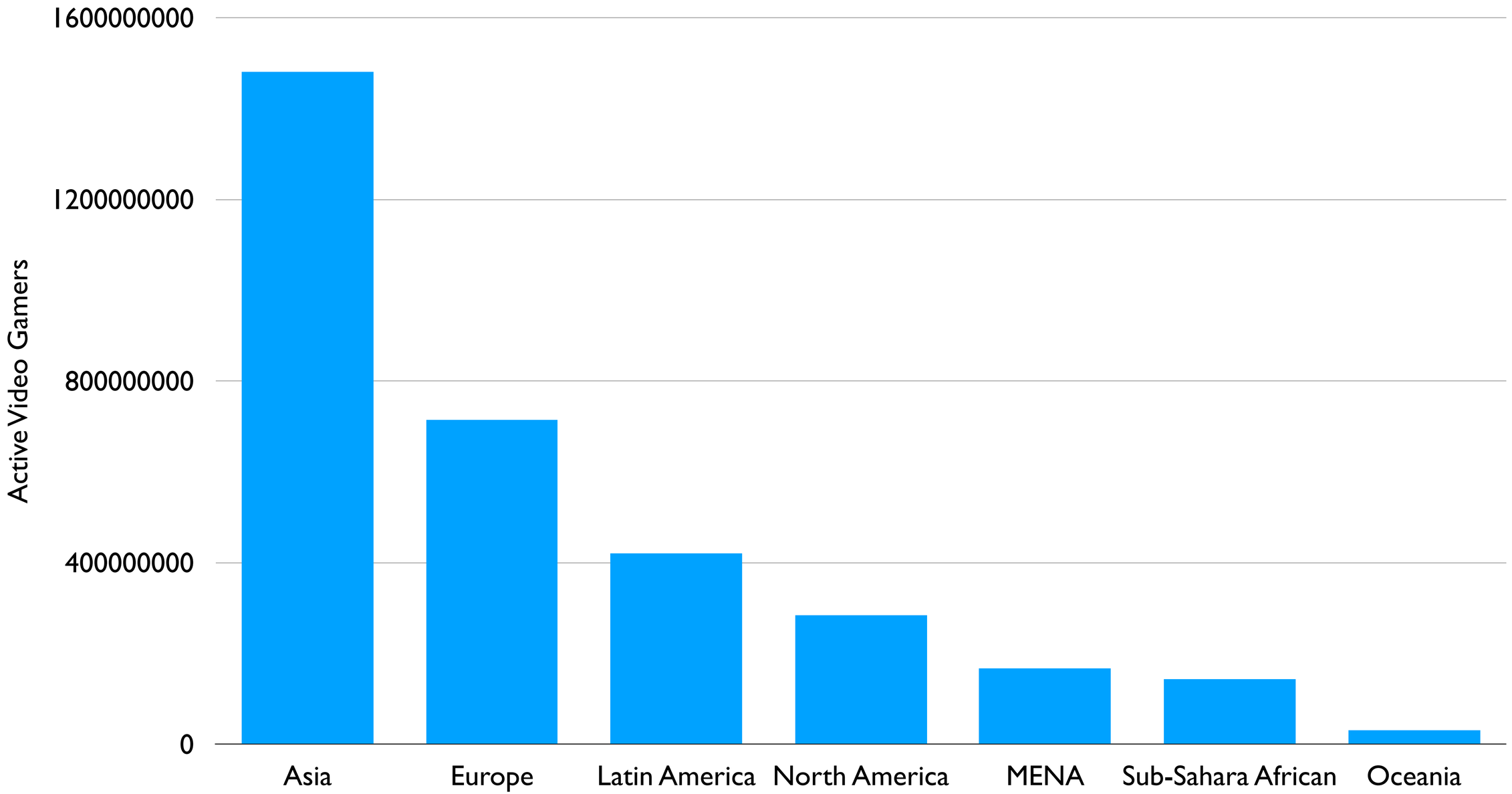

FUT CHAMPIONS
73:44
CLA 1-2 Mil
4 LACROIX
SALAH 21

The frame rate issues in Pokémon Scarlet & Violet are pretty bad. 😭

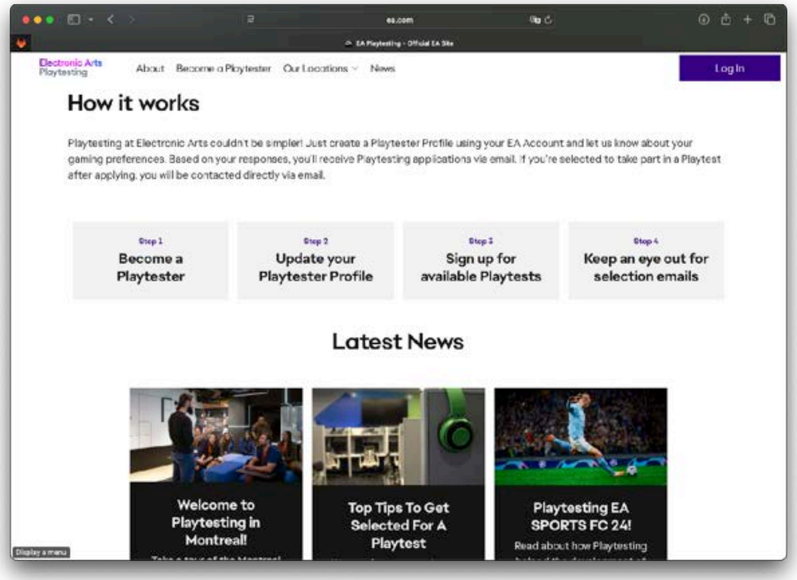
Total Game Industry Revenue (US$, Billions)

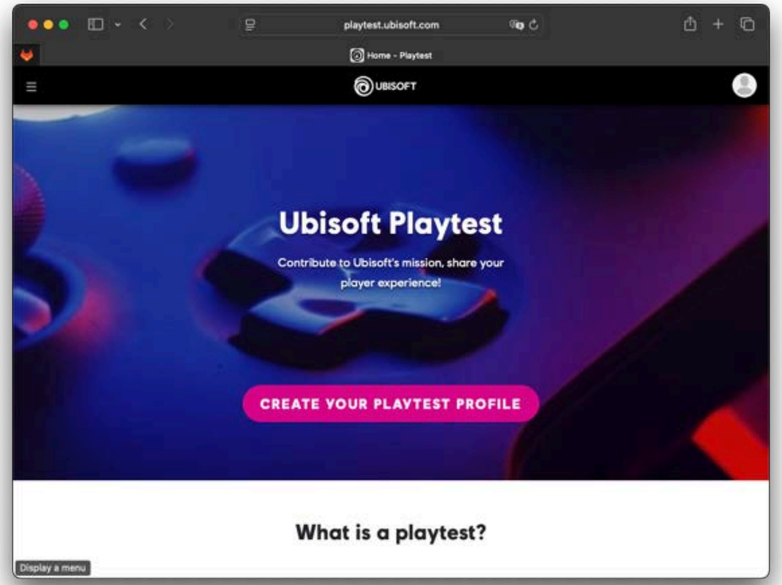


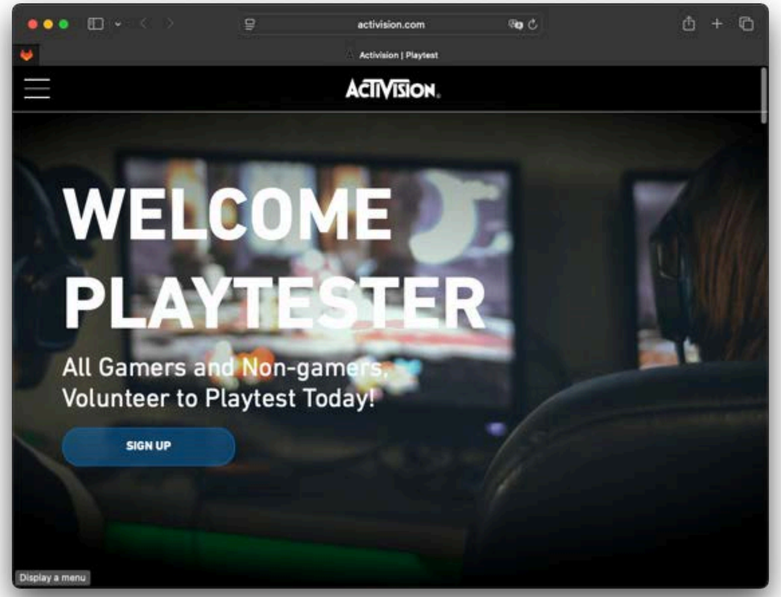
Active Video Game Players

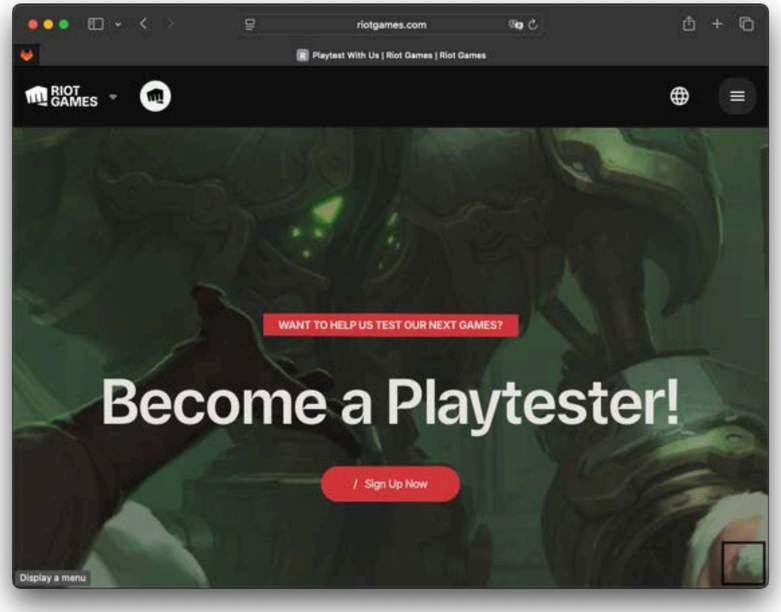


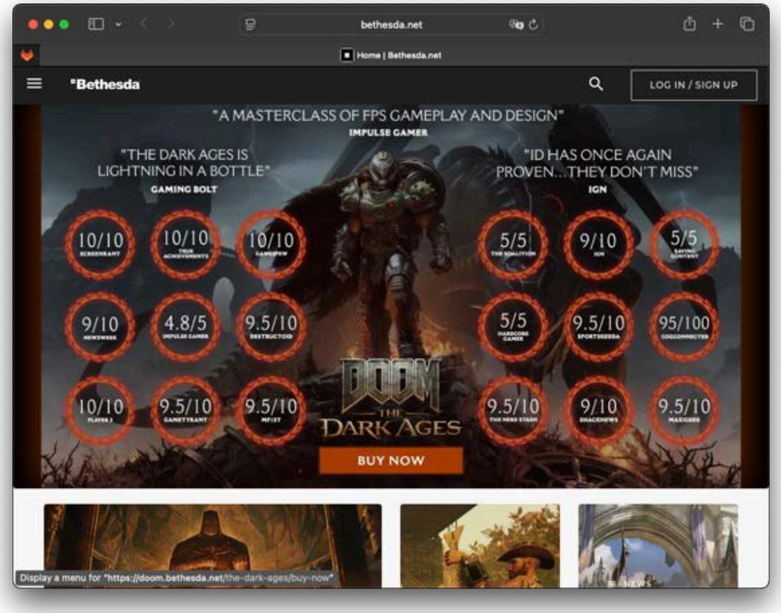
Play Testing

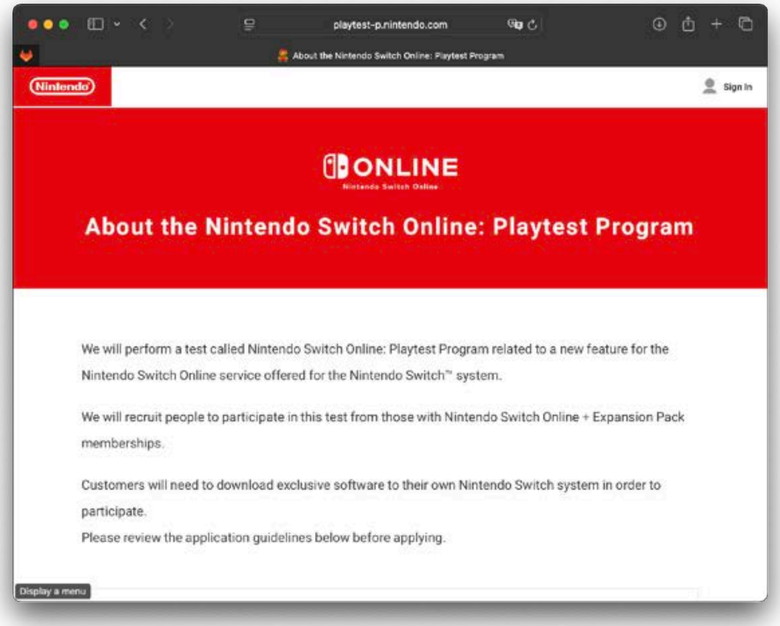

ea.com
Electronic Arts Playtesting
About
Become a Playtester
Our Locations
News
Log In
How it works
Playtesting at Electronic Arts couldn't be simpler! Just create a Playtester Profile using your EA Account and let us know about your gaming preferences. Based on your responses, you'll receive Playtesting applications via email. If you're selected to take part in a Playtest after applying, you will be contacted directly via email.
Step 1
Become a Playtester
Step 2
Update your Playtester Profile
Step 3
Sign up for available Playtests
Step 4
Keep an eye out for selection emails
Latest News
Welcome to Playtesting in Montreal!
Top Tips To Get Selected For A Playtest
Playtesting EA SPORTS FC 24!
Read about how Playtesting
Display a menu


playtest.ubisoft.com
Home - Playtest
UBISOFT
Ubisoft Playtest
Contribute to Ubisoft's mission, share your player experience!
CREATE YOUR PLAYTEST PROFILE
What is a playtest?
Display a menu


activision.com
Activision | Playtest
ACTIVISION.
WELCOME PLAYTESTER
All Gamers and Non-gamers, Volunteer to Playtest Today!
SIGN UP
Display a menu


riotgames.com
Playtest With Us | Riot Games | Riot Games
RIOT GAMES
WANT TO HELP US TEST OUR NEXT GAMES?
Become a Playtester!
Sign Up Now
Display a menu


bethesda.net
Home | Bethesda.net
"Bethesda
LOG IN / SIGN UP
"A MASTERCLASS OF FPS GAMEPLAY AND DESIGN"
IMPULSE GAMER
"THE DARK AGES IS LIGHTNING IN A BOTTLE"
GAMING BOLT
"ID HAS ONCE AGAIN PROVEN...THEY DON'T MISS"
IGN
10/10
10/10
10/10
5/5
9/10
5/5
9/10
4.8/5
9.5/10
5/5
9.5/10
95/100
10/10
9.5/10
9.5/10
9.5/10
9/10
9.5/10
DOOM THE DARK AGES
BUY NOW
Display a menu for "https://doom.bethesda.net/the-dark-ages/buy-now"

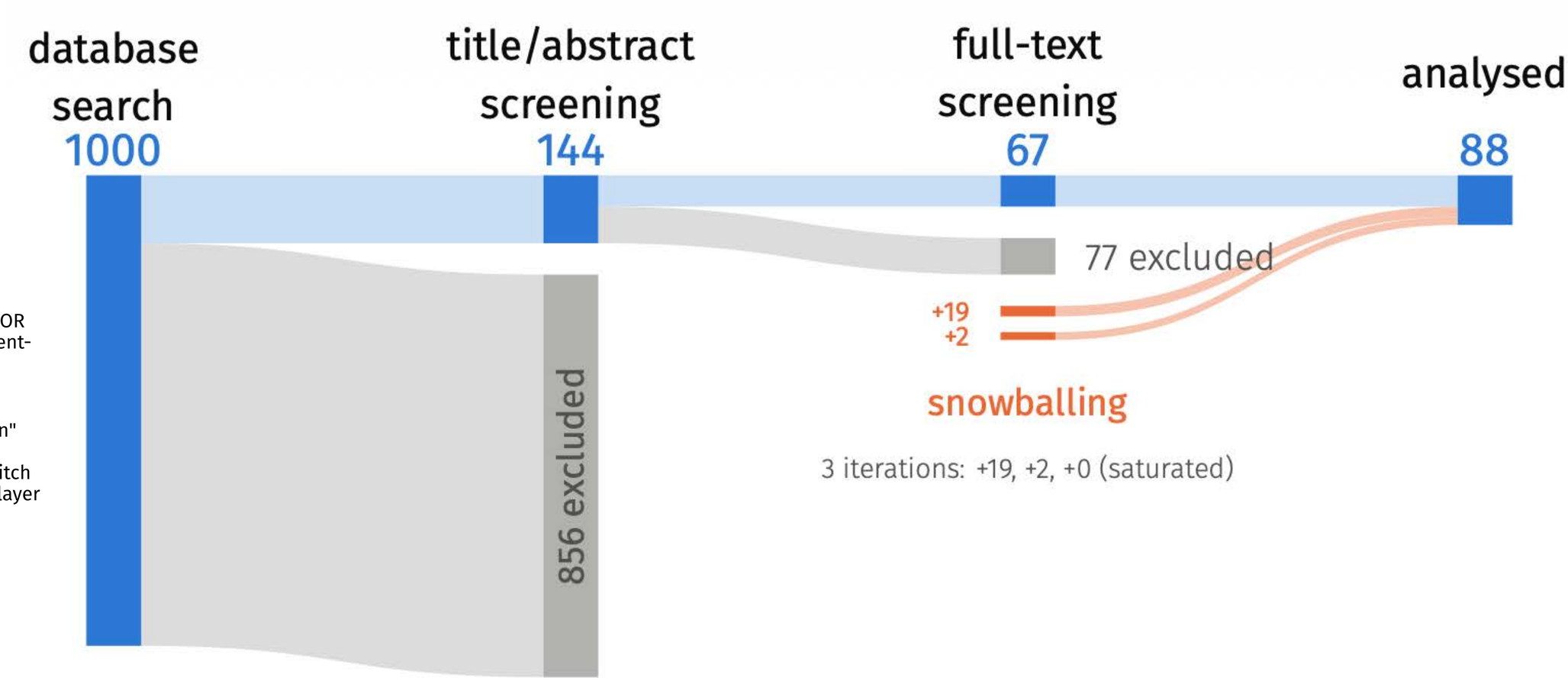

playtest-p.nintendo.com
About the Nintendo Switch Online: Playtest Program
Nintendo
Sign In
ONLINE
Nintendo Switch Online
About the Nintendo Switch Online: Playtest Program
We will perform a test called Nintendo Switch Online: Playtest Program related to a new feature for the Nintendo Switch Online service offered for the Nintendo Switch™ system.
We will recruit people to participate in this test from those with Nintendo Switch Online + Expansion Pack memberships.
Customers will need to download exclusive software to their own Nintendo Switch system in order to participate.
Please review the application guidelines below before applying.
Display a menu

Literature Survey

("game testing" OR "testing of games" OR "playtesting" OR "play-testing" OR "agent-based testing")

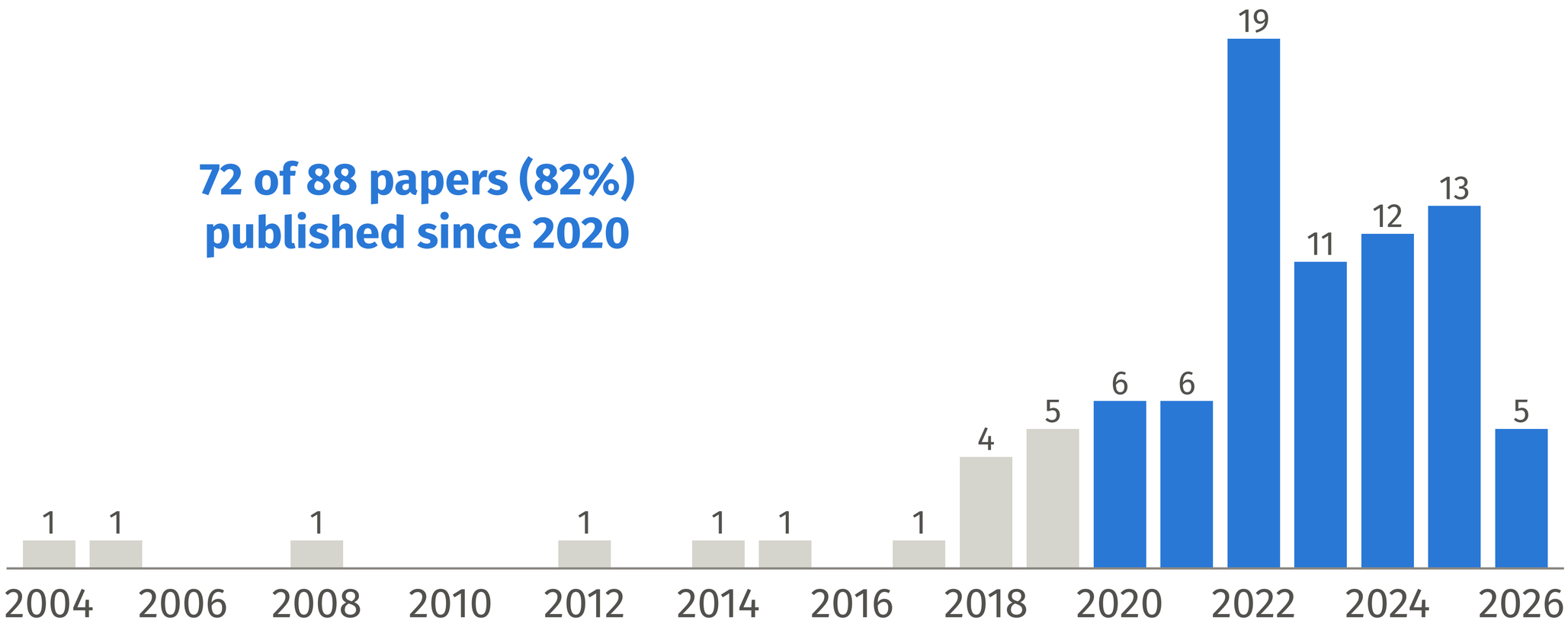
AND

("test generation" OR "input generation" OR "oracle inference" OR "oracle generation" OR "bug detection" OR "glitch detection" OR "crash detection" OR "player experience")

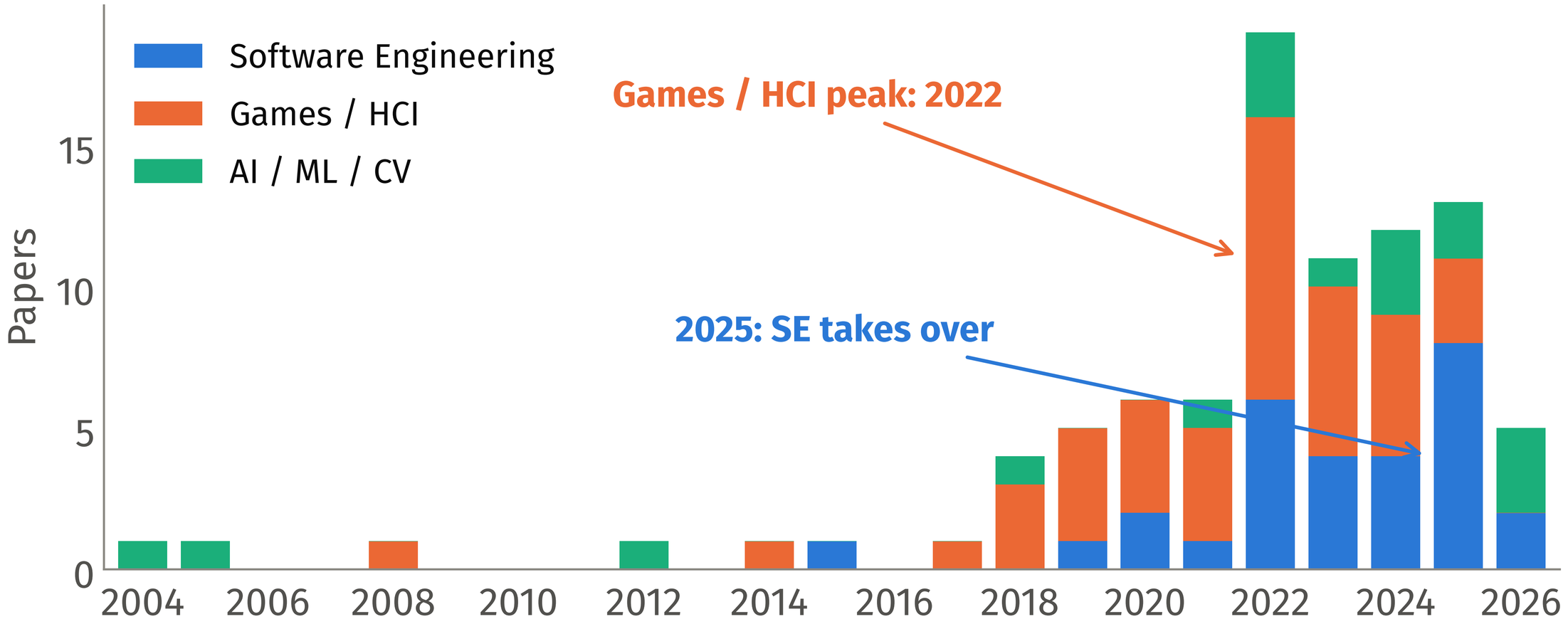


Literature Survey

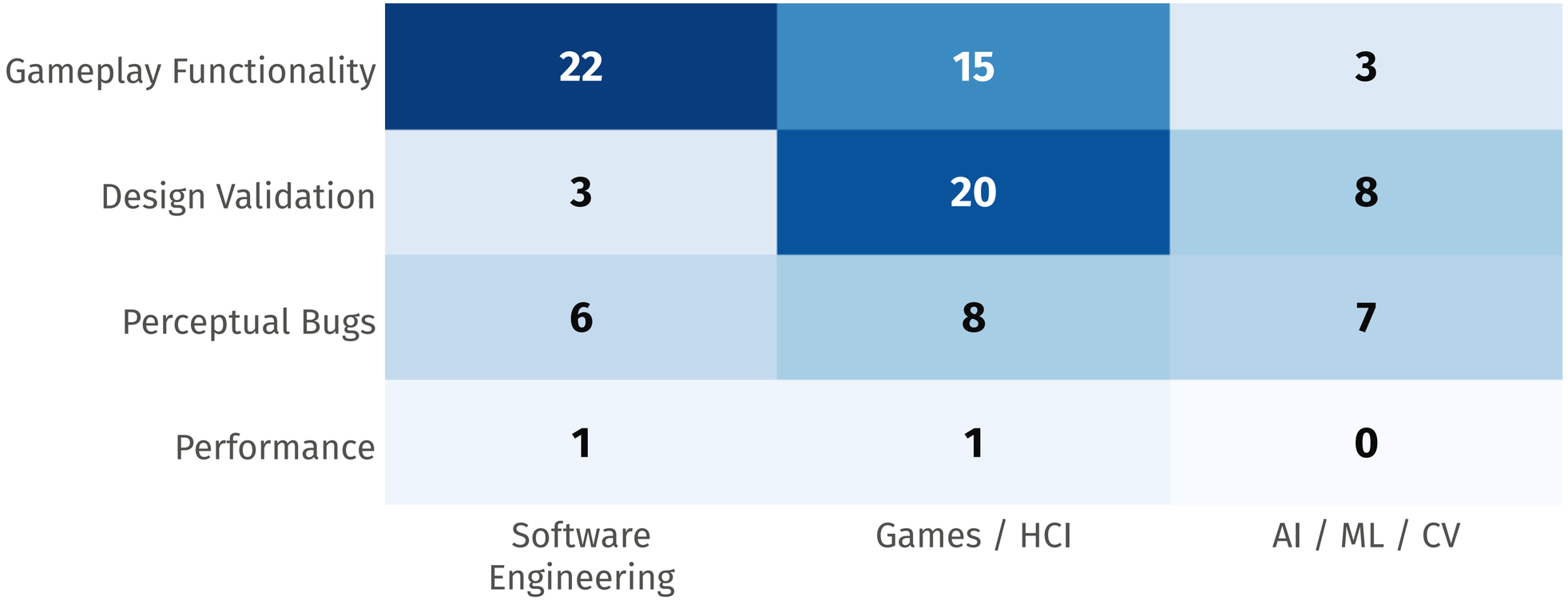
72 of 88 papers (82%) published since 2020



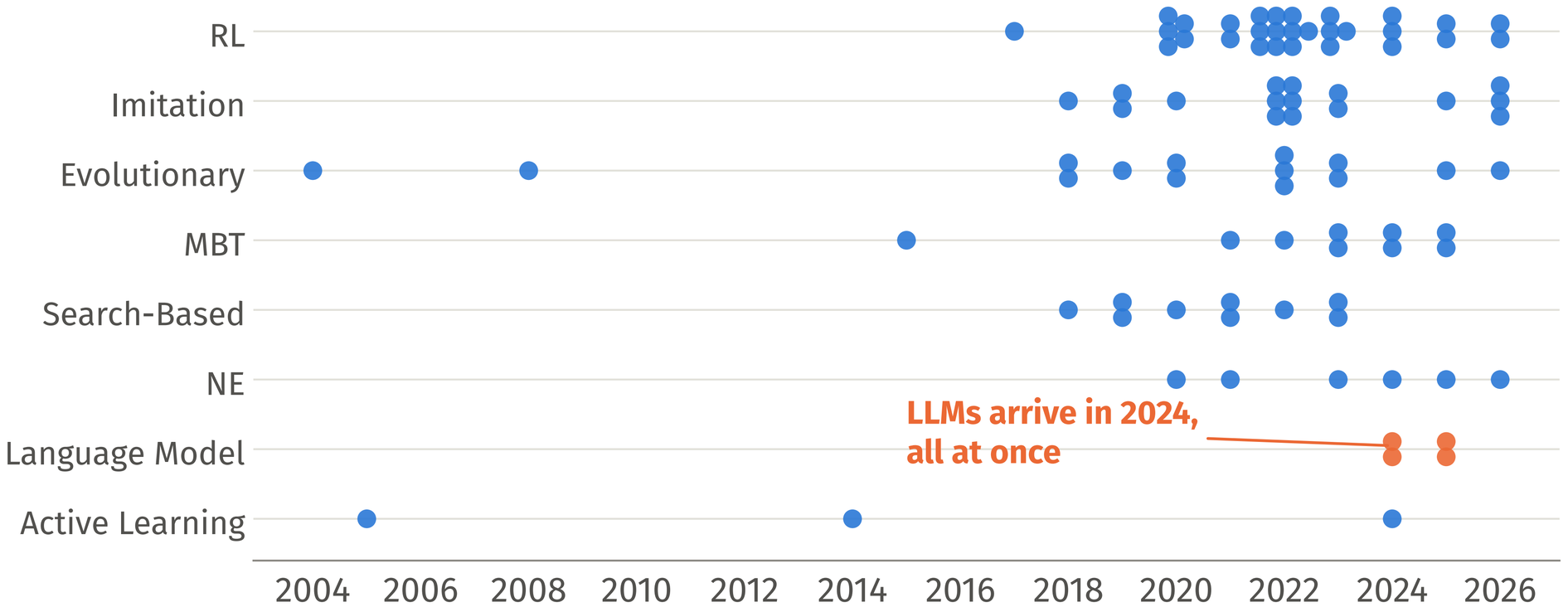
Literature Survey



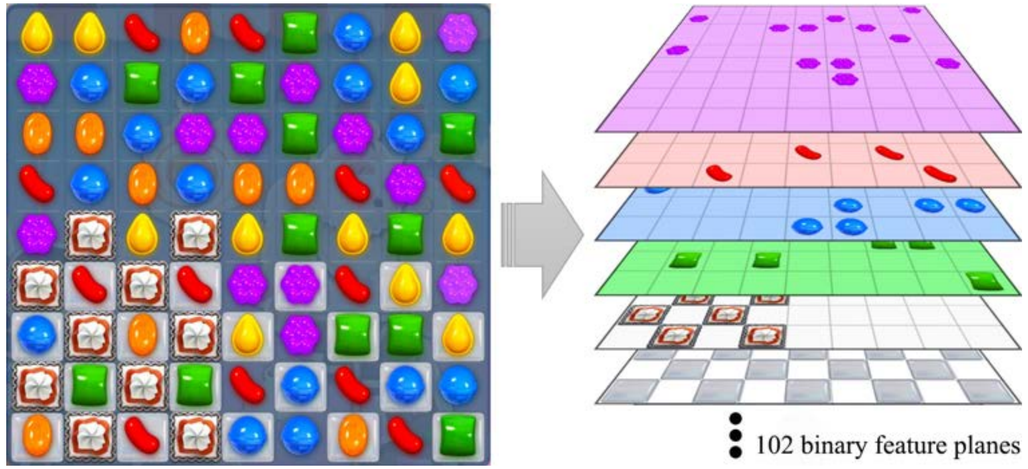
Literature Survey

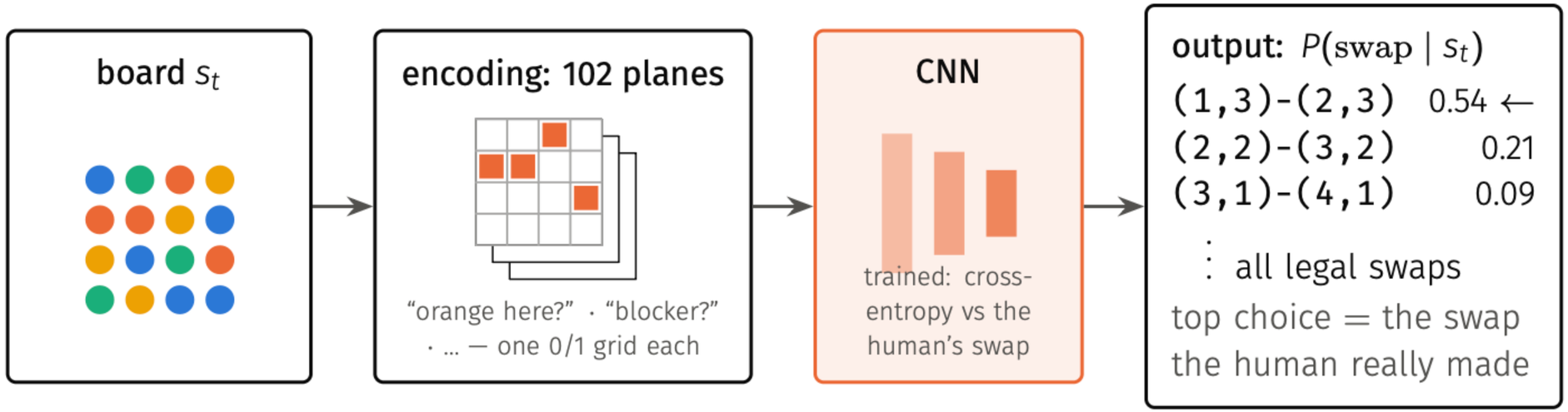


Literature Survey



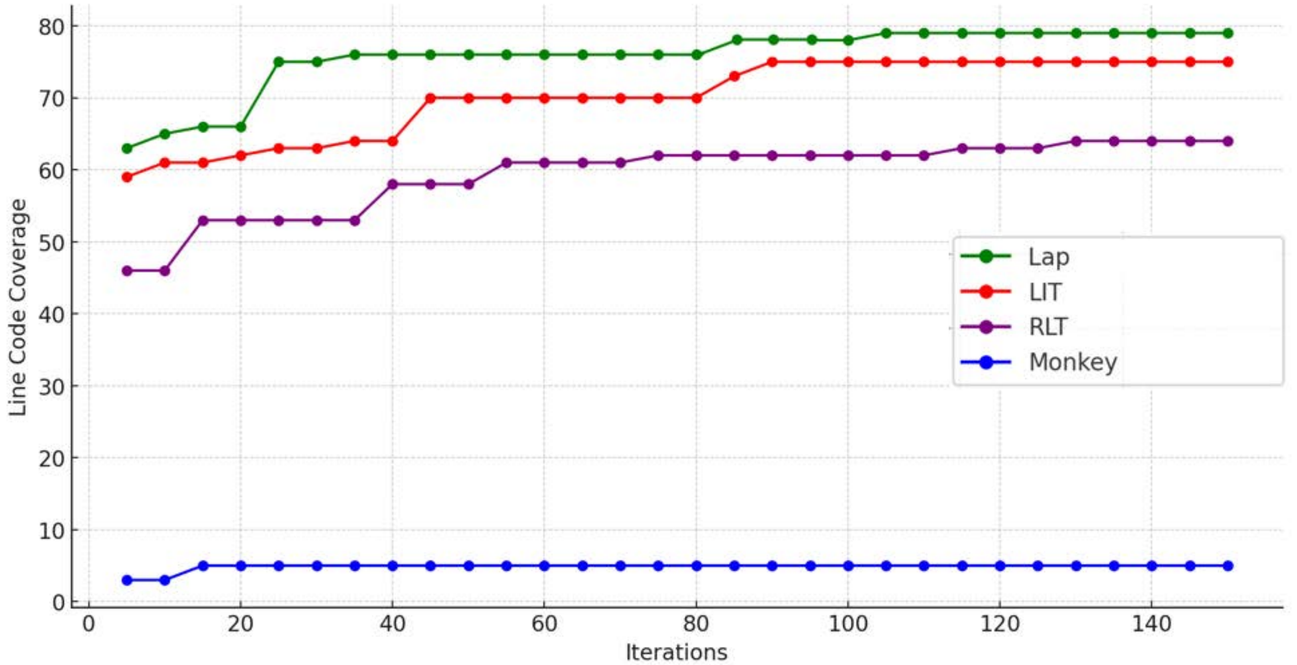
Imitation Learning





Gudmundsson et al. 2018

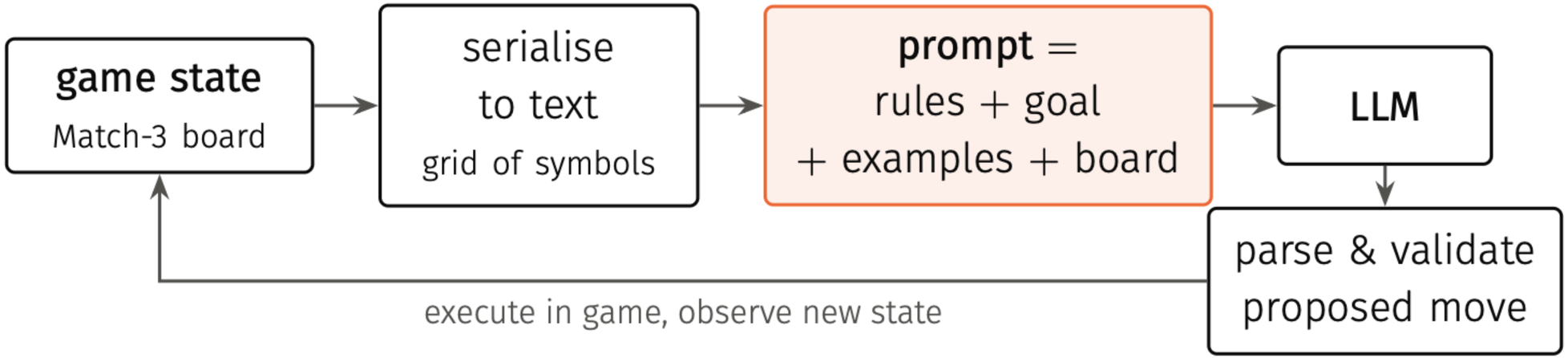
LLM-based Game Testing



- LAP: the Match-3 board is serialised to text; the LLM answers with the swap to play (blue boxes) [Yan Zhao and Tang 2025]

- Line coverage over iterations: the LLM (green) beats RL and random baselines on this game

LLM-based Game Testing



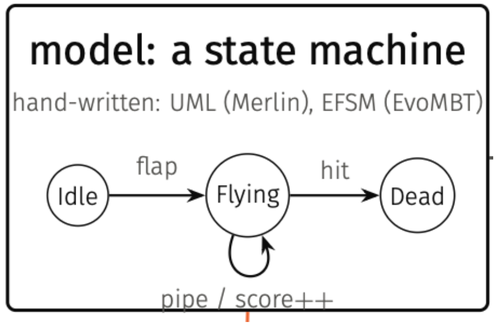
paraphrased from LAP:

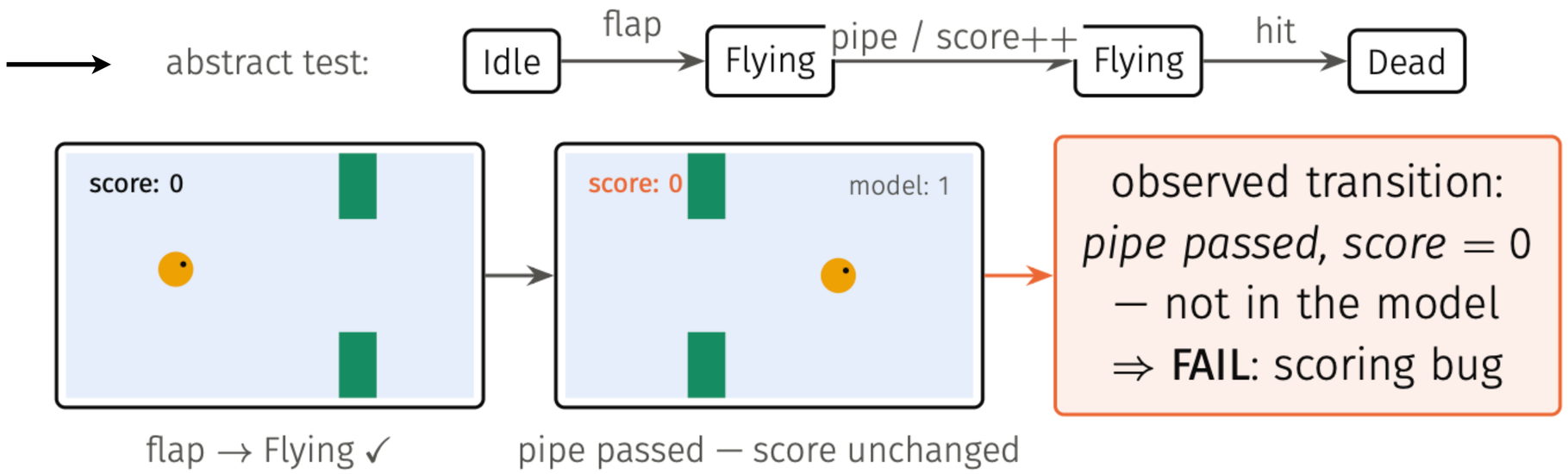
```
Rules: swap two adjacent tiles; 3 in a row clears.
Goal: clear the level. Moves left: 20.
Example:  R R G B / B G R G → swap (1,3)--(2,3)
Current board:  R G B G B  /  G R R B R  /  B G B G G
Which swap should I play?          ⇒ LLM: “swap (2,4)–(2,5)”
```

what the text encodes

swap → R R R clears

Model-based Testing

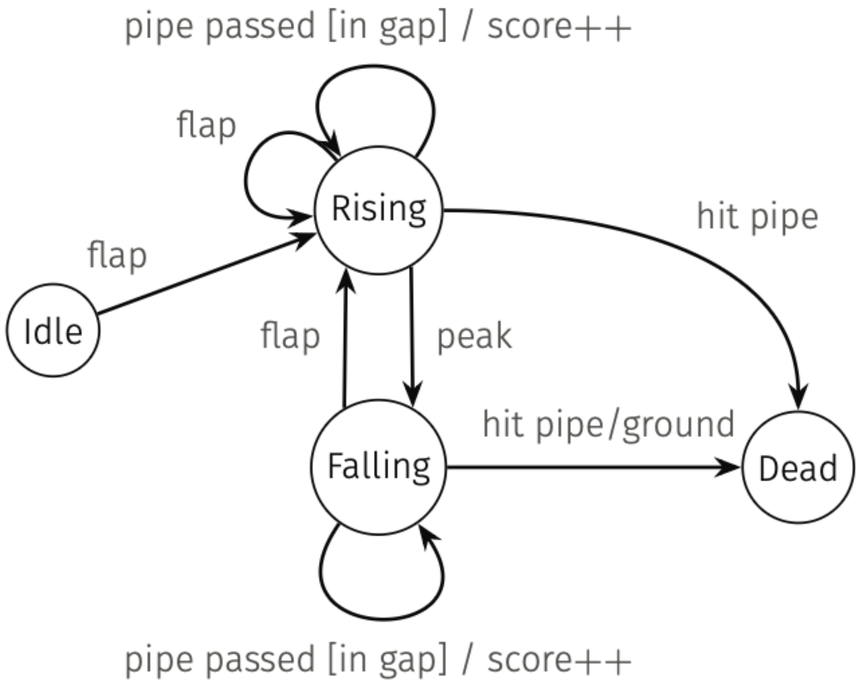




the concrete trace the glue layer runs (abstract `flap` ↦ concrete input `tap`):

```
0.0s  send: tap                  observe: Idle → Flying          ✓ flap
0.3s  wait for event ``pipe passed'' (timeout 5s)                no input — just wait
2.1s  event: pipe passed         observe: score = 0              × model: score = 1
```

Model-based Testing



flap·peak·flap·flap·peak·[pipe passed]·flap …

the executor's job: make the path's next transition fire — here `pipe passed [in gap]`

1 · scripted adapter

`flap` ↦ `tap`
`pipe passed` ↦ `poll the game state`

a hand-written dictionary, action by action

\+ instant, deterministic
− only if transitions ≈ single inputs

2 · goal solver (EvoMBT)

transition = goal:
`be in the gap when the pipe arrives`
tactics read the state and compute the flap times

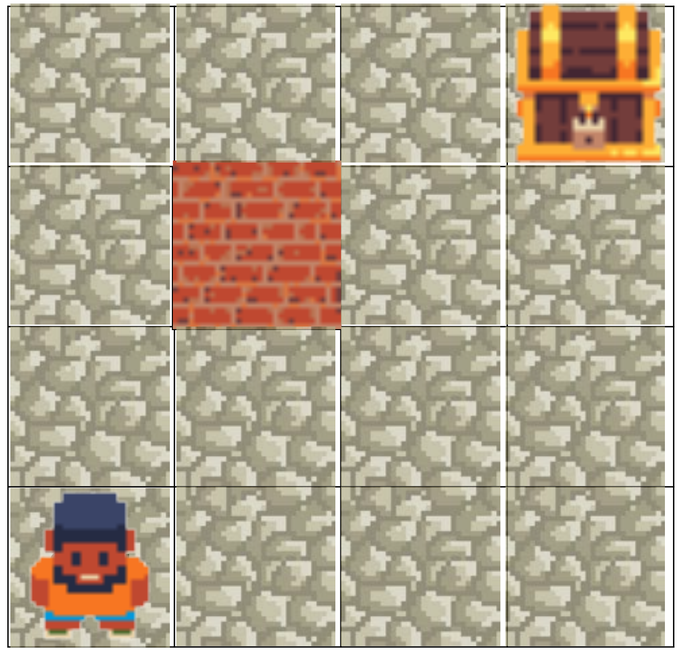
\+ no training, replannable
− needs readable game state (API)

3 · RL agent (Merlin)

`reward` = +1 `when the next transition fires`
a DQN *learns* when to flap — the model is the reward

\+ no per-game script/API
− training, and retraining as the game evolves

Reinforcement Learning



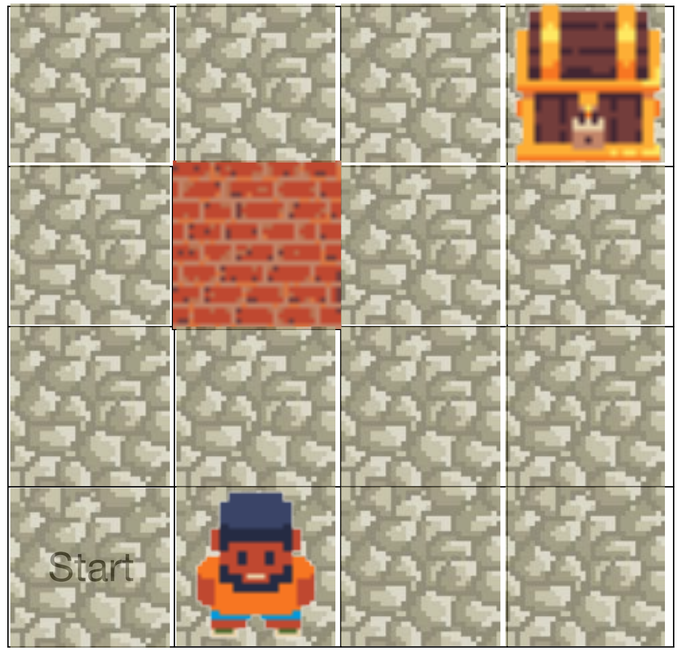
0	0	0	1
0	-1	0	0
0	0	0	0
0	0	0	0

Reward

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Q-Table

Q-Learning



0	0	0	1
0	-1	0	0
0	0	0	0
0	0	0	0

Reward

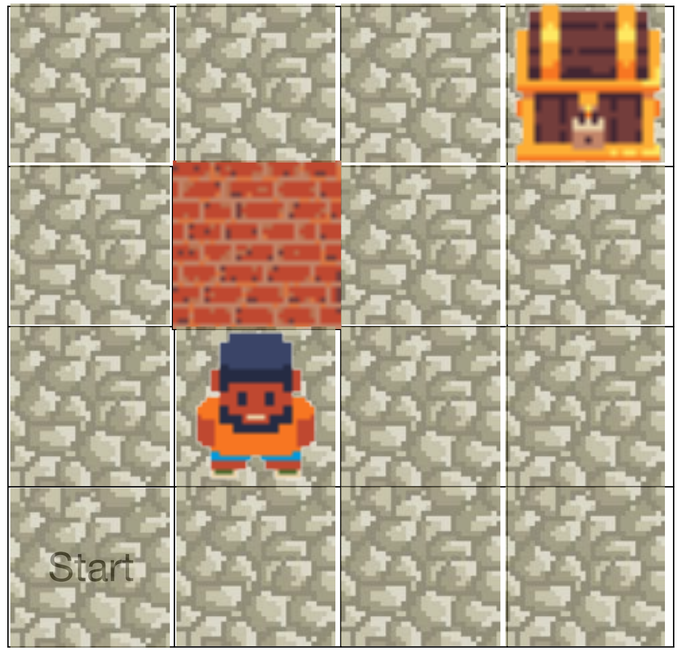
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Q-Table

State (3,0)
Action: Right
Next state (3,1)
Reward: 0

$$Q((3,0), \text{right}) = 0 + 0.1 \cdot [0 + 0.9 \cdot \max(Q(3,1)) - 0]$$

Q-Learning



State (3,1)
Action: Up
Next state (2,1)
Reward: 0

0	0	0	1
0	-1	0	0
0	0	0	0
0	0	0	0

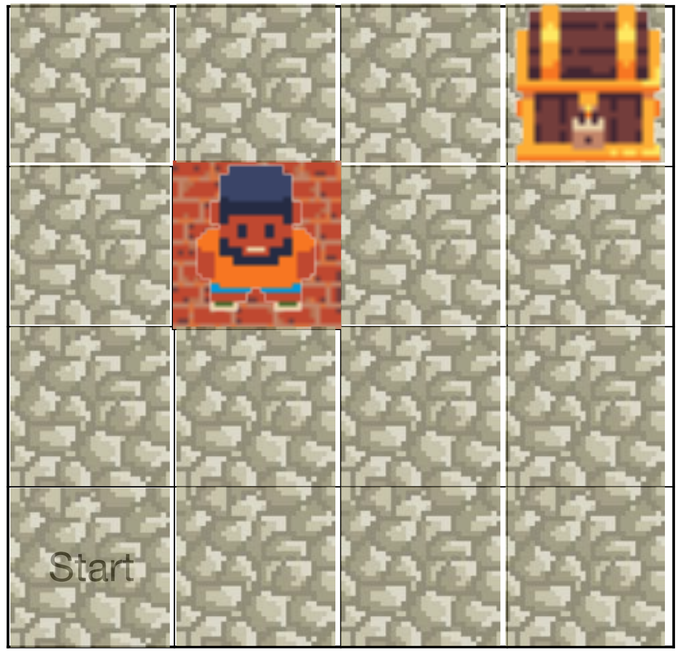
Reward

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Q-Table

$$Q((3,1), up) = 0 + 0.1 \cdot [0 + 0.9 \cdot \max(Q(2,1)) - 0]$$

Q-Learning



State (2,1)
Action: Right
Next state (1,1)
Reward: -1

0	0	0	1
0	-1	0	0
0	0	0	0
0	0	0	0

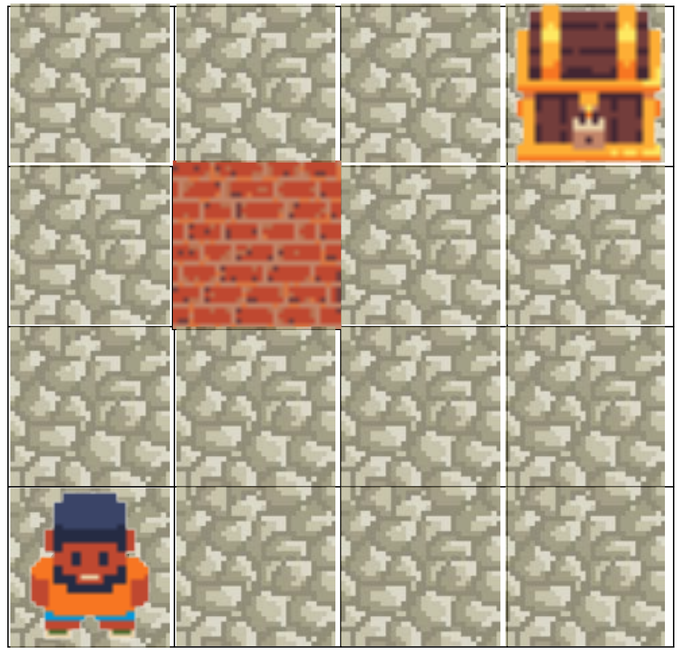
Reward

0	0	0	0
0	0	0	0
0	-0.1	0	0
0	0	0	0

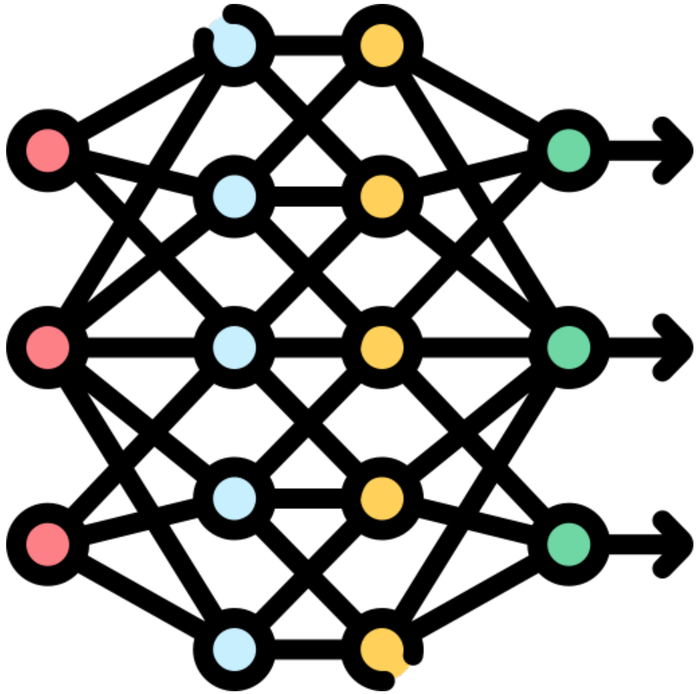
Q-Table

$Q((2,1), up) = 0 + 0.1 \cdot [-1 + 0] = -0.1$

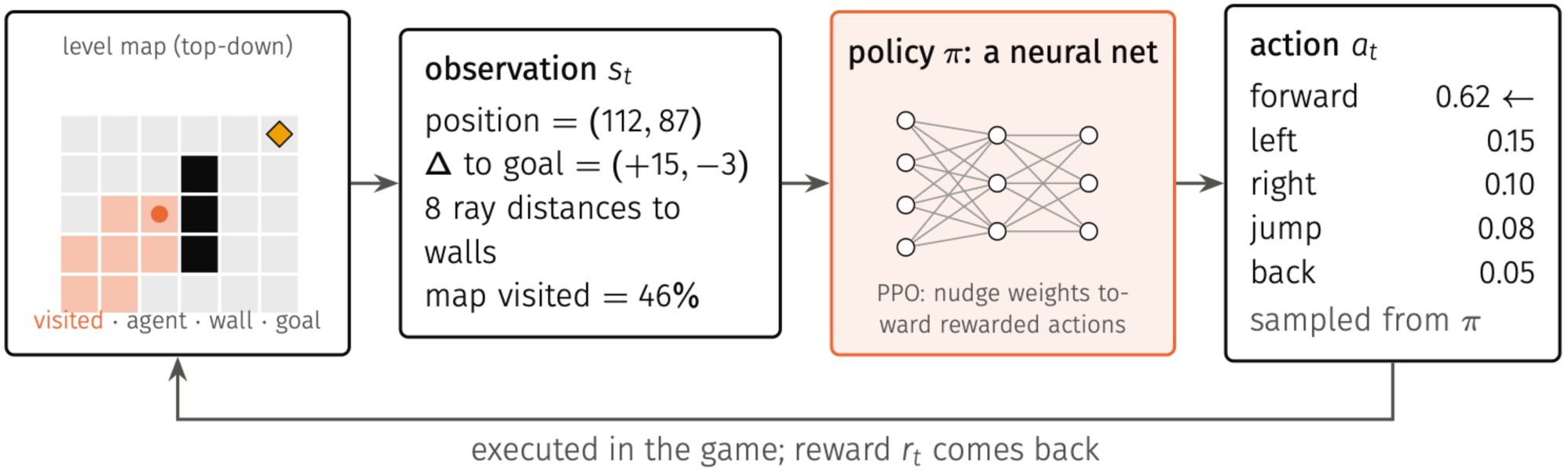
Deep Q-Learning



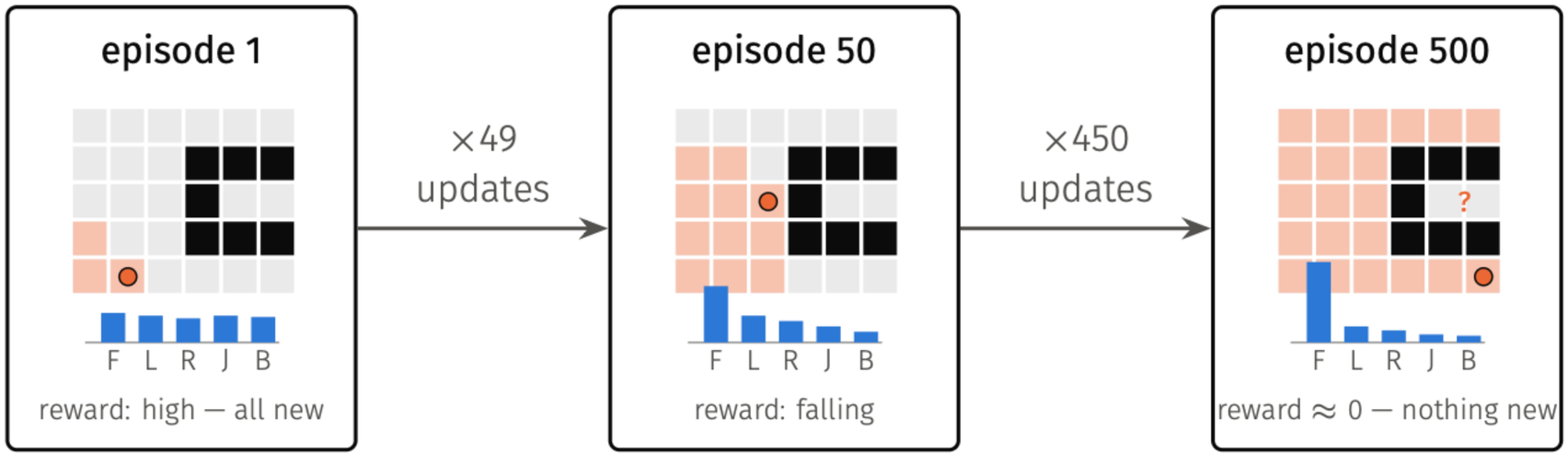
0	0	0	1
0	-1	0	0
0	0	0	0
0	0	0	0



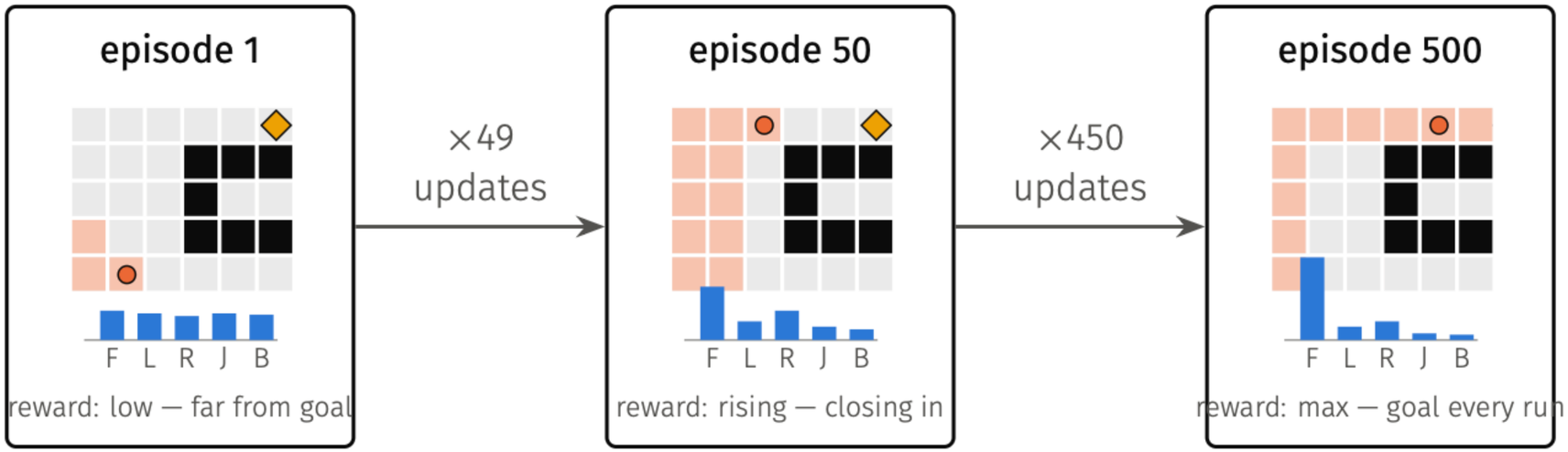
Reinforcement Learning



Reinforcement Learning



Reinforcement Learning



Reinforcement Learning

Reward function

Hand-crafted

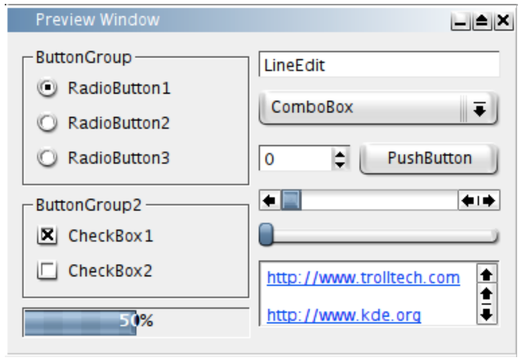
Usually winning state

Sparse

Single objective

Focus on exploitation

GUI-Testing vs. Game Testing



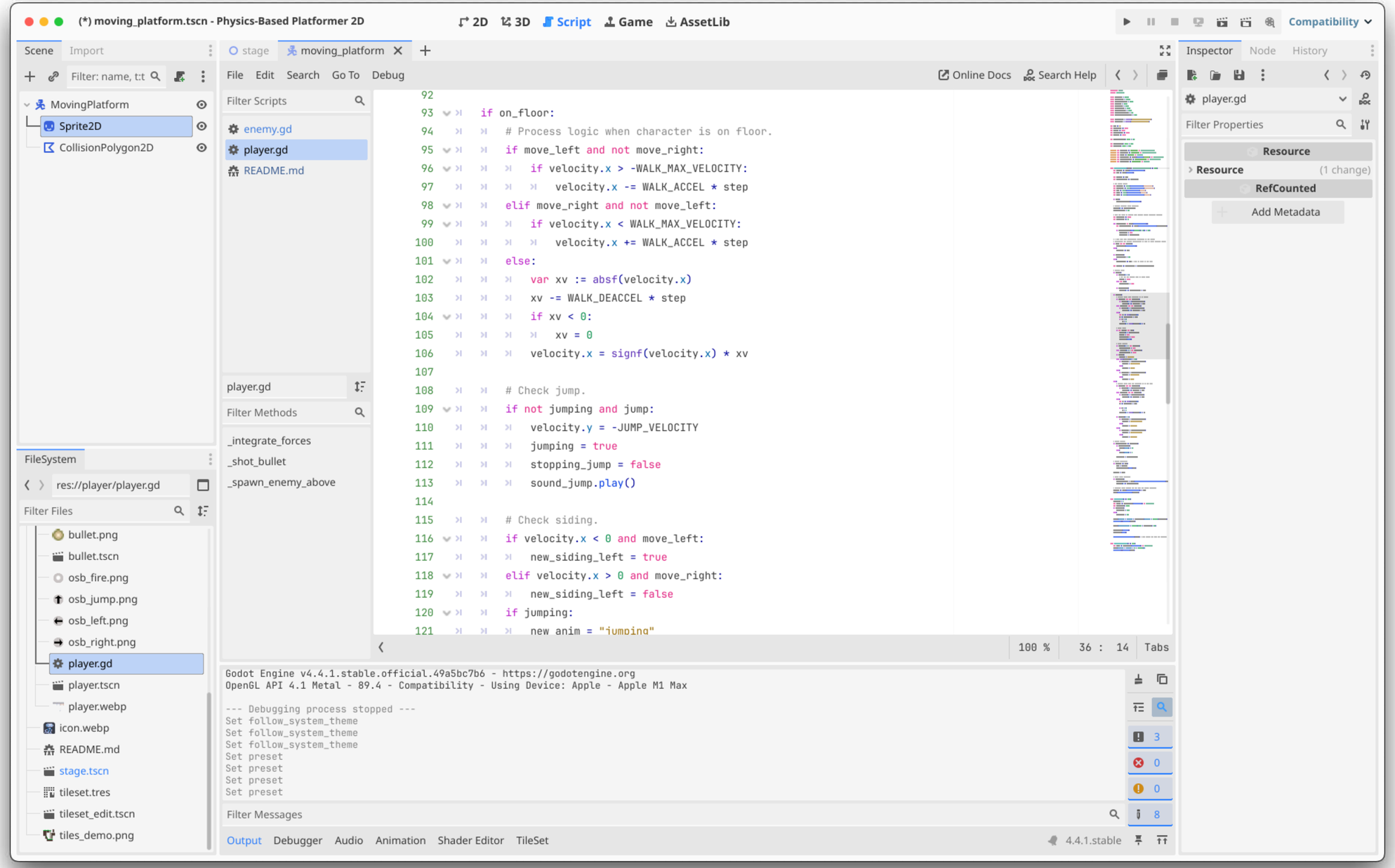
GUI

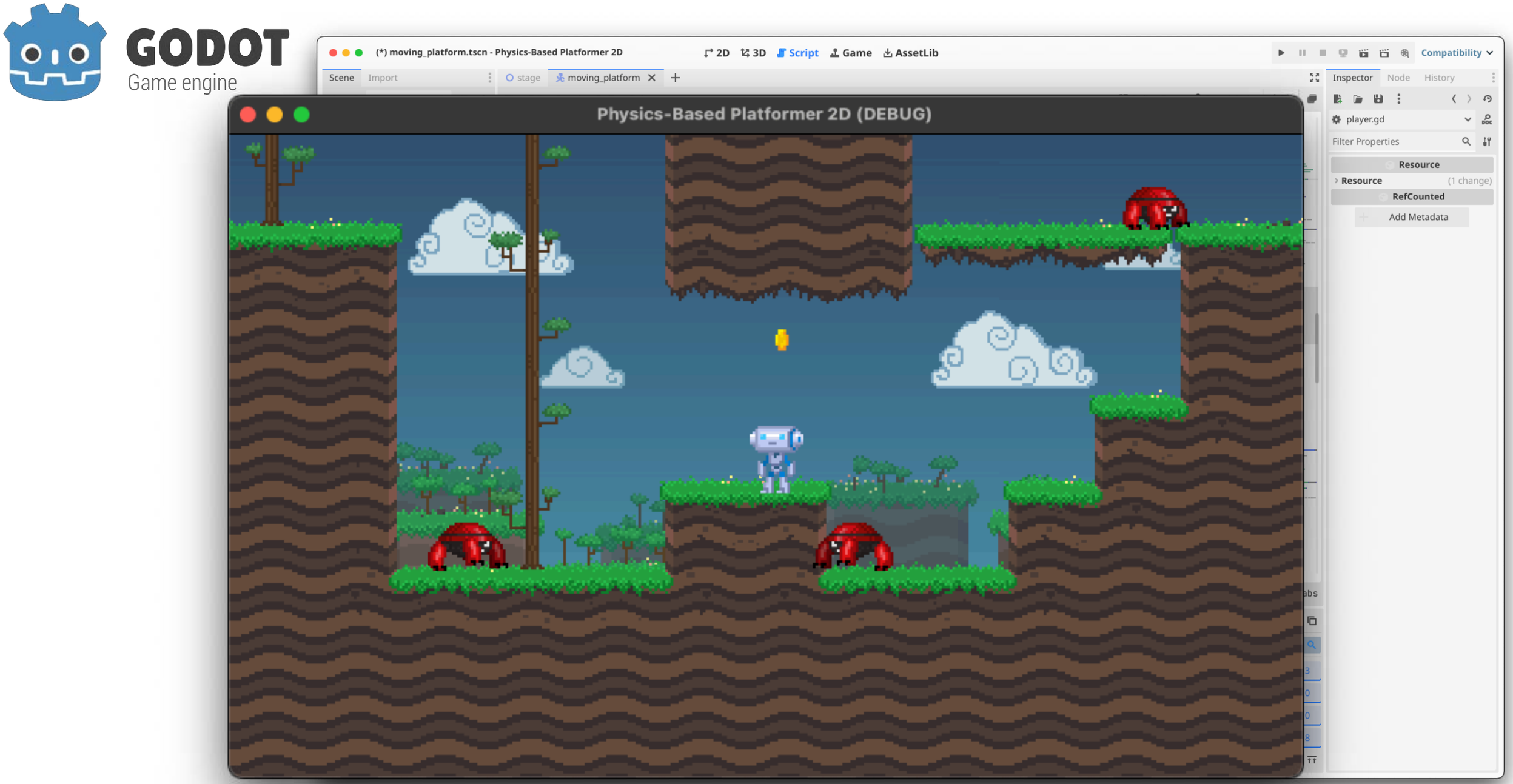
Game

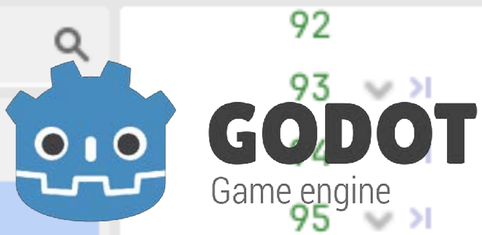
```
// display initial GUI screen
while (true)
{
  // poll for user input events
  // update internal variables based on input
  // update user interface
}
```

```
// display initial GUI screen
while (true)
{
  // process user input if any exists
  // update internal game simulation by one step
  // render current game screen
}
```

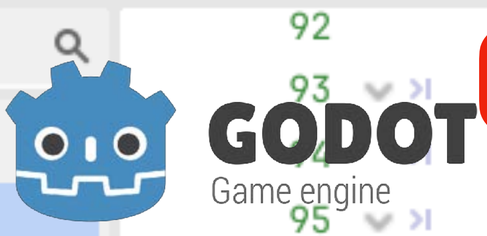
GODOT
Game engine




GODOT
Game engine
(*) moving_platform.tscn - Physics-Based Platformer 2D
2D
3D
Script
Game
AssetLib
Compatibility
Scene
Import
stage
moving_platform
Inspector
Node
History
player.gd
Filter Properties
Resource
Resource
(1 change)
RefCounted
Add Metadata
Physics-Based Platformer 2D (DEBUG)



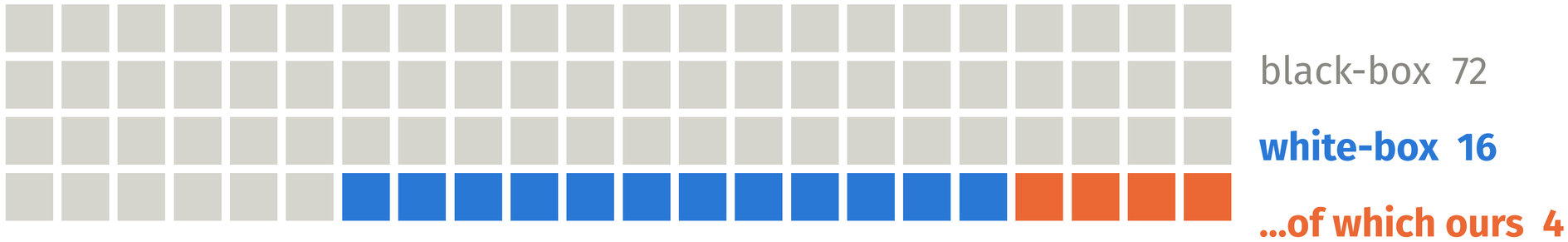

```
92
93      if on_floor:
94          # Process logic when character is on floor.
95          if move_left and not move_right:
96              if velocity.x > -WALK_MAX_VELOCITY:
97                  velocity.x -= WALK_ACCEL * step
98          elif move_right and not move_left:
99              if velocity.x < WALK_MAX_VELOCITY:
100                 velocity.x += WALK_ACCEL * step
101         else:
102             var xv := absf(velocity.x)
103             xv -= WALK_DEACCEL * step
104             if xv < 0:
105                 xv = 0
106             velocity.x = signf(velocity.x) * xv
107
108         # Check jump.
109         if not jumping and jump:
110             velocity.y = -JUMP_VELOCITY
111             jumping = true
112             stopping_jump = false
113             sound_jump.play()
```

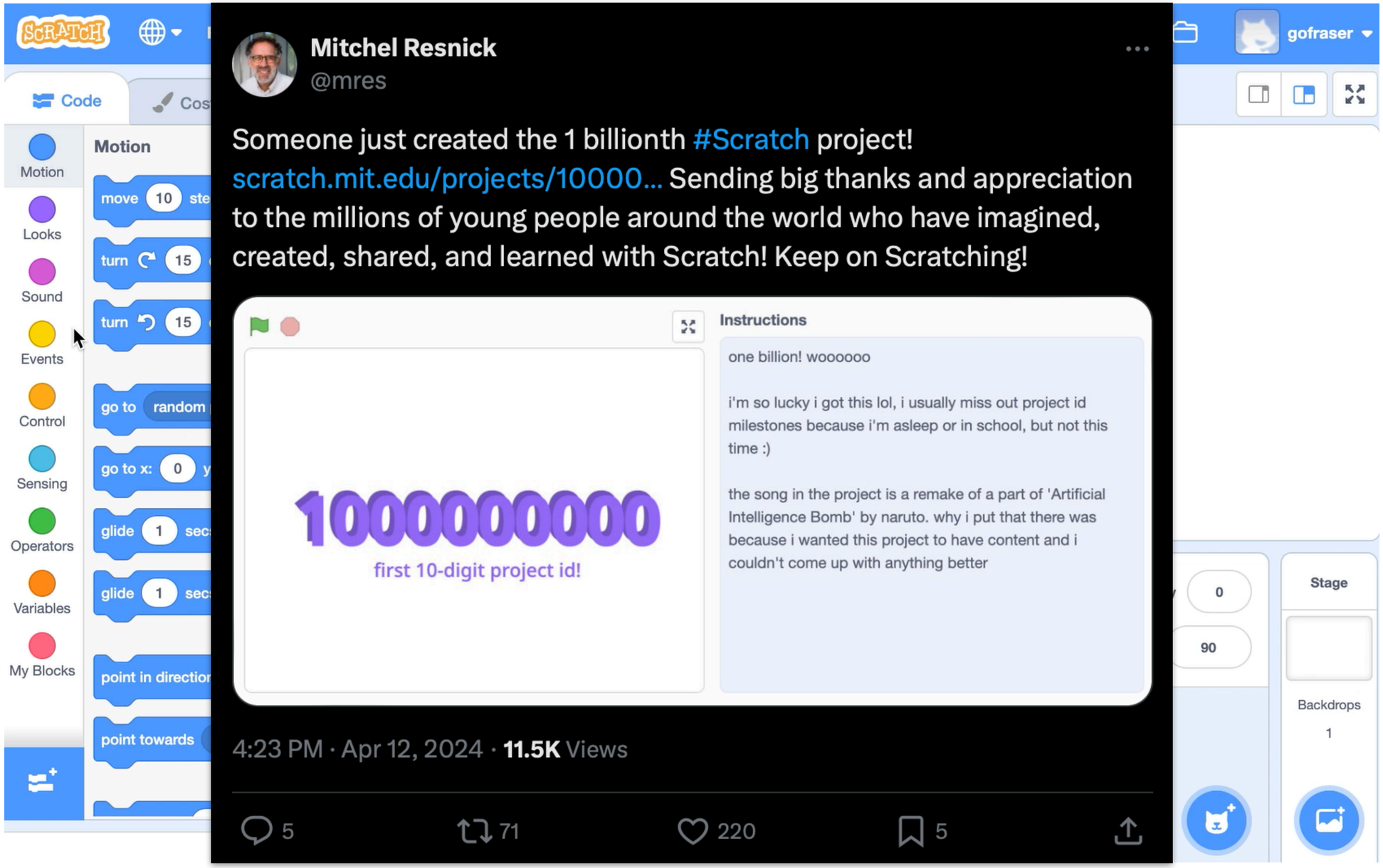


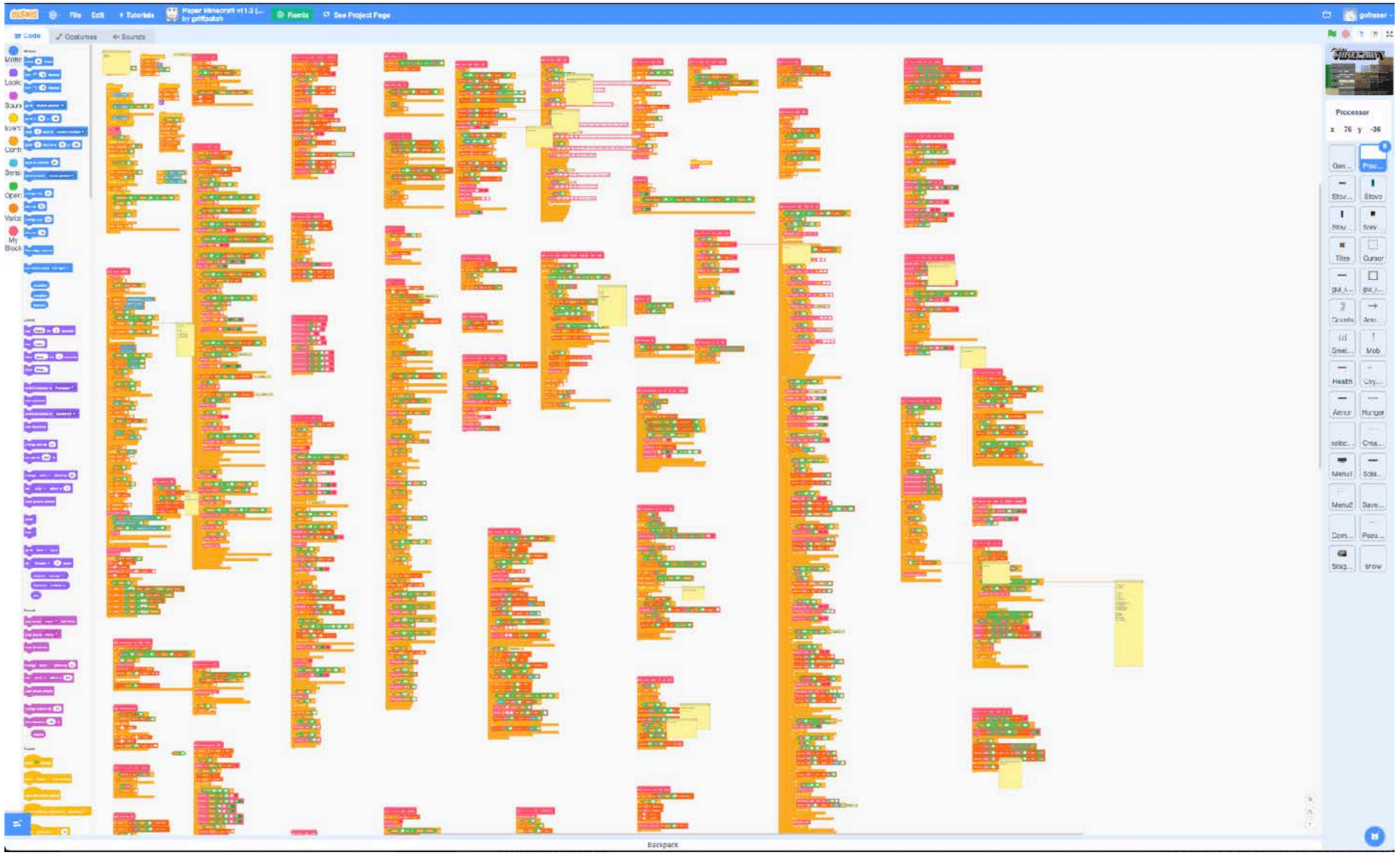

```
92
93      if on_floor:
94          # Process logic when character is on floor.
95          if move_left and not move_right:
96              if velocity.x > -WALK_MAX_VELOCITY
97                  velocity.x = WALK_ACCEL * step
98          elif move_right and not move_left:
99              if velocity.x < WALK_MAX_VELOCITY:
100                 velocity.x += WALK_ACCEL * step
101         else:
102             var xv := absf(velocity.x)
103             xv -= WALK_DEACCEL * step
104             if xv < 0:
105                 xv = 0
106             velocity.x = signf(velocity.x) * xv
107
108         # Check jump.
109         if not jumping and jump:
110             velocity.y = -JUMP_VELOCITY
111             jumping = true
112             stopping_jump = false
113             sound_jump.play()
```

Cover these?

Literature Survey




SCRATCH
gofraser
Code
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
move 10
turn 15
turn 15
go to random
go to x: 0
glide 1
glide 1
point in direction
point towards
Stage
Backdrops
1
0
90
Mitchel Resnick
@mres
Someone just created the 1 billionth #Scratch project! scratch.mit.edu/projects/10000... Sending big thanks and appreciation to the millions of young people around the world who have imagined, created, shared, and learned with Scratch! Keep on Scratching!
Instructions
one billion! woooooo
i'm so lucky i got this lol, i usually miss out project id milestones because i'm asleep or in school, but not this time :)
the song in the project is a remake of a part of 'Artificial Intelligence Bomb' by naruto. why i put that there was because i wanted this project to have content and i couldn't come up with anything better
1000000000
first 10-digit project id!
4:23 PM · Apr 12, 2024 · 11.5K Views
5
71
220
5


File
Edit
Tutorials
Paper Minecraft v11.3 |...
by griffpatch
Remix
See Project Page
gofraser
Code
Costumes
Sounds
Processor
x 76 y -36
Backpack

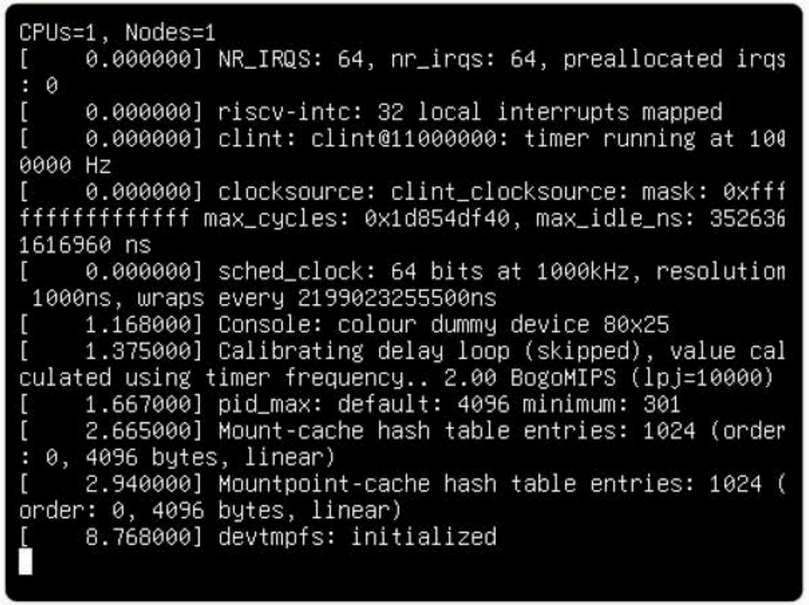
scratch.mit.edu

Create Explore Ideas About Search gofraser

Linux 6.1.14-rv32ima On Scratch

by bilman66

Remix See inside



Instructions

A real build of the Linux 6.1.14 kernel running in pure scratch code!
This is done by running the linux kernal on a RISC-V (rv32ima) emulator I wrote in Scratch, based off of Cnlohr's mini-rv32ima emulator.

Use turbowarp for results that are actually usable

Notes and Credits

Huge thanks to @Tacodiva7729 for making the terminal better as well as cleaning up/optimizing some of the code! And of course, for creating the version of TurboWarp that makes this thing run so fast!

Thanks to cnlohr on GitHub for the original version of the mini-rv32ima emulator written in C

306 281 0 2675 Sep 13, 2023 Report Add to Studio Copy Link

Pen

Tutorial

Home Shrink Close

Penalty shootout easy

Boat Race medium

Flappy Parrot medium

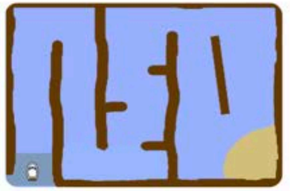
These and further tutorials you can find in the Raspberry Pi Code Club

Boat Race

Home Shrink Close

Step 1: starting position

First we have to add our boat and our parkour and then bring the boat to the starting position.



Delete the existing sprites and then click on "Upload sprite" at the bottom right. Now select the previously downloaded Boat.png. Name your

These and further tutorials you can find in the Raspberry Pi Code Club

Boat Race

Home Shrink Close

Delete the existing sprites and then click on "Upload sprite" at the bottom right. Now select the previously downloaded Boat.png. Name your character "Boot" and set the size to 10. You can also upload the stage design using the "Upload Backdrop" option at the bottom right. Now add suitable blocks so that the boat is at the position (- 190, -150) and points in the direction of 0 as soon as the green flag has been clicked.

Test

These and further tutorials you can find in the Raspberry Pi Code Club

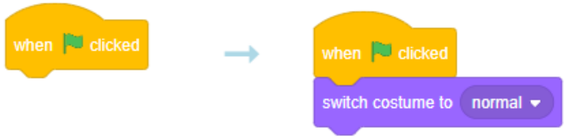
Boat Race

Home Shrink Close

Hint: Your boat is not visible. Change this on the right under the stage.

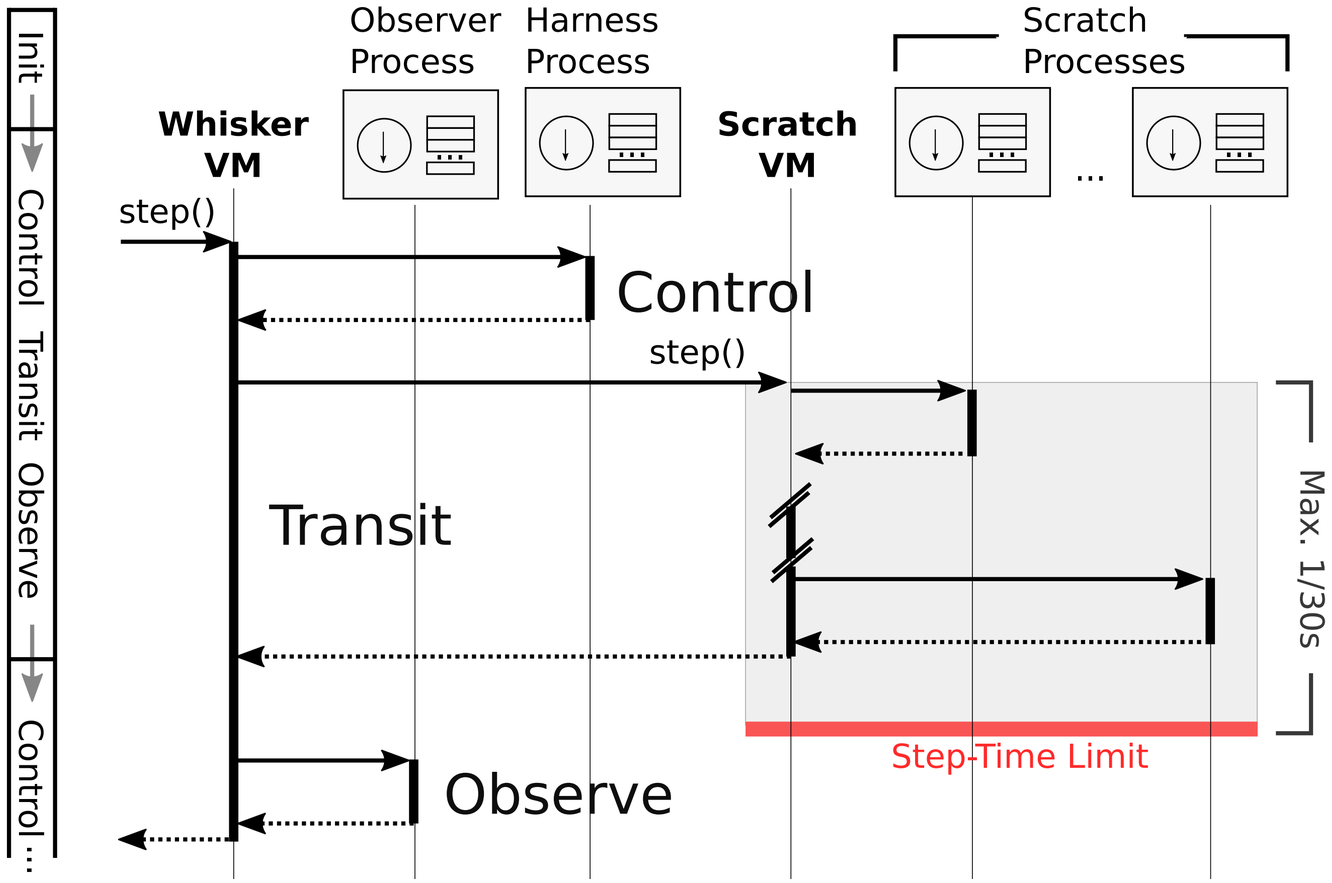
Test ✕ Your solution is not yet correct. Please try it again.

Hint

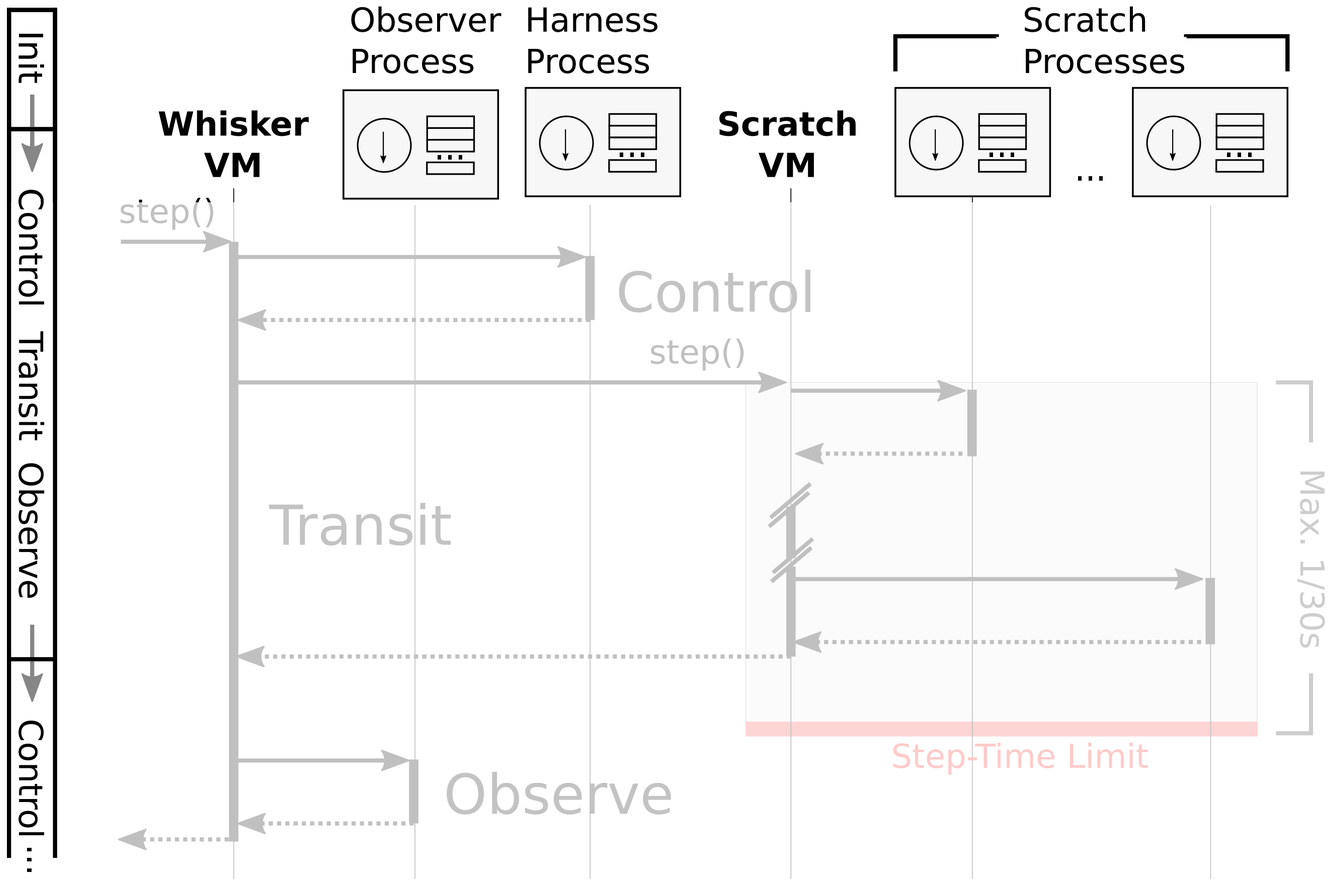


These and further tutorials you can find in the Raspberry Pi Code Club

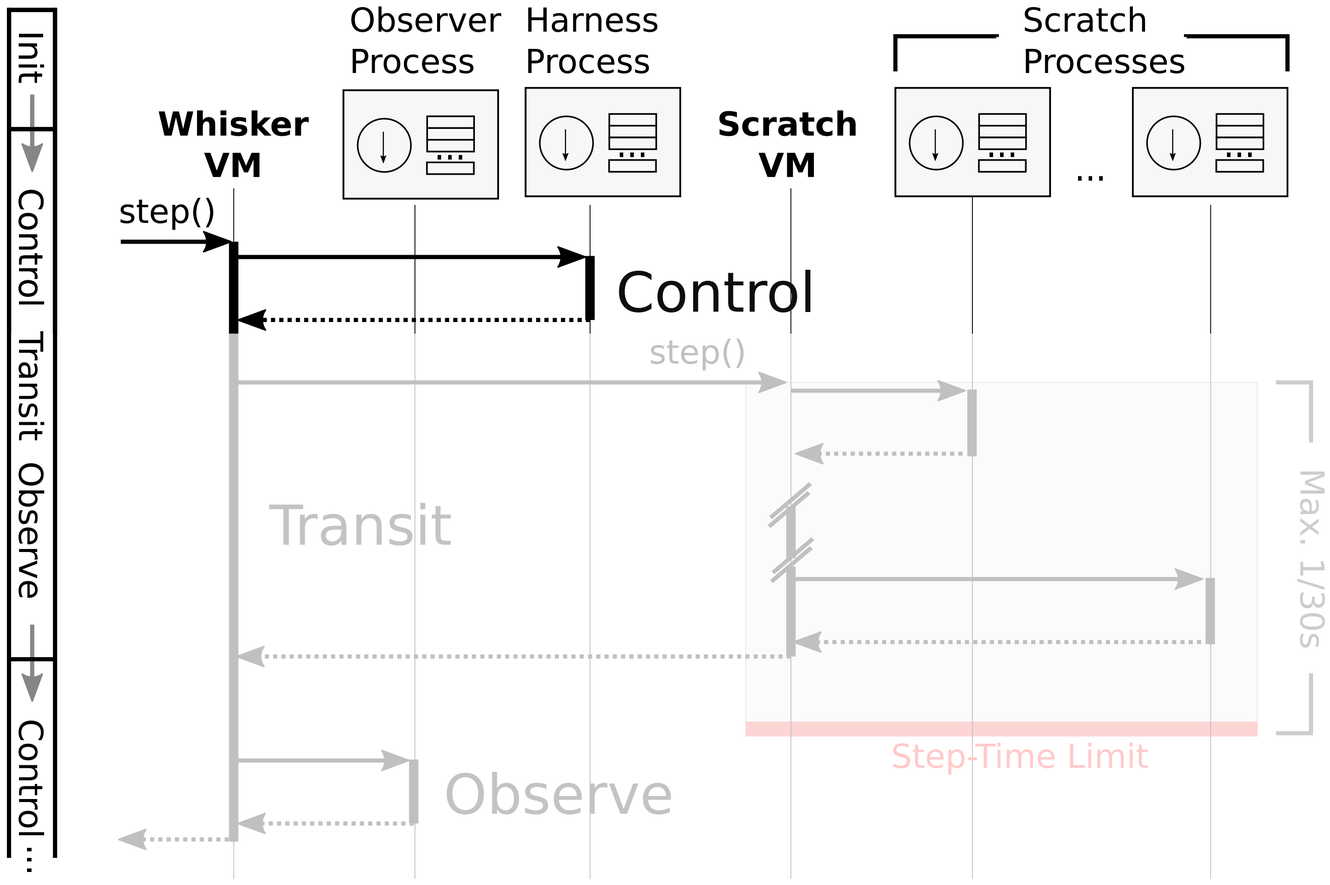
Testing Scratch: WHISKER



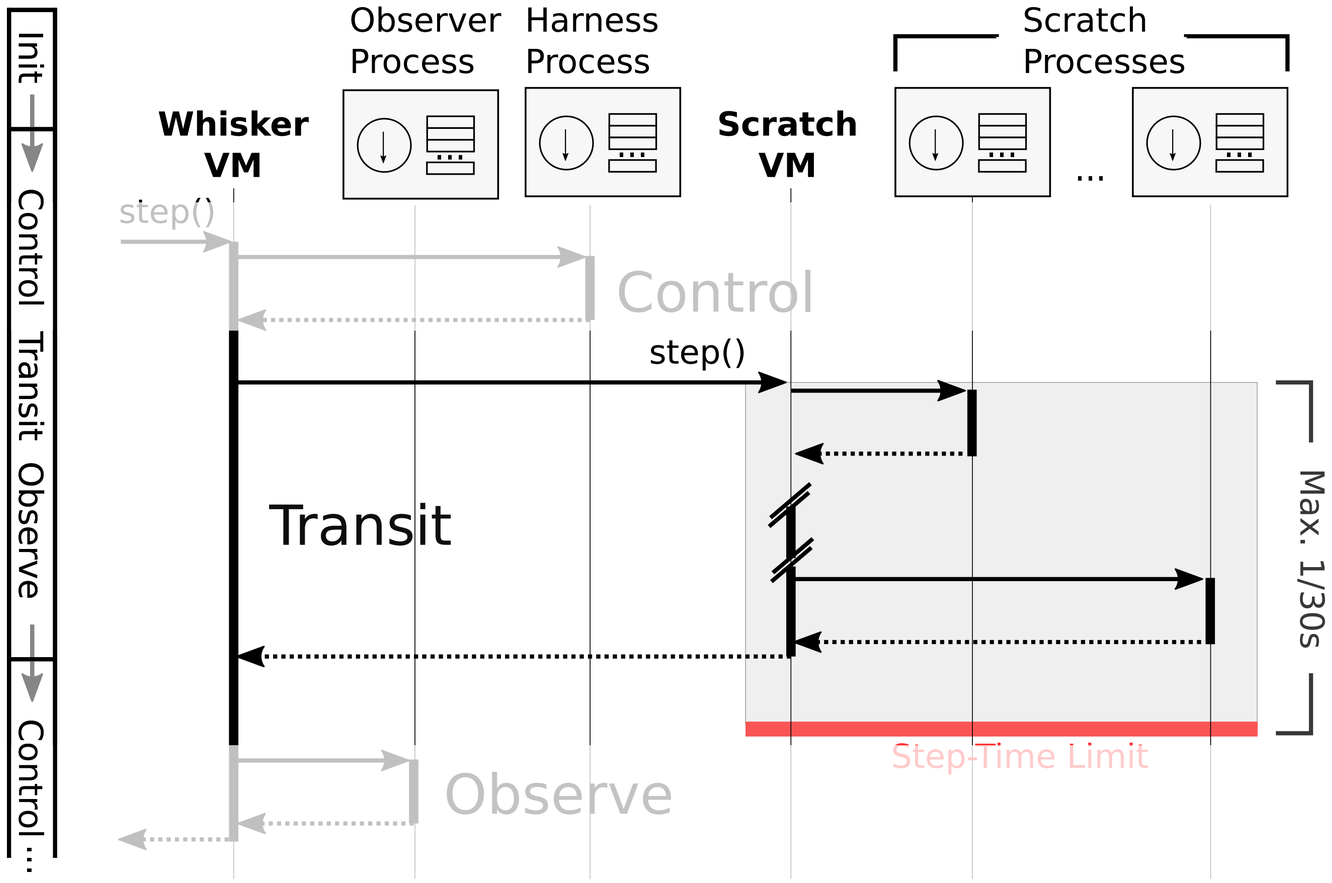
Testing Scratch: WHISKER



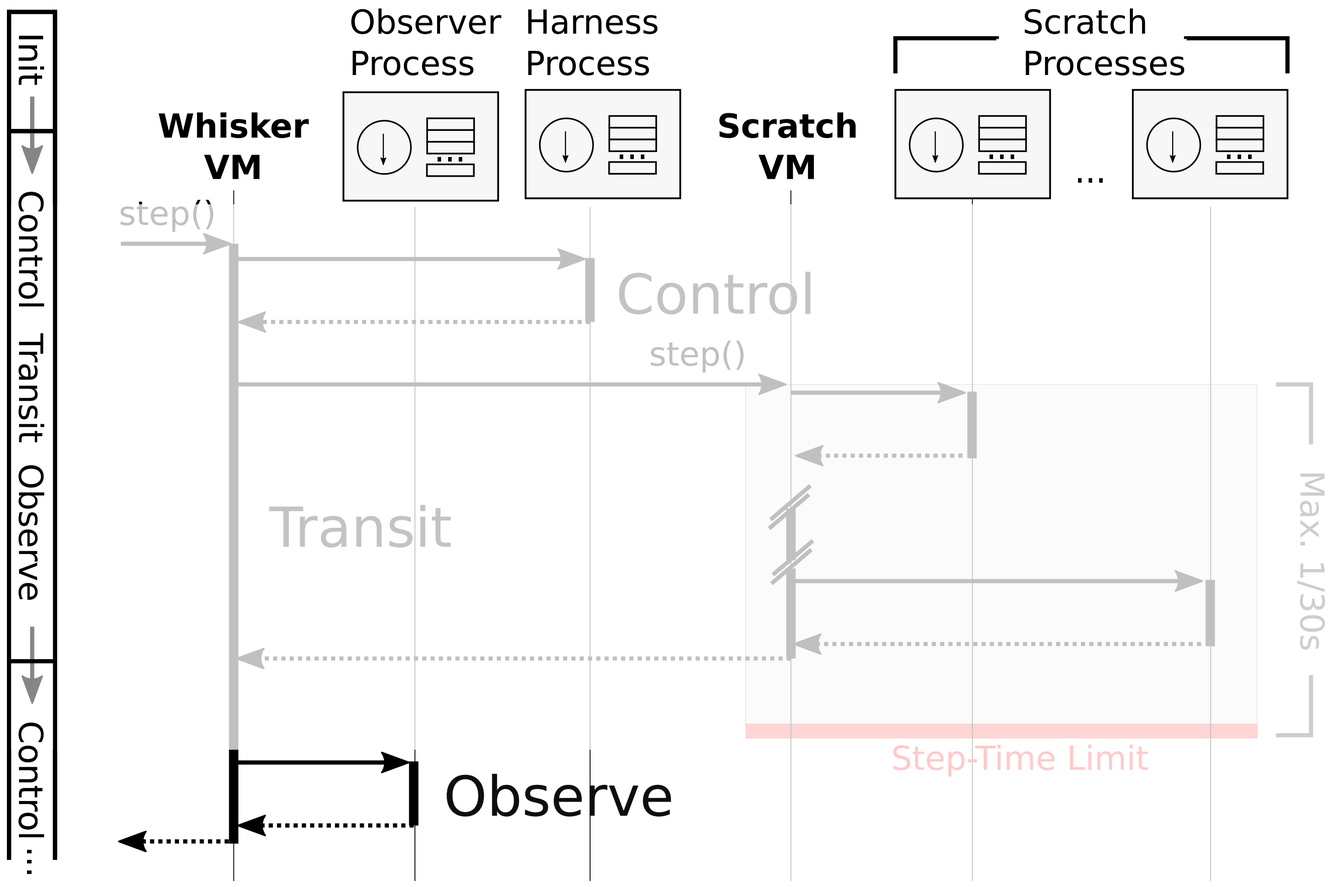
Testing Scratch: WHISKER



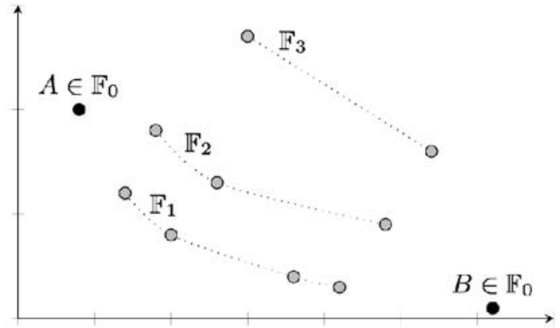
Testing Scratch: WHISKER



Testing Scratch: WHISKER



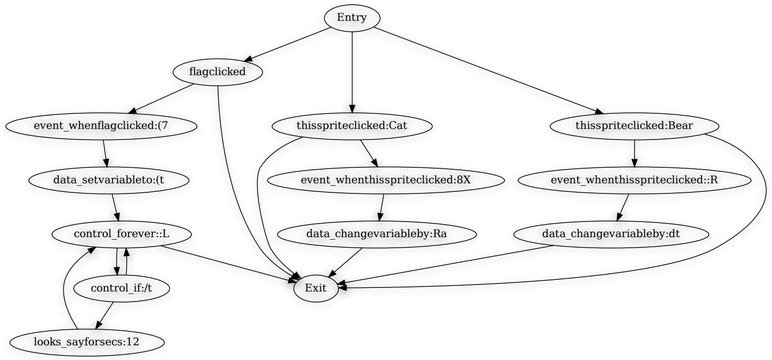
Search-based Testing with WHISKER



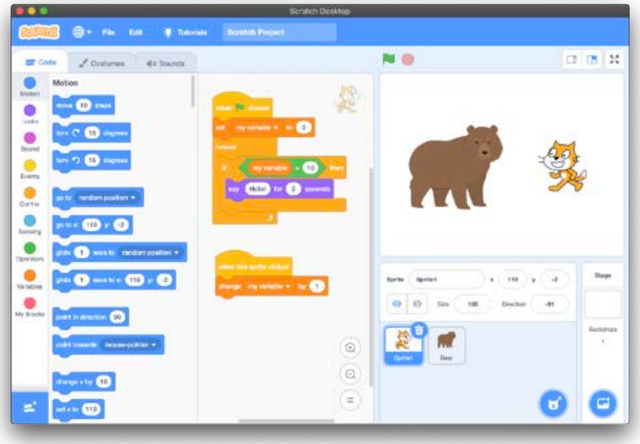
Many-Objective Search

```
<TestCase> ::= <Input> <Input> <Input>

<Input> ::= <Click> | <Keypress> | <Wait>

<Click> ::= <Sprite 1> | <Sprite 2>

…

<Wait> ::= <Wait 2s> | …
```

Grammatical Evolution



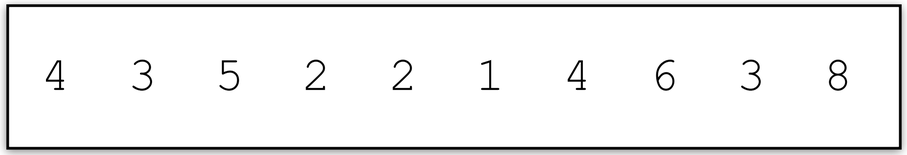
Interprocedural Approach Level + Branch Distance + CFG Distance

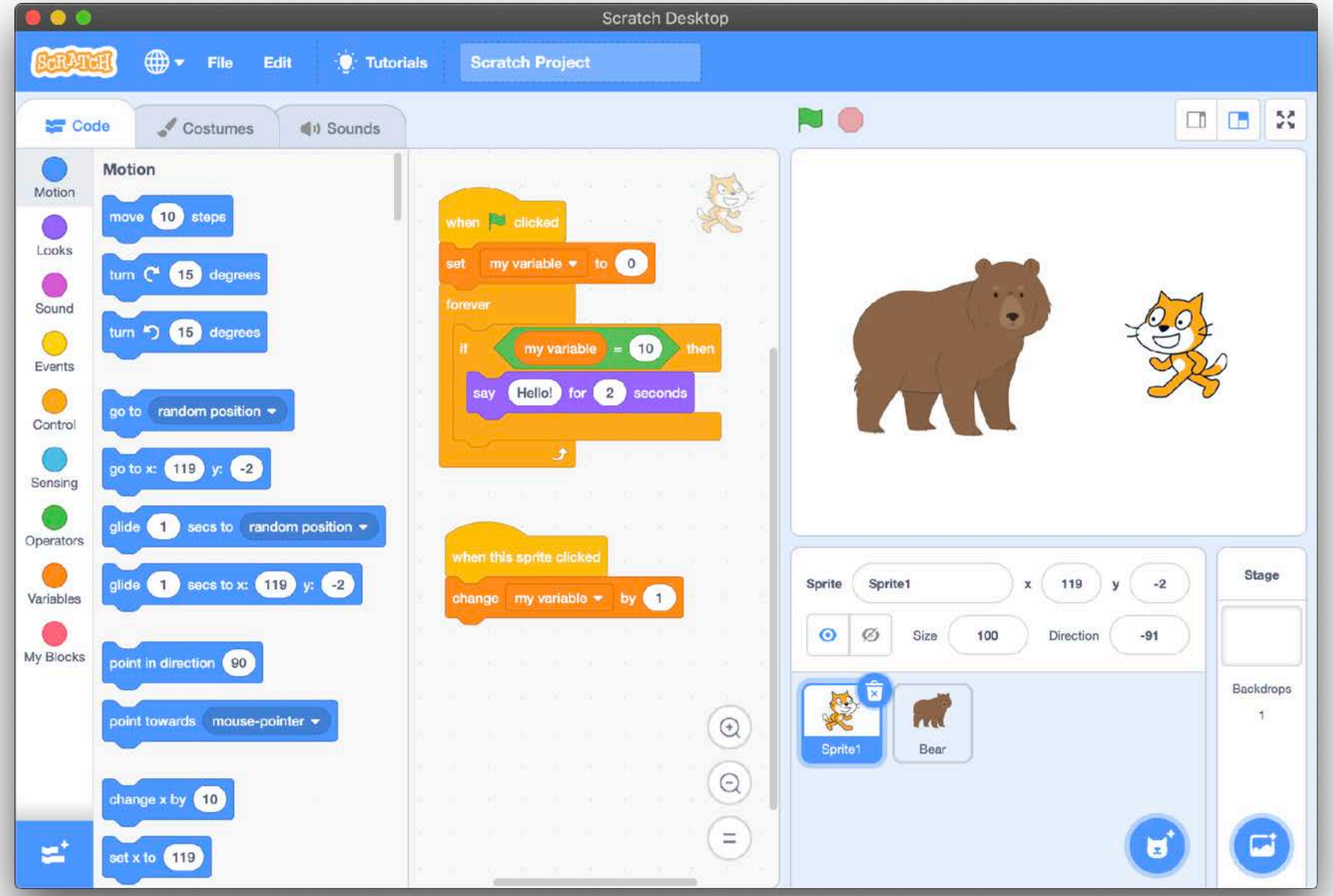


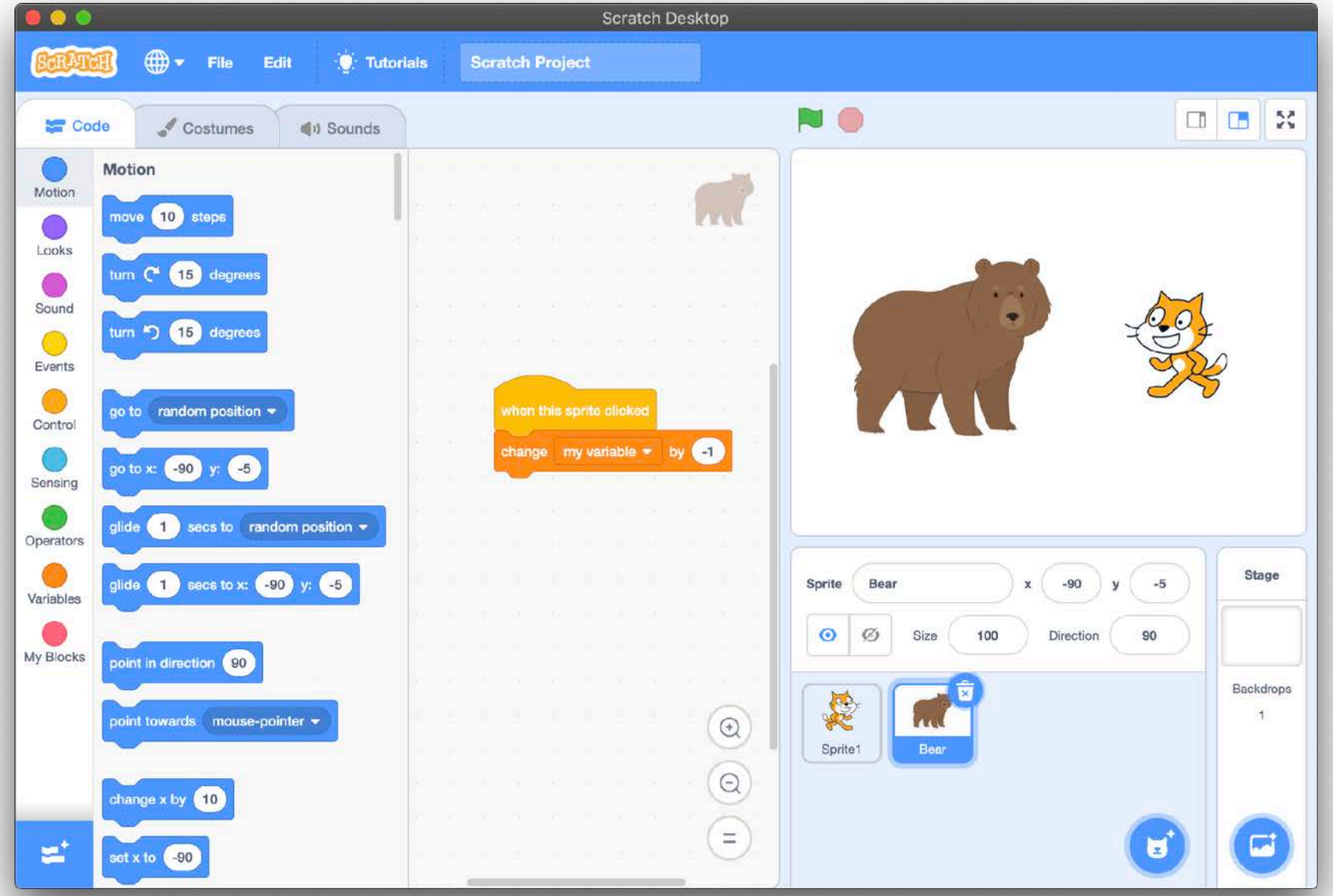
Headless, Accelerated Execution

Grammatical Evolution

```
<TestCase> ::= <Input> <Input> <Input> <Input> …
<Input> ::= <Click> | <Keypress> | <Wait> …
<Click> ::= <Click Sprite 1> | <Click Sprite 2> …
…
<Wait> ::= <Wait 2s> | …
```




Scratch Desktop
File
Edit
Tutorials
Scratch Project
Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
Motion
move 10 steps
turn 15 degrees
turn 15 degrees
go to random position
go to x: 119 y: -2
glide 1 secs to random position
glide 1 secs to x: 119 y: -2
point in direction 90
point towards mouse-pointer
change x by 10
set x to 119
when clicked
set my variable to 0
forever
if my variable = 10 then
say Hello! for 2 seconds
when this sprite clicked
change my variable by 1
Sprite
Sprite1
x
119
y
-2
Size
100
Direction
-91
Sprite1
Bear
Stage
Backdrops
1


Scratch Desktop
File
Edit
Tutorials
Scratch Project
Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
Motion
move 10 steps
turn 15 degrees
turn 15 degrees
go to random position
go to x: -90 y: -5
glide 1 secs to random position
glide 1 secs to x: -90 y: -5
point in direction 90
point towards mouse-pointer
change x by 10
set x to -90
when this sprite clicked
change my variable by -1
Sprite Bear
x -90
y -5
Size 100
Direction 90
Sprite1
Bear
Stage
Backdrops
1

Grammatical Evolution

<TestCase> ::= <Input$_1$> <Input$_2$> <Input$_3$> … <Input$_{10}$>

<Input> ::= ClickSprite bear | ClickSprite cat | Wait 2s | Wait default

4 3 5 2 2 1 4 6 3 8

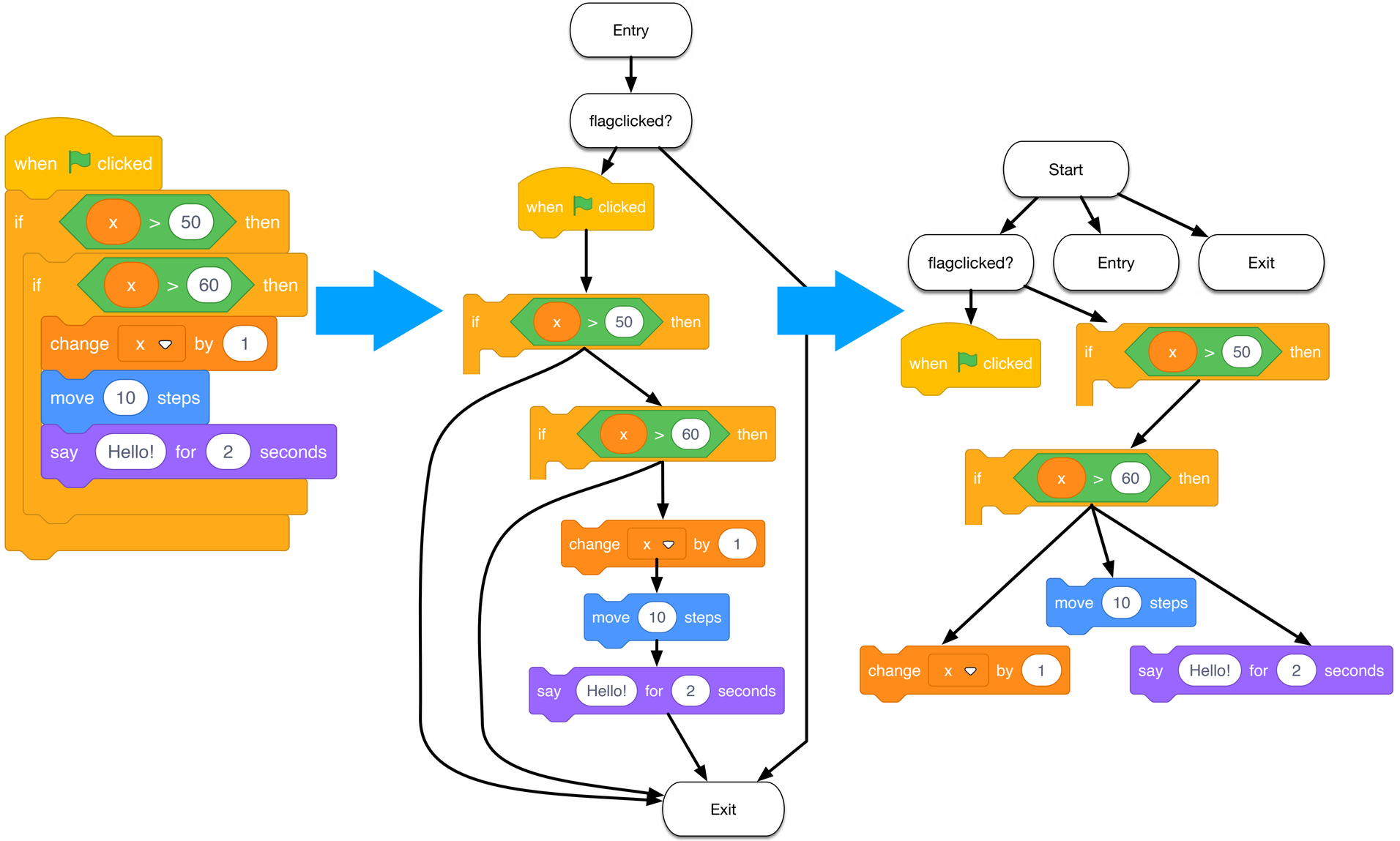
4 mod 4 = 0 → ClickSprite bear

3 mod 4 = 3 → Wait default

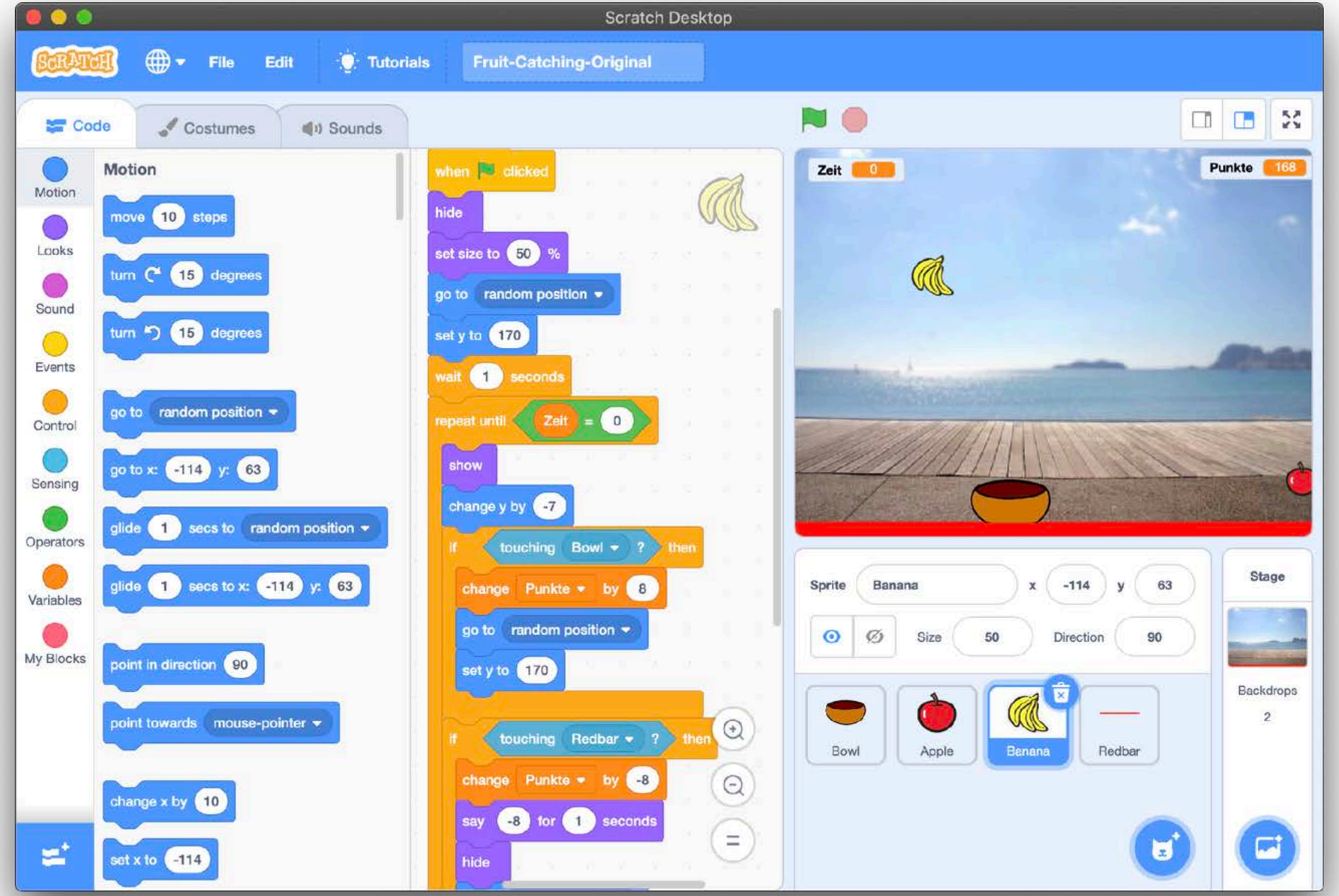
5 mod 4 = 1 → ClickSprite cat

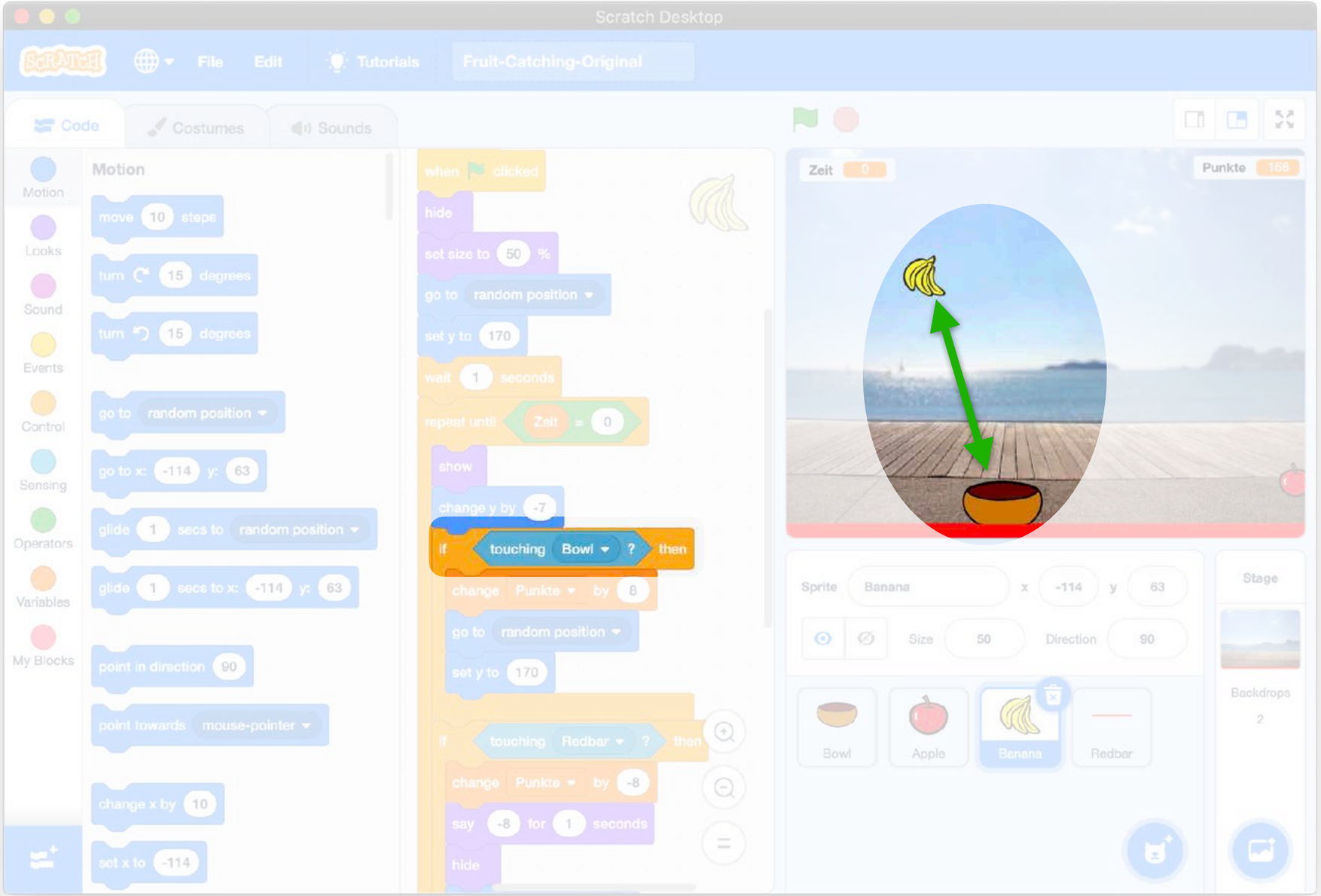
...

8 mod 4 = 0 → ClickSprite bear

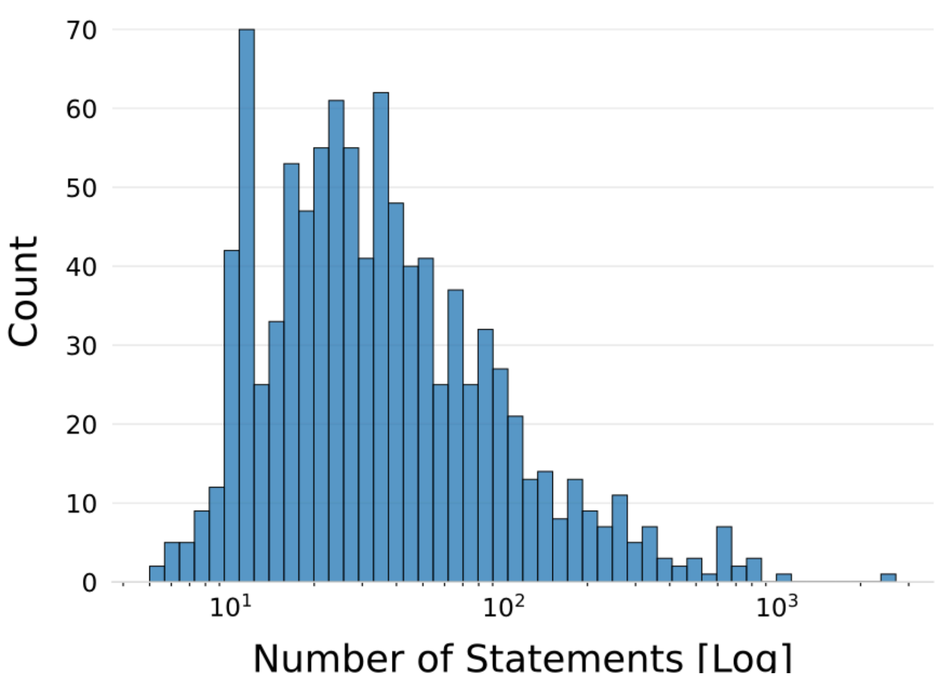


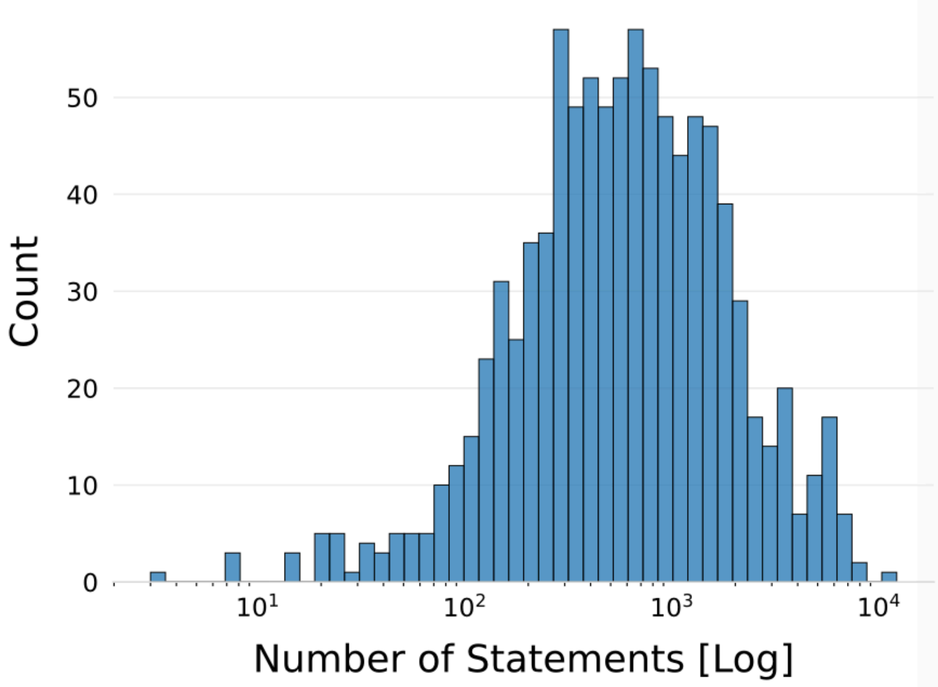
Fitness = Approach level + branch distance + control flow distance


Scratch Desktop
File
Edit
Tutorials
Fruit-Catching-Original
Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
move 10 steps
turn 15 degrees
turn 15 degrees
go to random position
go to x: -114 y: 63
glide 1 secs to random position
glide 1 secs to x: -114 y: 63
point in direction 90
point towards mouse-pointer
change x by 10
set x to -114
when clicked
hide
set size to 50 %
go to random position
set y to 170
wait 1 seconds
repeat until Zeit = 0
show
change y by -7
if touching Bowl ? then
change Punkte by 8
go to random position
set y to 170
if touching Redbar ? then
change Punkte by -8
say -8 for 1 seconds
hide
Zeit 0
Punkte 168
Sprite Banana
x -114
y 63
Size 50
Direction 90
Bowl
Apple
Banana
Redbar
Stage
Backdrops 2

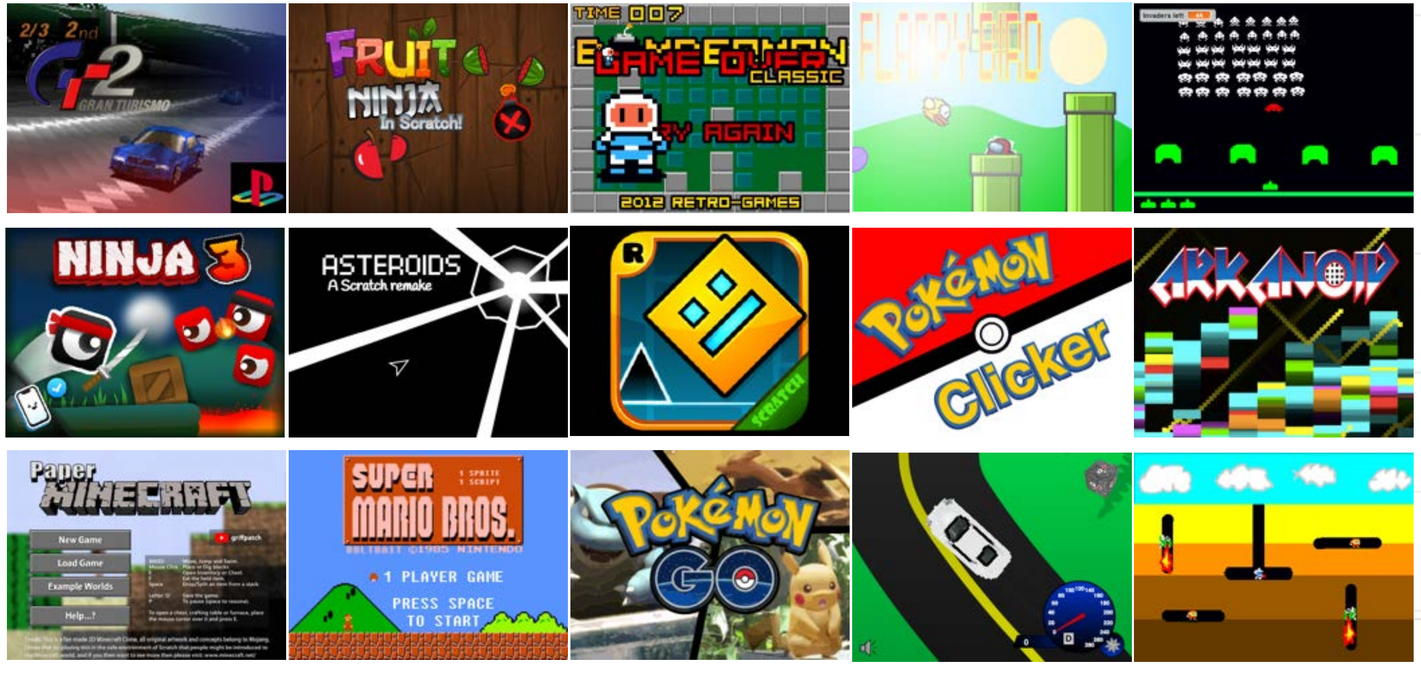

Scratch Desktop
File
Edit
Tutorials
Fruit-Catching-Original
Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
move 10 steps
turn 15 degrees
turn 15 degrees
go to random position
go to x: -114 y: 63
glide 1 secs to random position
glide 1 secs to x: -114 y: 63
point in direction 90
point towards mouse-pointer
change x by 10
set x to -114
when clicked
hide
set size to 50 %
go to random position
set y to 170
wait 1 seconds
repeat until Zeit = 0
show
change y by -7
if touching Bowl ? then
change Punkte by 8
go to random position
set y to 170
if touching Redbar ? then
change Punkte by -8
say -8 for 1 seconds
hide
Zeit 0
Punkte 168
Sprite Banana
x -114
y 63
Size 50
Direction 90
Bowl
Apple
Banana
Redbar
Stage
Backdrops 2

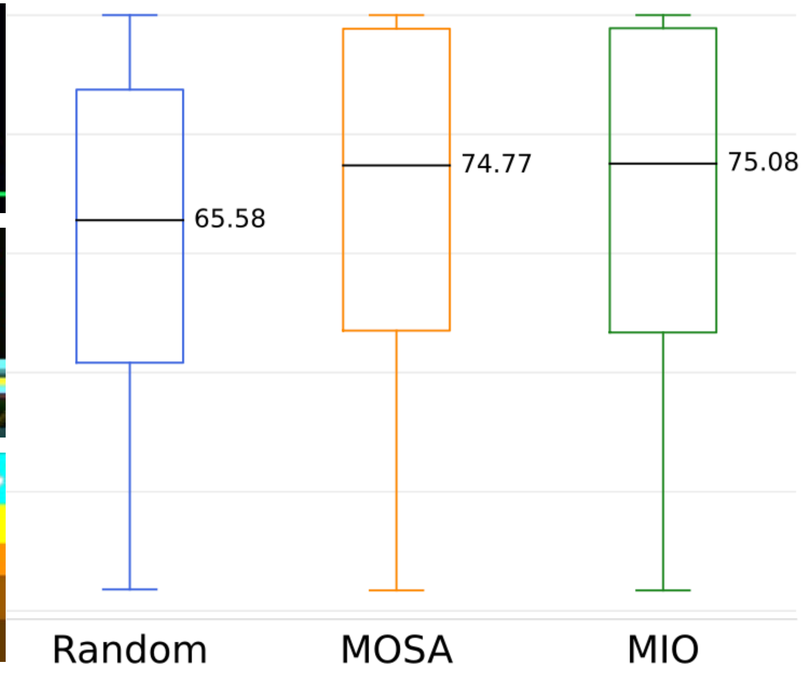
```
>

> node servant.js -d -a 10 -k -g -s example.sb3 -c mosa_config.json
```

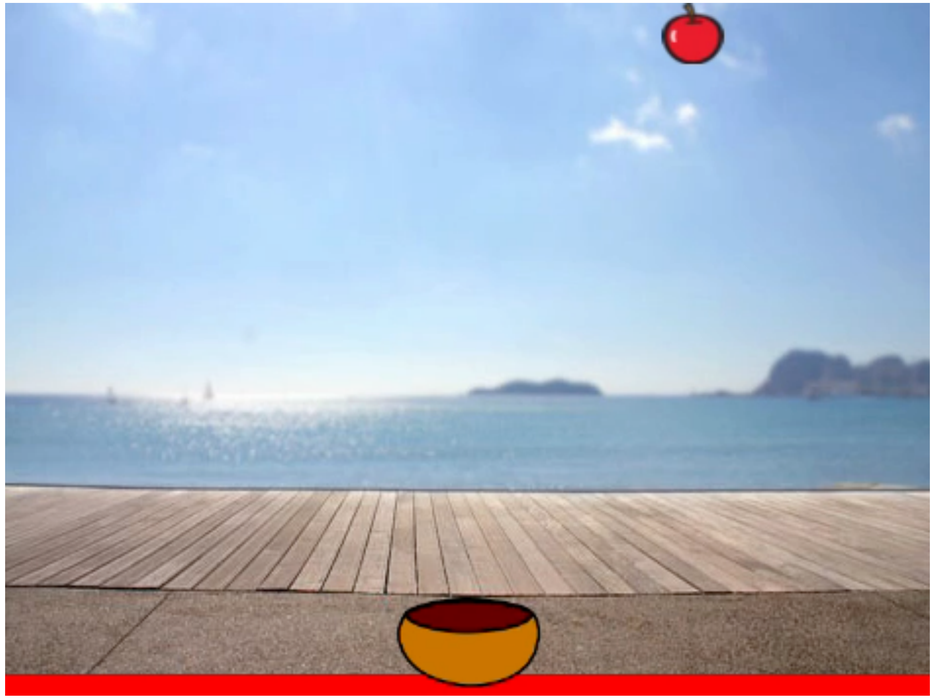

Count
Number of Statements [Log]

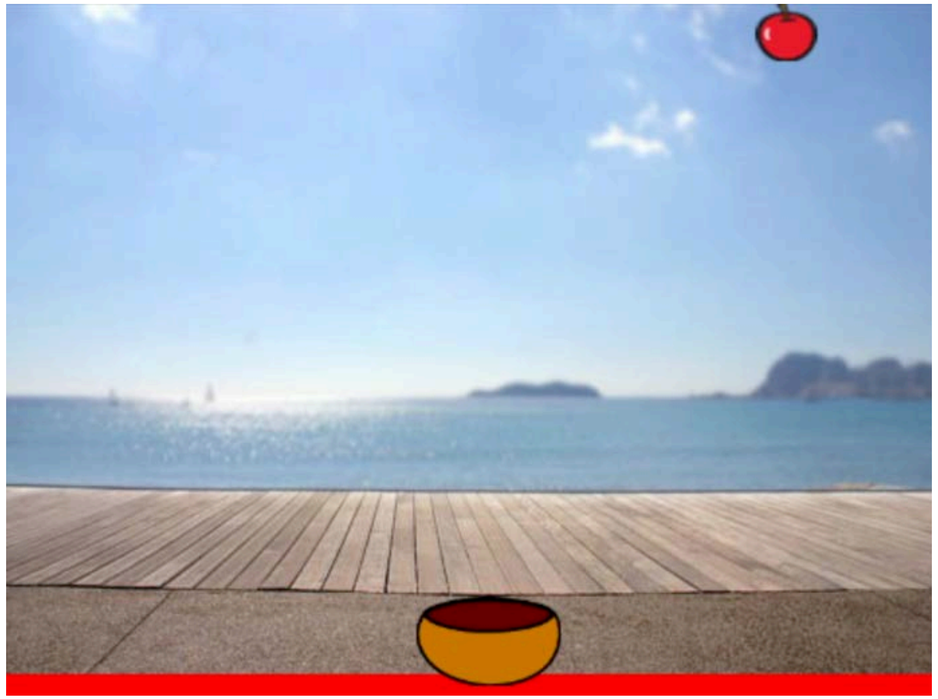

Count
Number of Statements [Log]

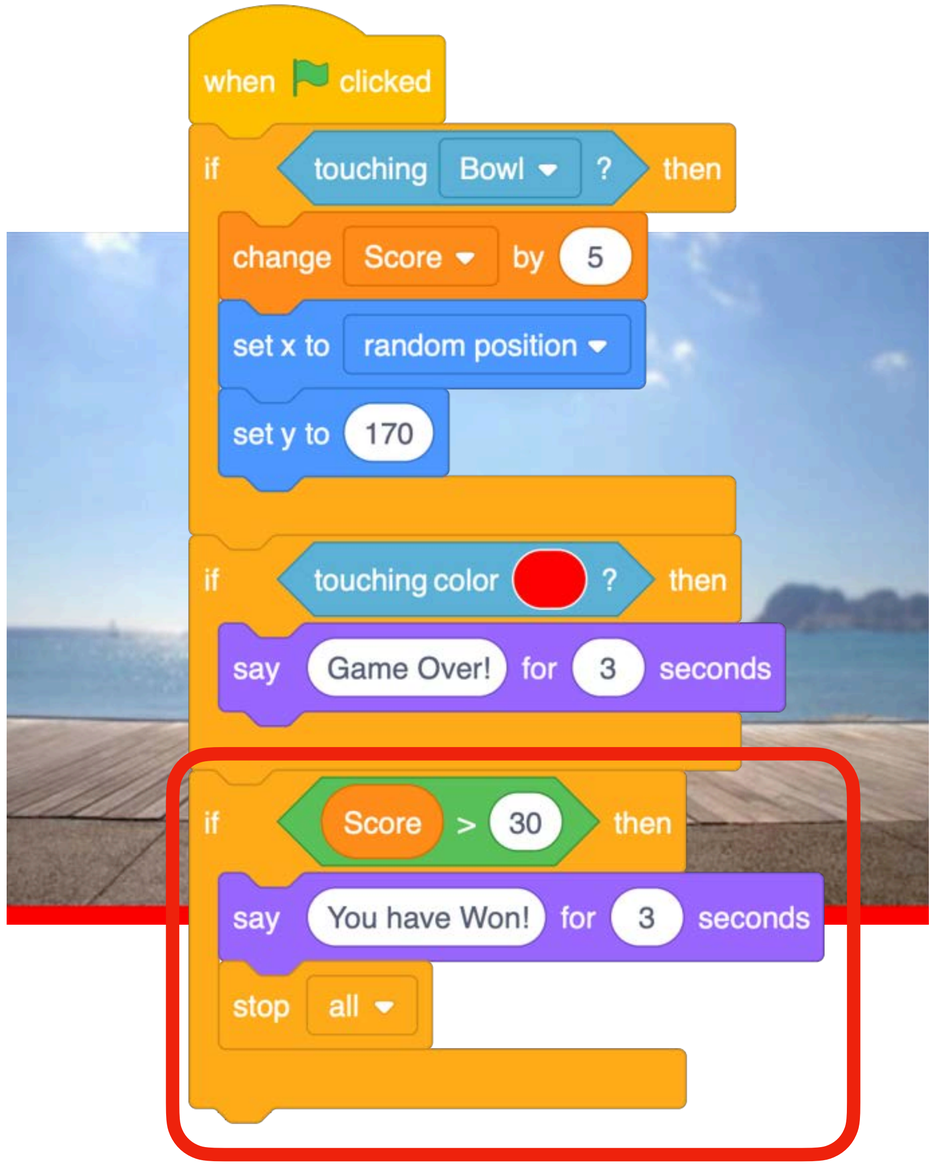
(b) Top1000

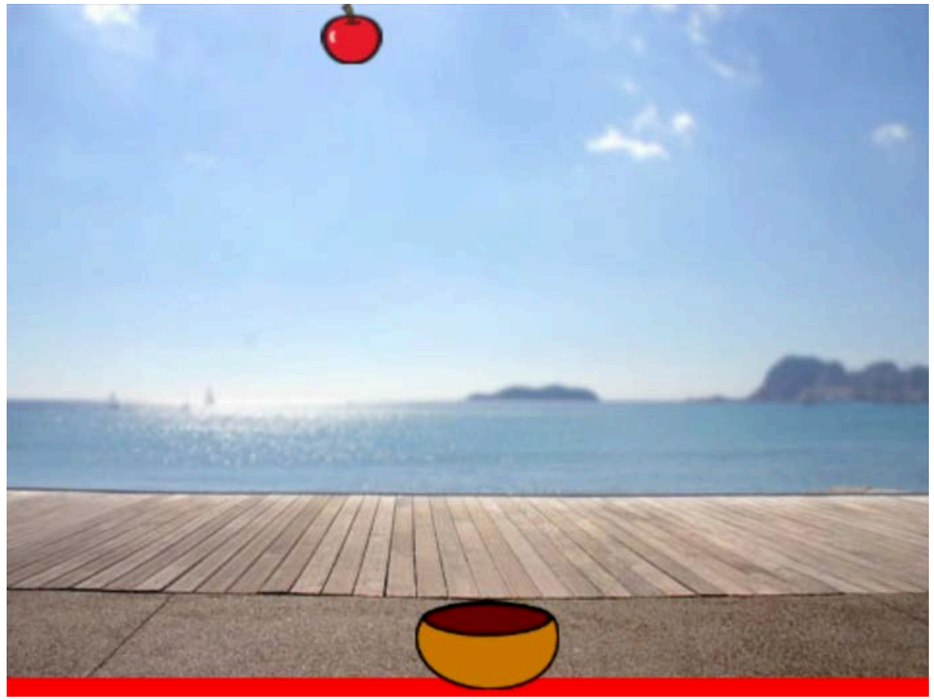

NINJA 3
ASTEROIDS
A Scratch remake
Pokémon Clicker
ARKANOID
Paper MINECRAFT
New Game
Load Game
Example Worlds
Help...!
SUPER MARIO BROS.
1 PLAYER GAME
PRESS SPACE TO START
Pokémon GO

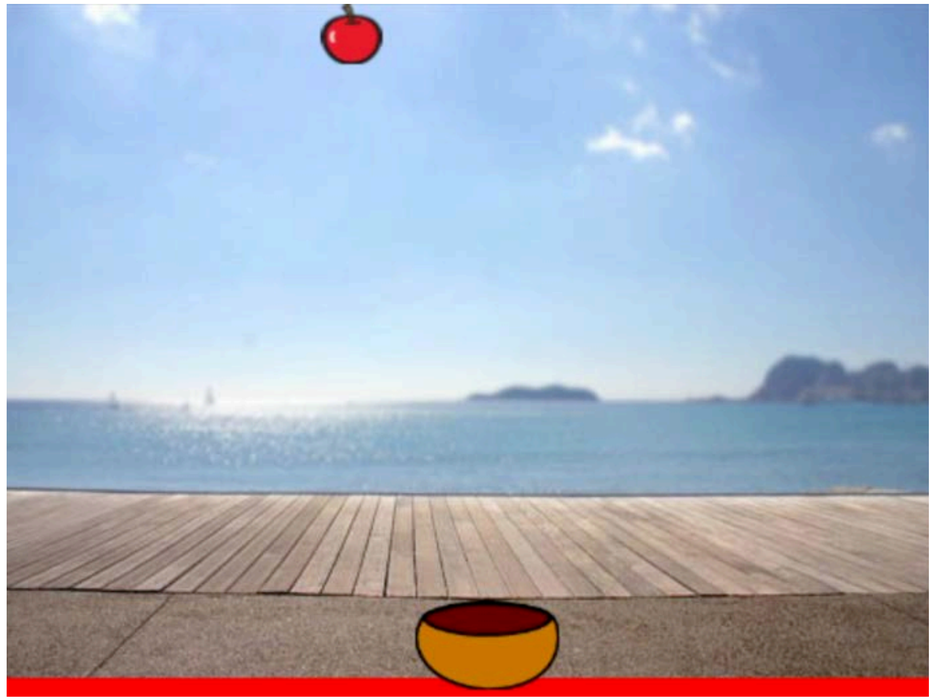

65.58
74.77
75.08
Random
MOSA
MIO






when clicked
if touching Bowl ? then
change Score by 5
set x to random position
set y to 170
if touching color ? then
say Game Over! for 3 seconds
if Score > 30 then
say You have Won! for 3 seconds
stop all





Reinforcement Learning vs. SBST

Reward function

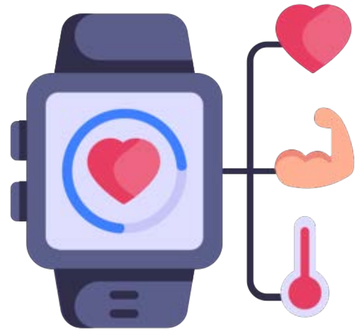
Hand-crafted

Usually winning state

Sparse

Single objective

Focus on exploitation



Fitness function

Automatically generated

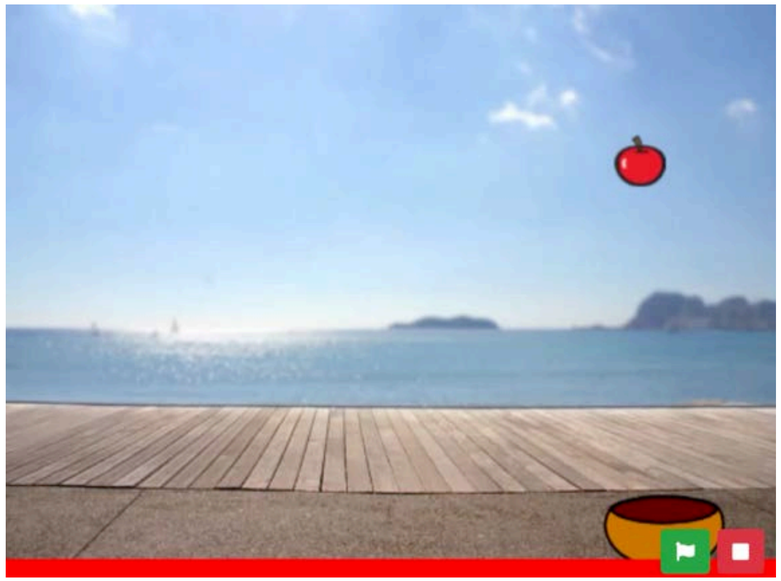
Based on coverage

Dense

Many objectives

Focus on exploration

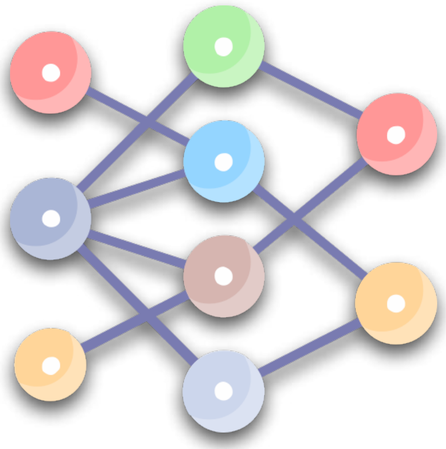
Neurevolution-based Testing with WHISKER



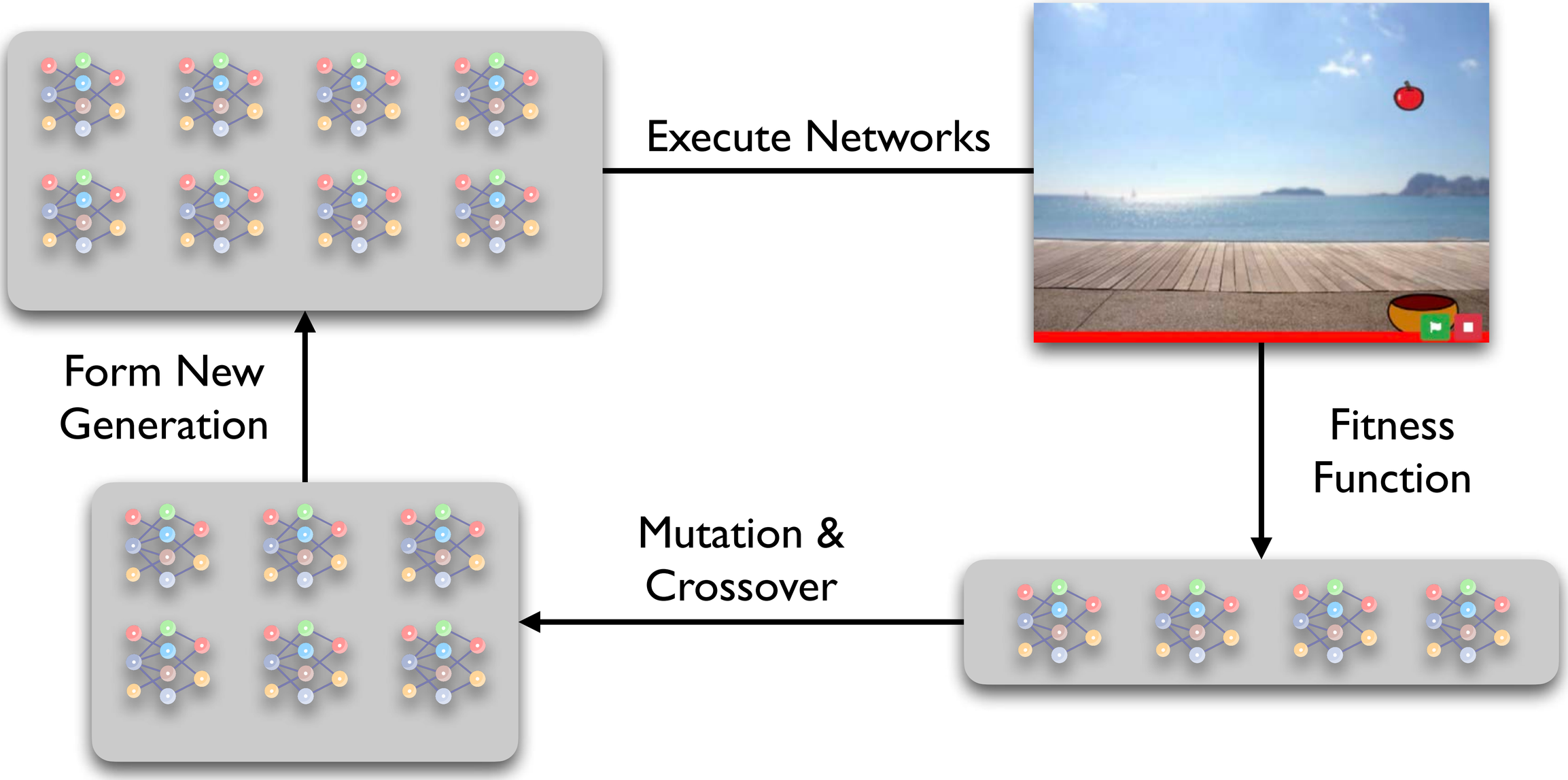
Learn to play

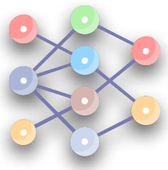
Validate behaviour

Dynamic Test Suites



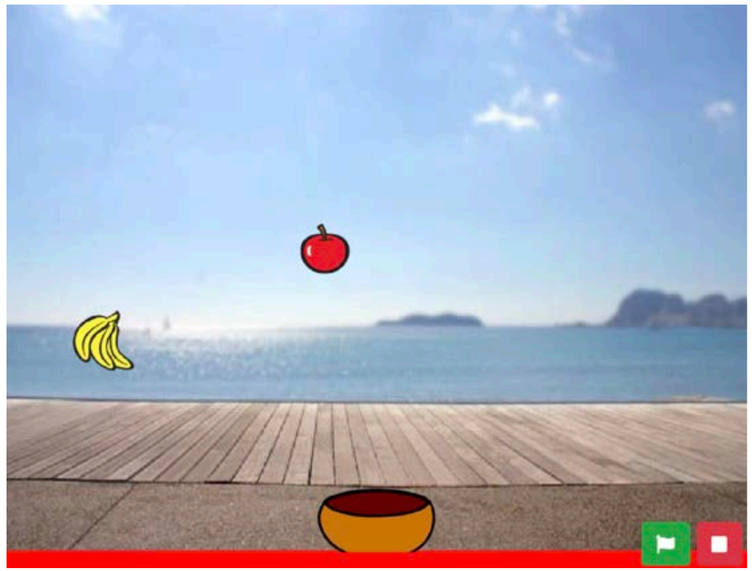
Generating Network Suites Using Neuroevolution

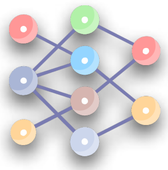




Generating Dynamic Test Suites

1. Select a target statement

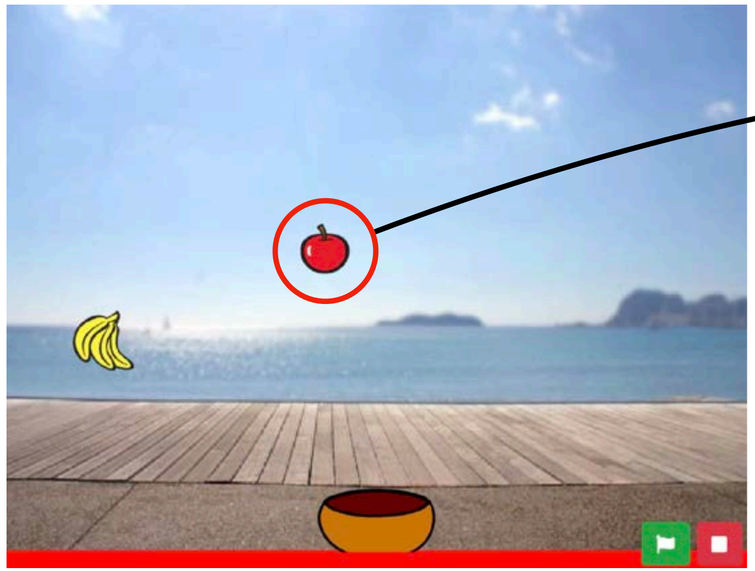




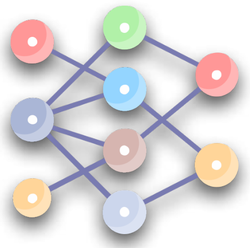
Generating Dynamic Test Suites

2. Optimise networks to cover the selected statement using Neuroevolution

➡ Fitness = distance to target statement



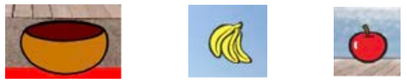
Neuroevolution

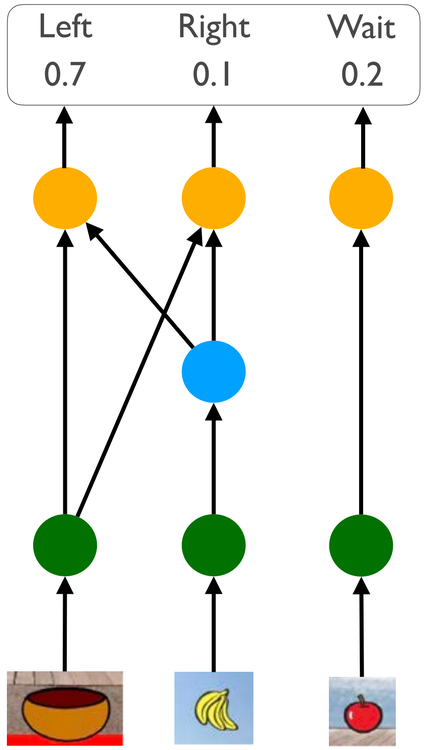


Send Event

Left
0.7

Extract Sprites





Genotype

Node Gene List

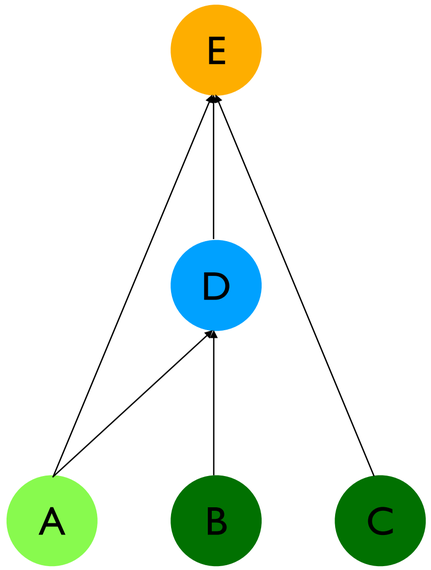
A	B	C	D	E
Bias	Input	Input	Hidden	Output

+

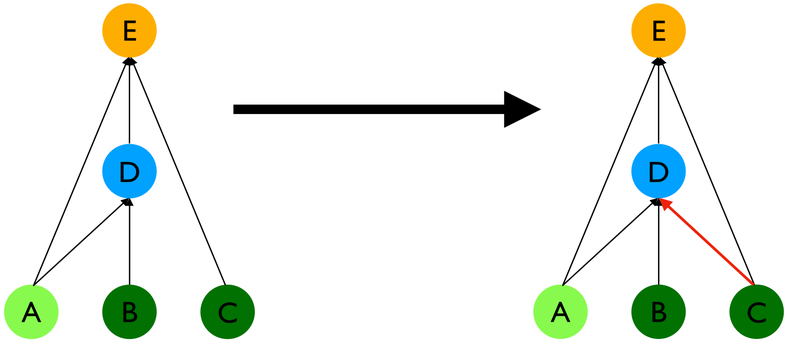
Connection Gene List

1	2	3	4	5
A → E	A → D	B → D	C → E	D → E

Phenotype



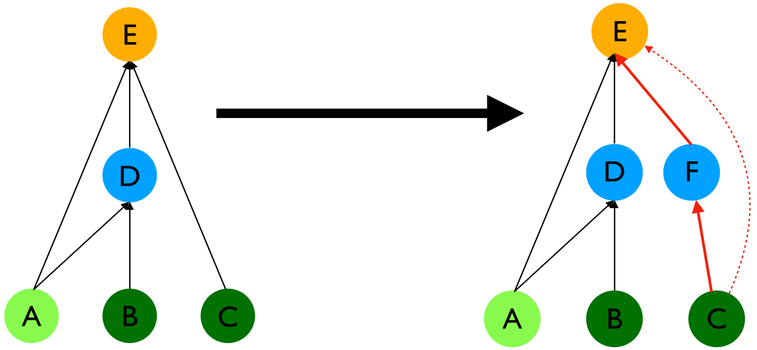
Add Connection Mutation



1	2	3	4	5
A → E	A → D	B → D	C → E	D → E

1	2	3	4	5	6
A → E	A → D	B → D	C → E	D → E	C → D

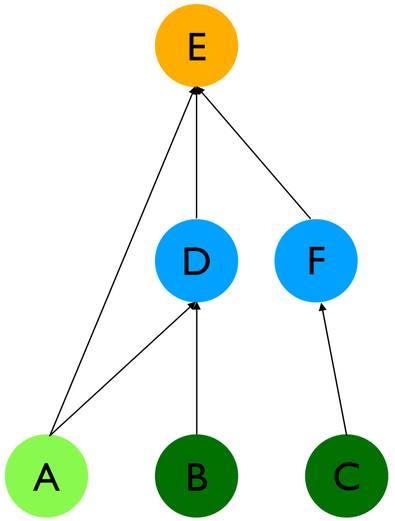
Add Node Mutation



1	2	3	4	5
A → E	A → D	B → D	C → E	D → E

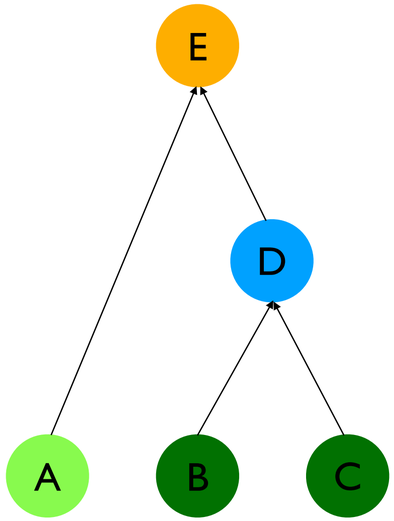
1	2	3	4	5	7	8
A → E	A → D	B → D	C → E	D → E	C → F	F → E

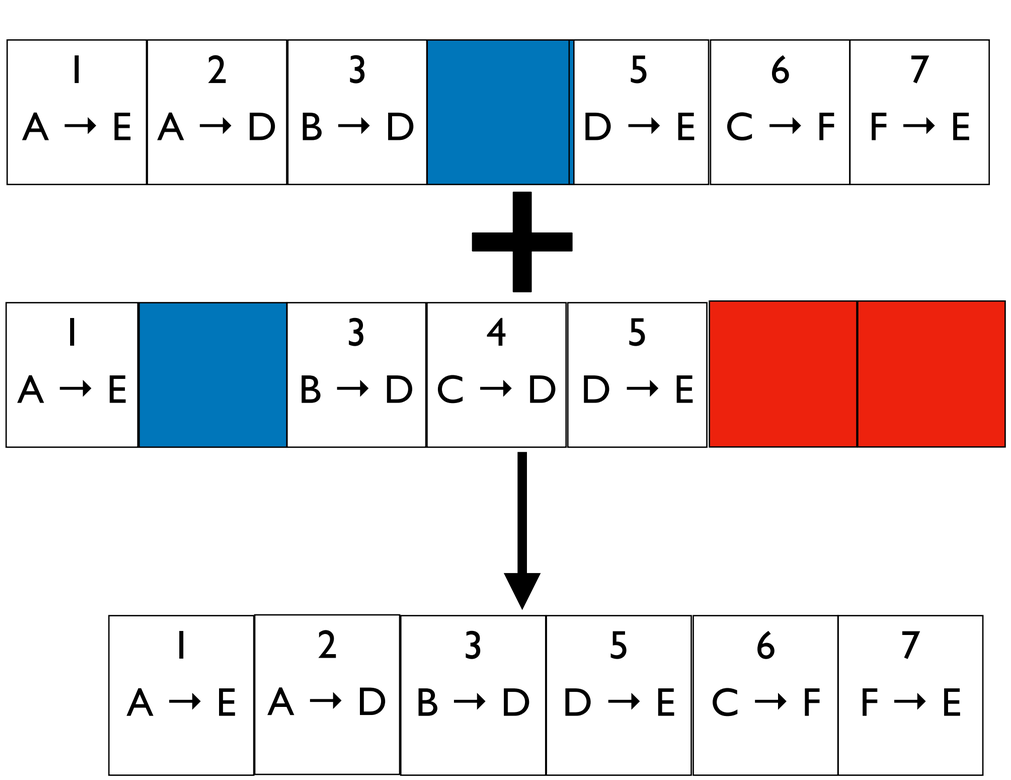
1	2	3	5	6	7
A → E	A → D	B → D	D → E	C → F	F → E

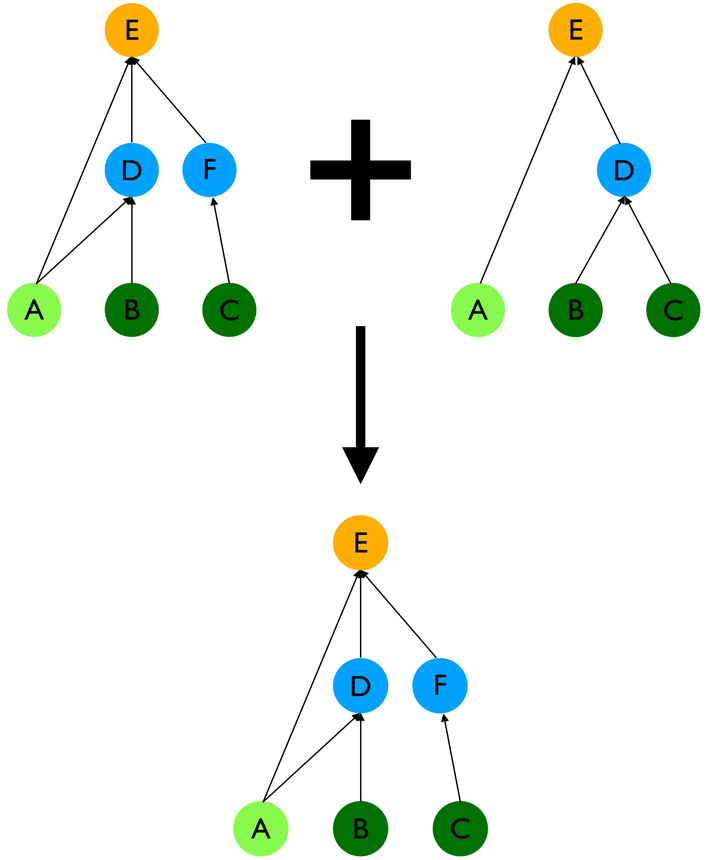


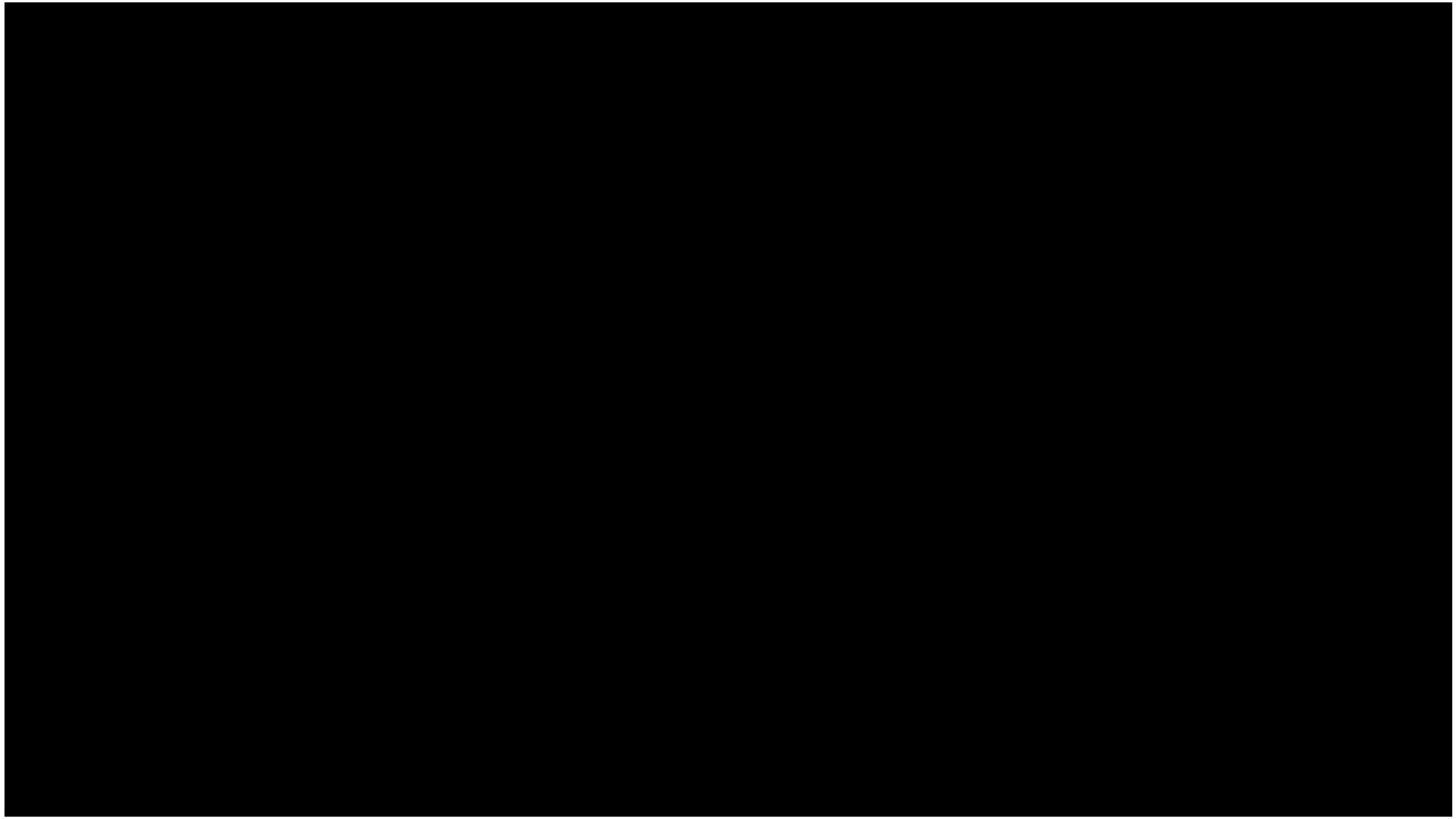
+

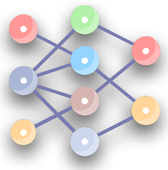
1	3	4	5
A → E	B → D	C → D	D → E




1
A → E
2
A → D
3
B → D
5
D → E
6
C → F
7
F → E
1
A → E
3
B → D
4
C → D
5
D → E
1
A → E
2
A → D
3
B → D
5
D → E
6
C → F
7
F → E


E
D
F
A
B
C
E
D
A
B
C
E
D
F
A
B
C



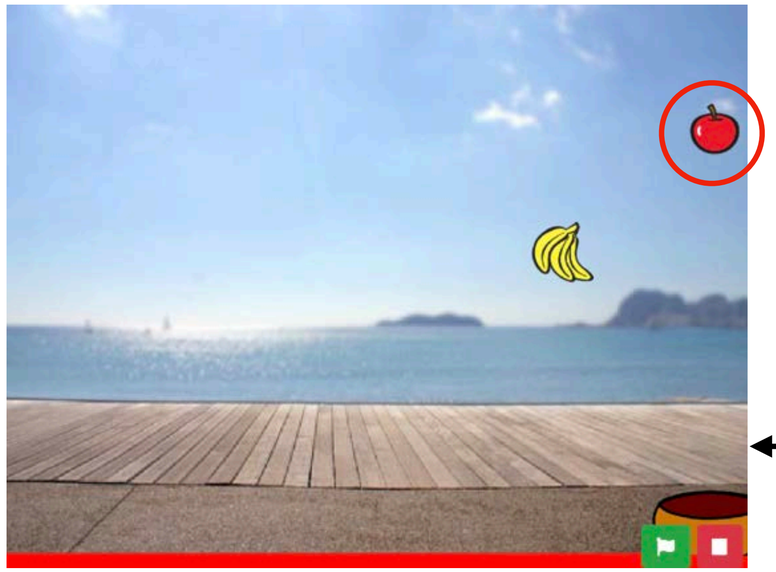


Generating Dynamic Test Suites

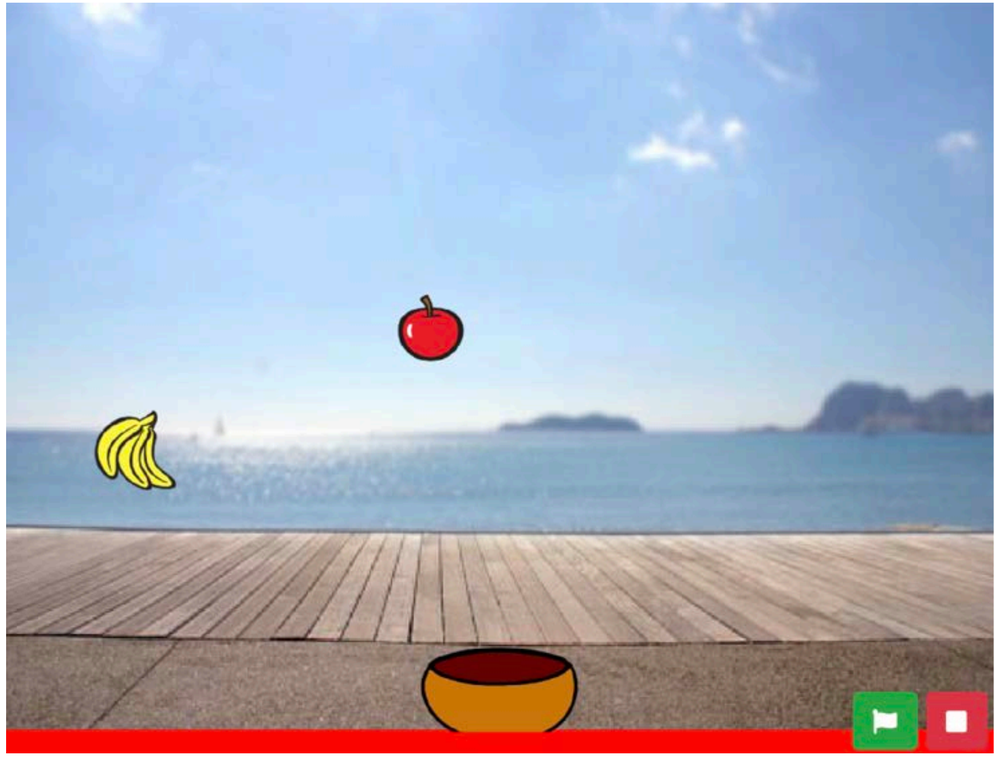
3. Validate and improve the robustness of networks

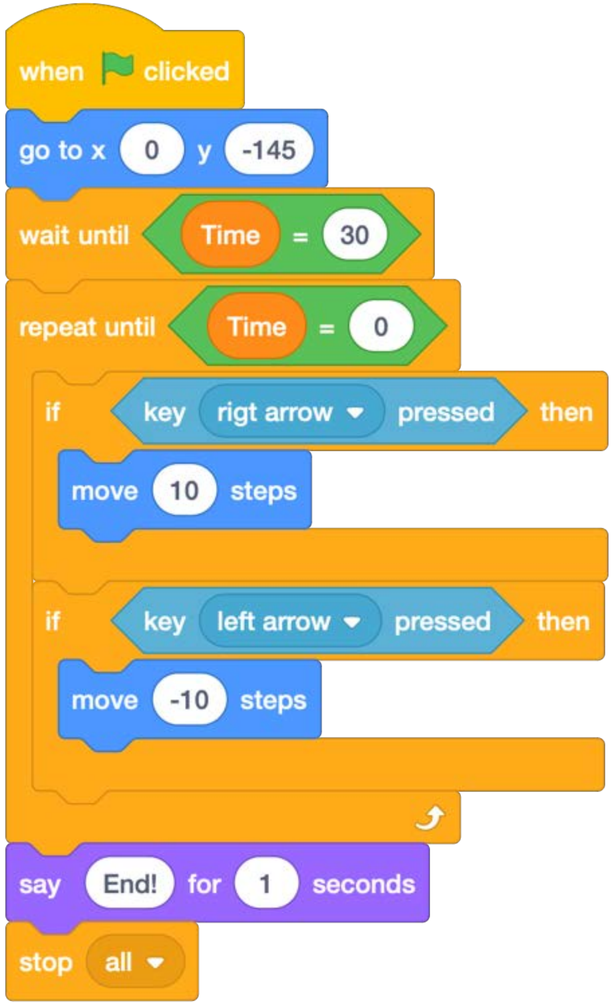
➡ Cover 

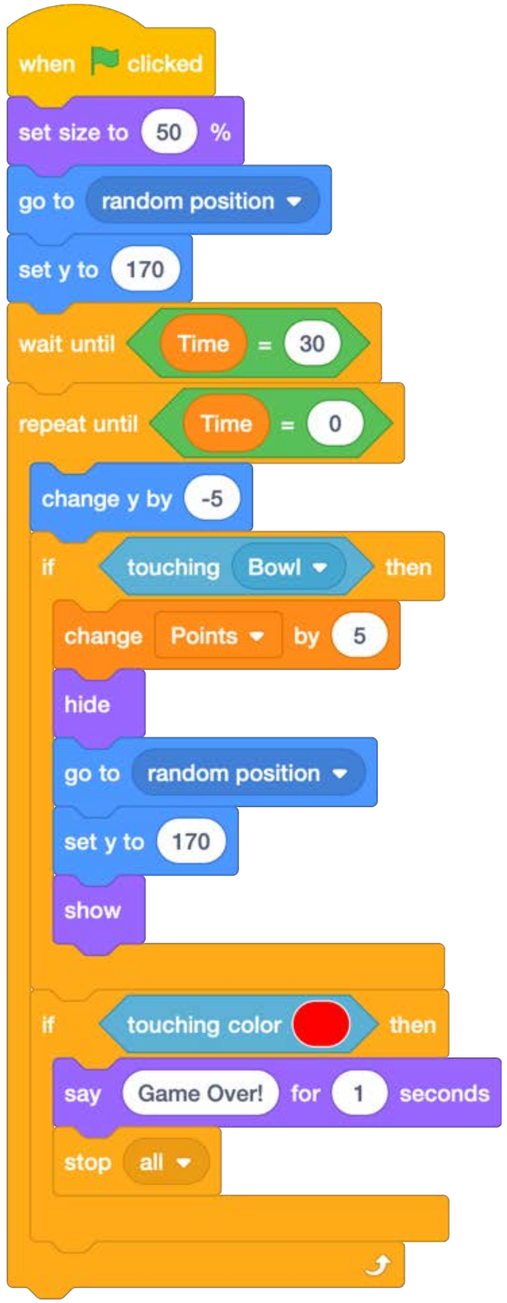
multiple times using different seeds



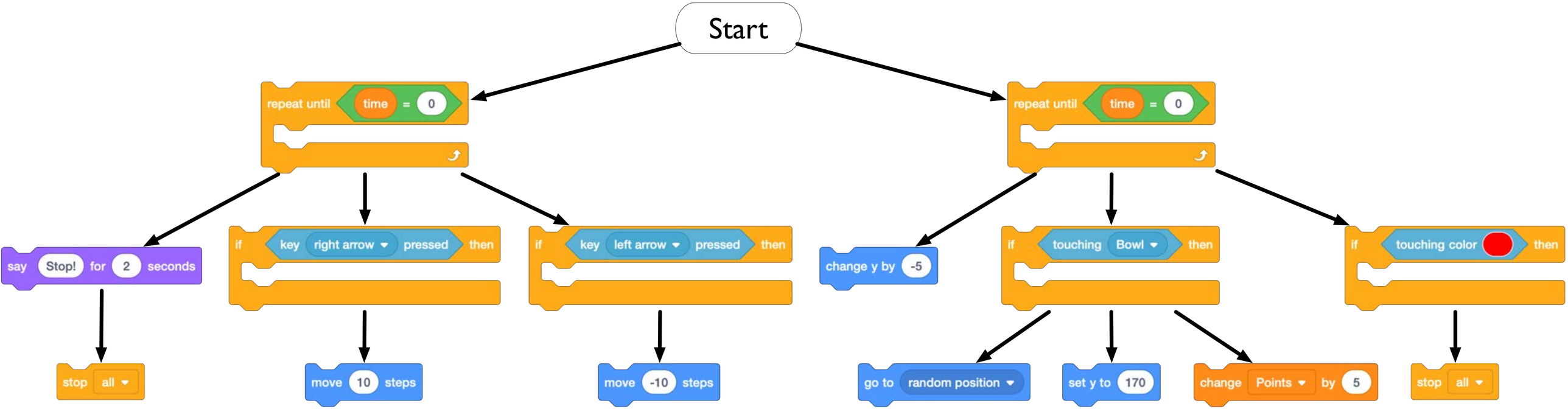
Selecting Target Statements







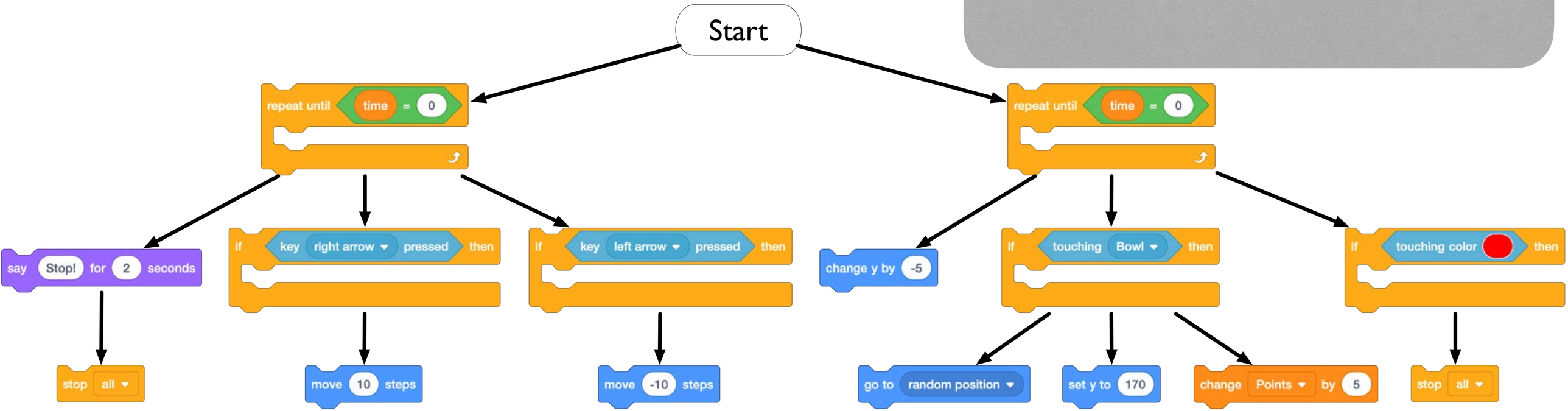
Selecting Target Statements



Selecting Target Statements

Dynamic Test Suite

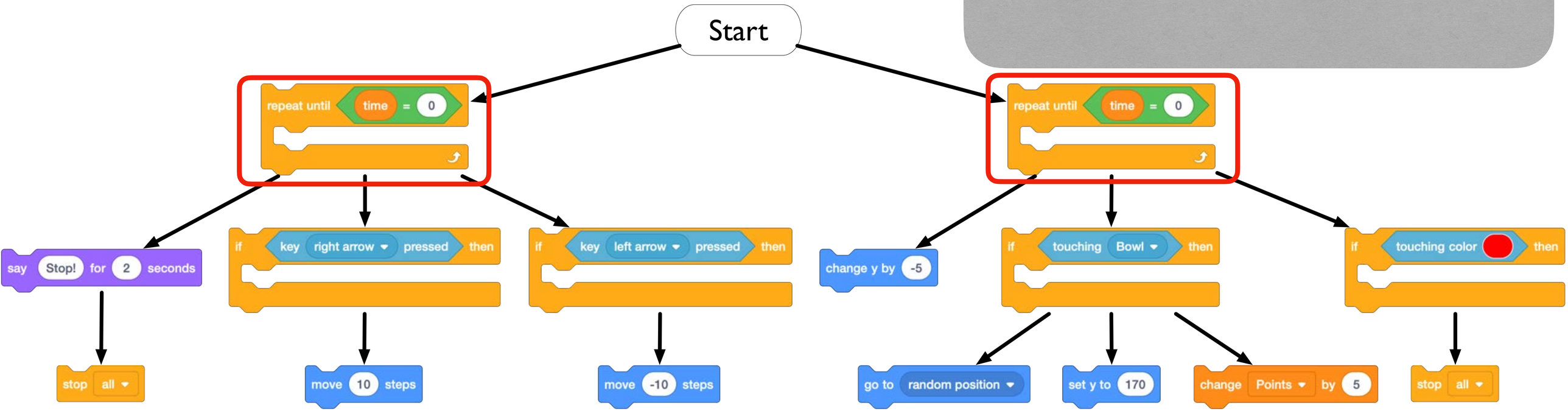
- Select direct children of covered control nodes



Selecting Target Statements

Dynamic Test Suite

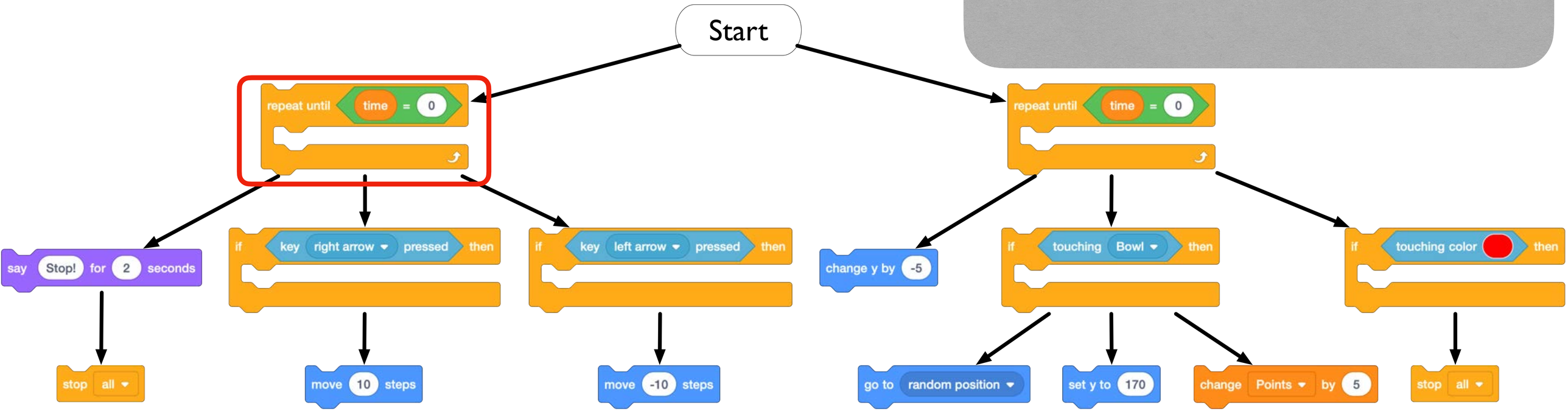
- Select direct children of covered control nodes



Selecting Target Statements

Dynamic Test Suite

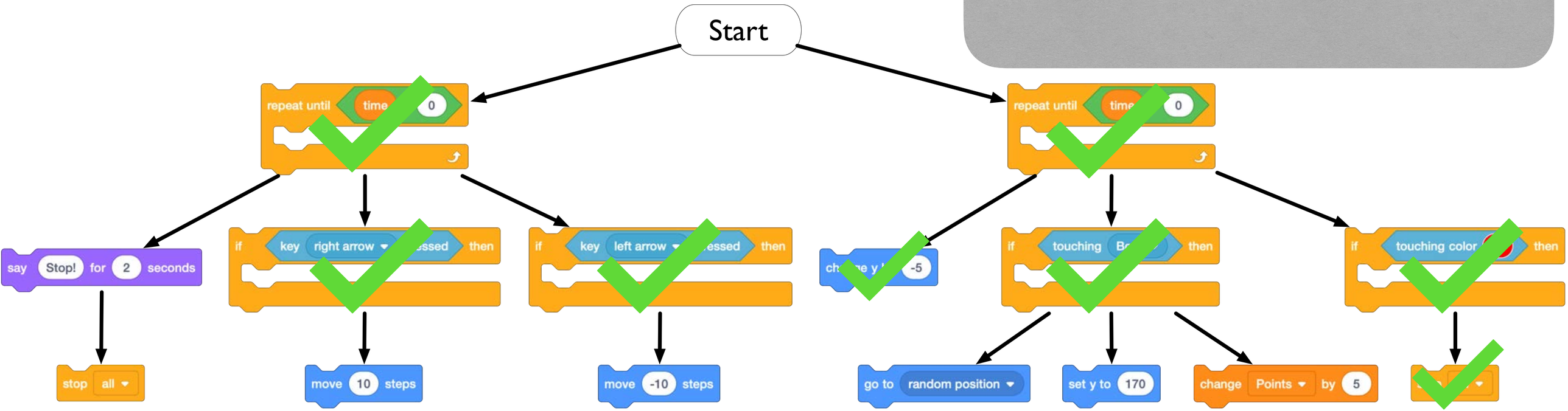
- Select direct children of covered control nodes



Selecting Target Statements

Dynamic Test Suite

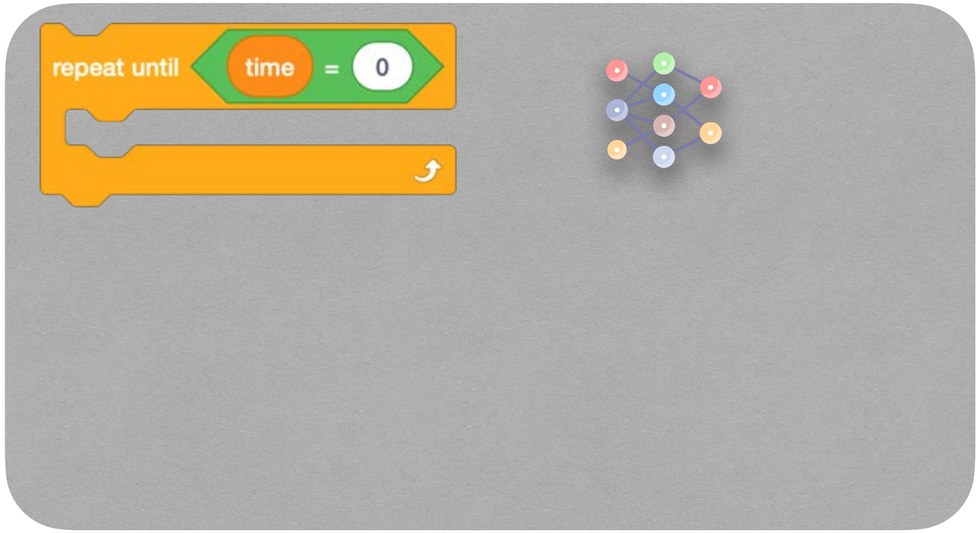
- Select direct children of covered control nodes

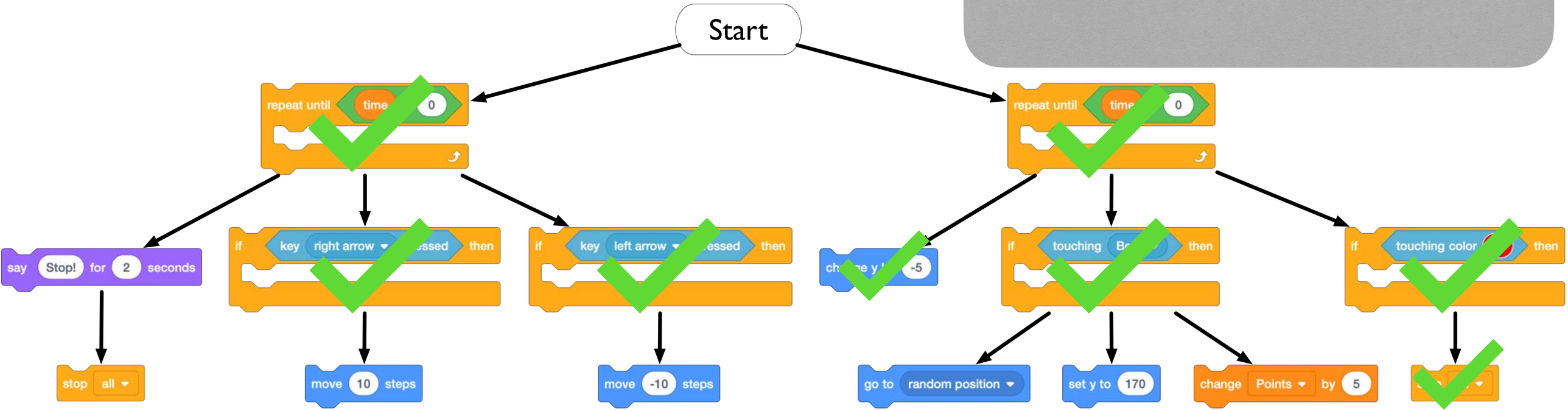


Selecting Target Statements

- Select direct children of covered control nodes

Dynamic Test Suite



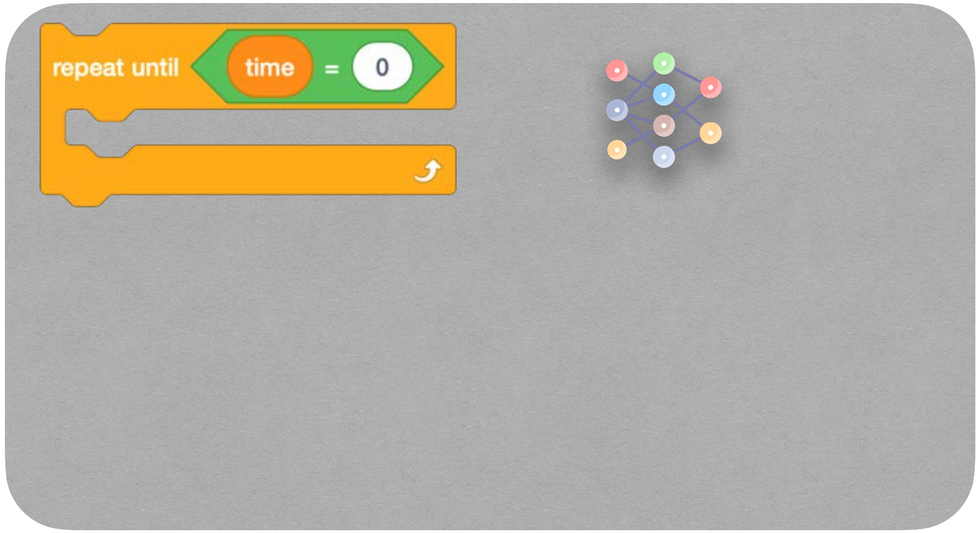


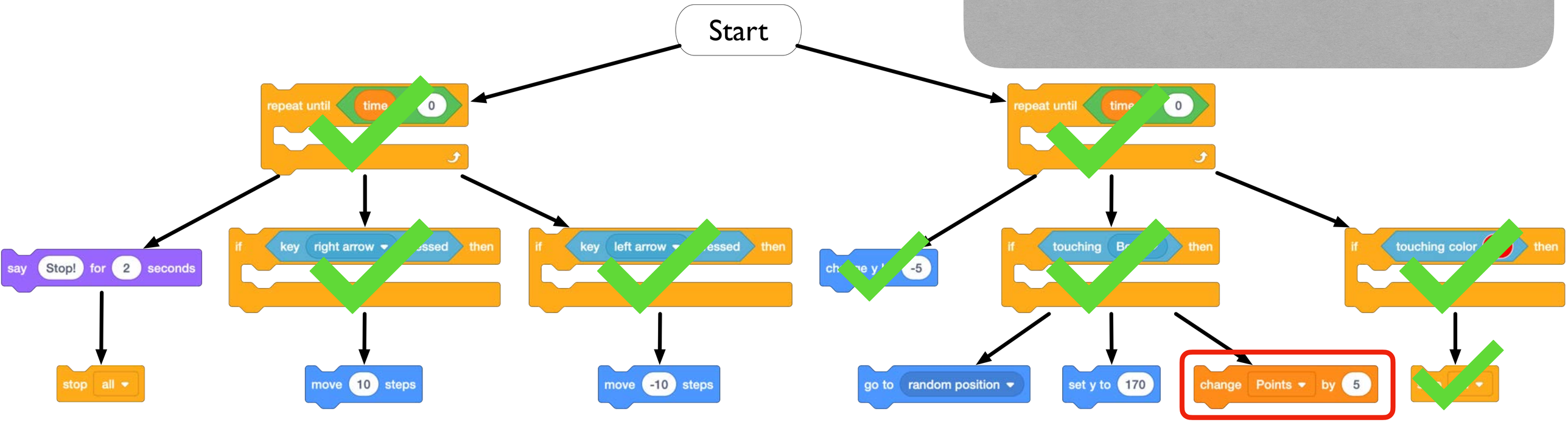
Selecting Target Statements

- Select direct children of covered control nodes

  ➡Deep nodes require meaningful gameplay

Dynamic Test Suite

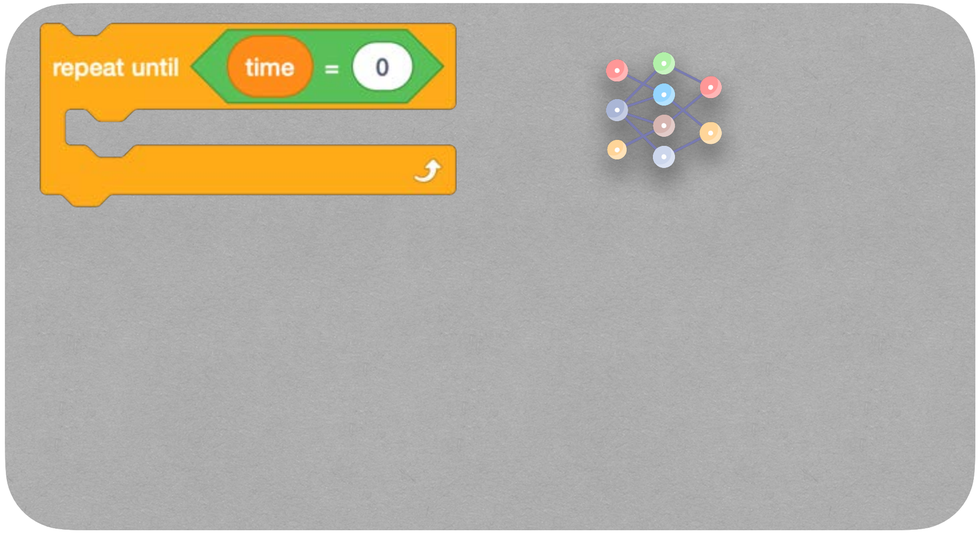


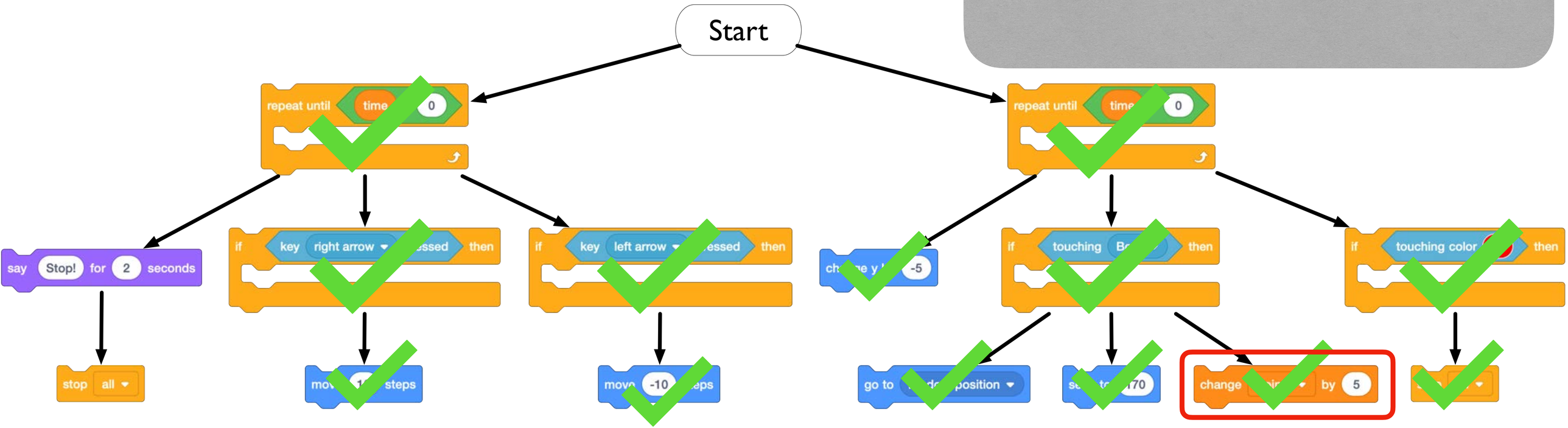


Selecting Target Statements

- Select direct children of covered control nodes
  ➡Deep nodes require meaningful gameplay

Dynamic Test Suite



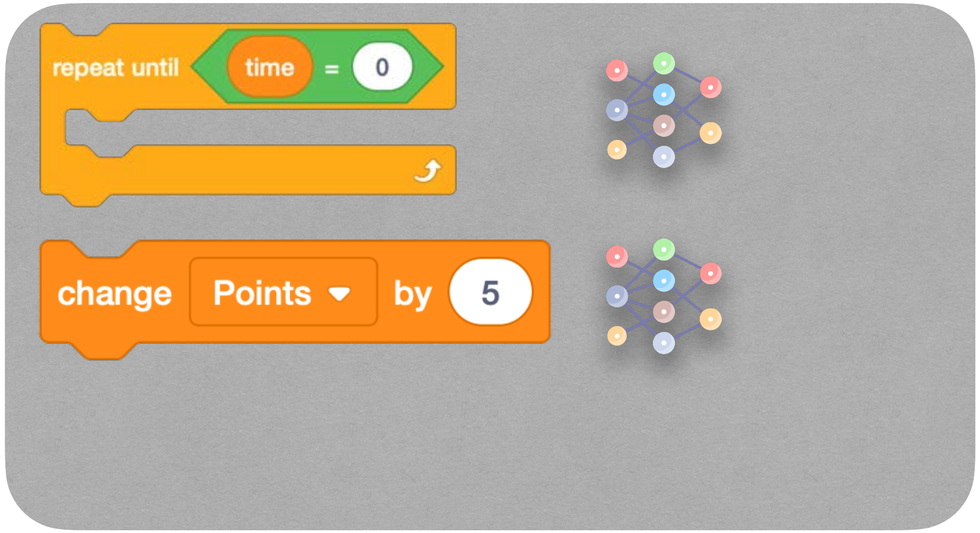


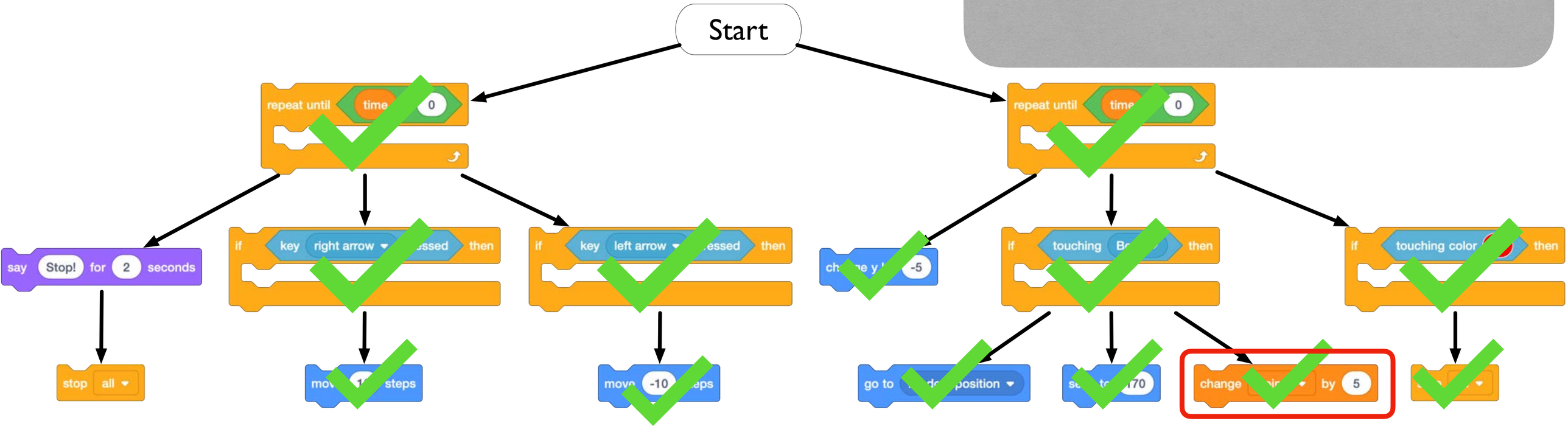
Selecting Target Statements

- Select direct children of covered control nodes

  ➡Deep nodes require meaningful gameplay

Dynamic Test Suite



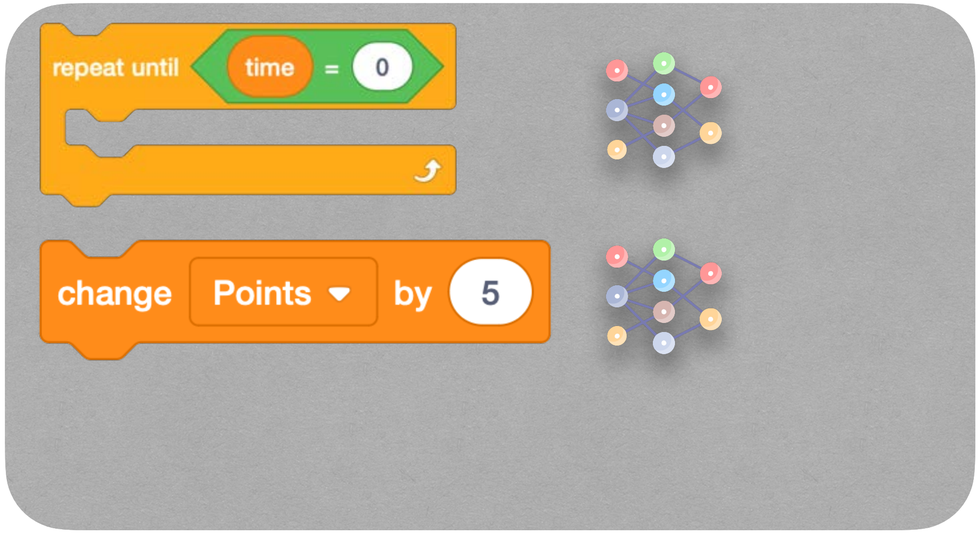


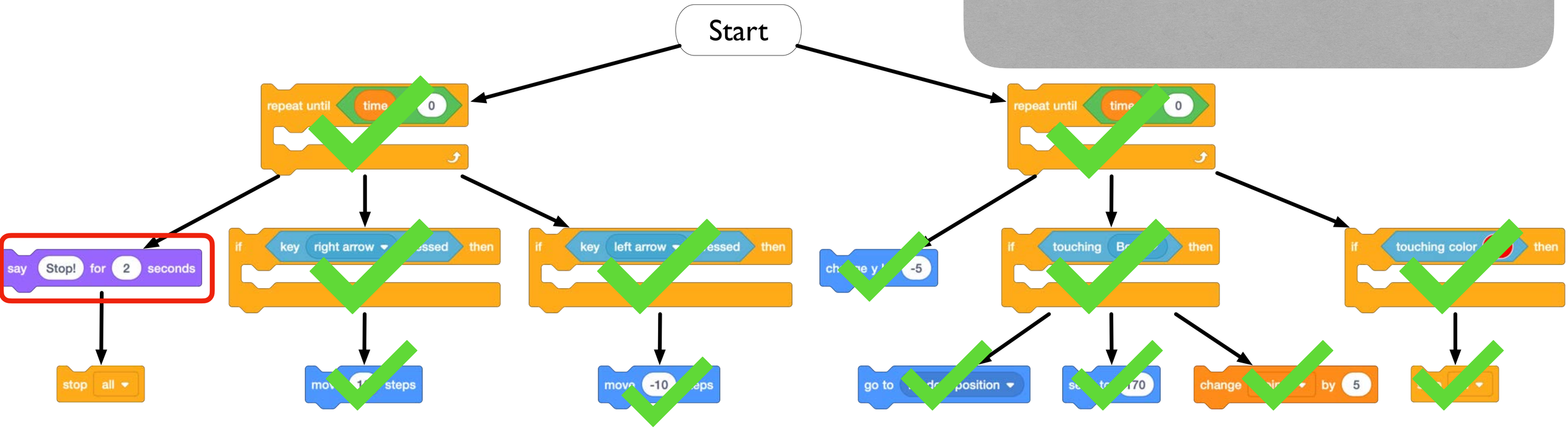
Selecting Target Statements

- Select direct children of covered control nodes

  ➡Deep nodes require meaningful gameplay

Dynamic Test Suite

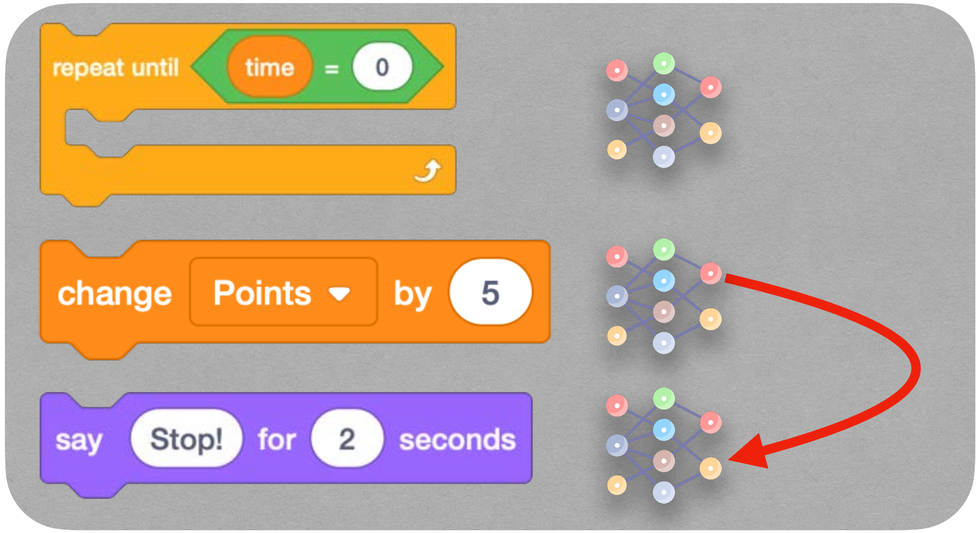


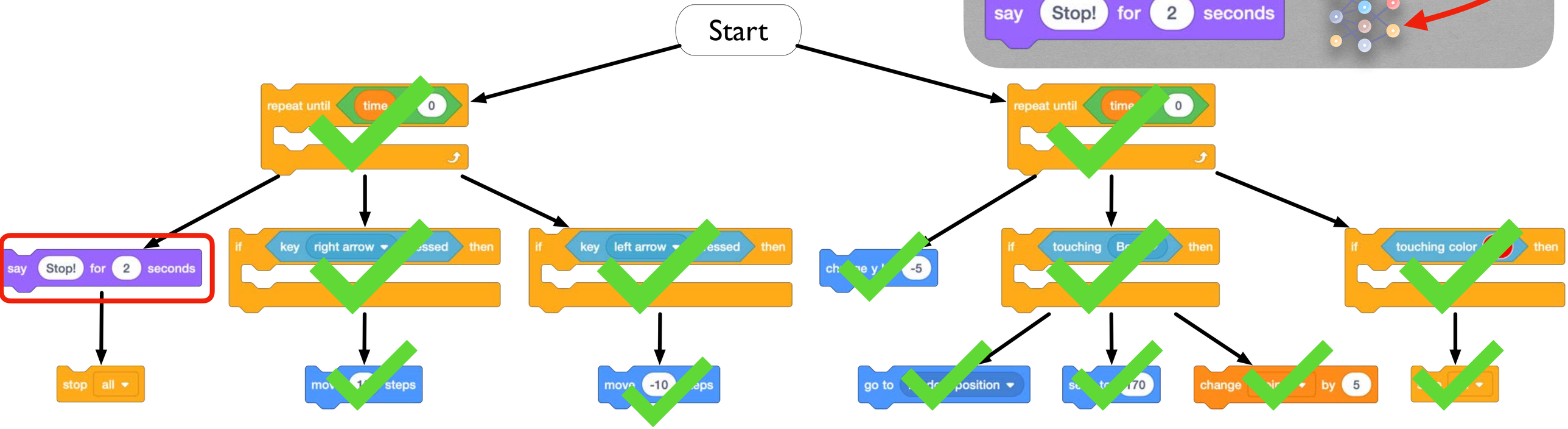


Selecting Target Statements

- Select direct children of covered control nodes
  - ➡Deep nodes require meaningful gameplay
  - ➡Build upon previously optimised networks

Dynamic Test Suite

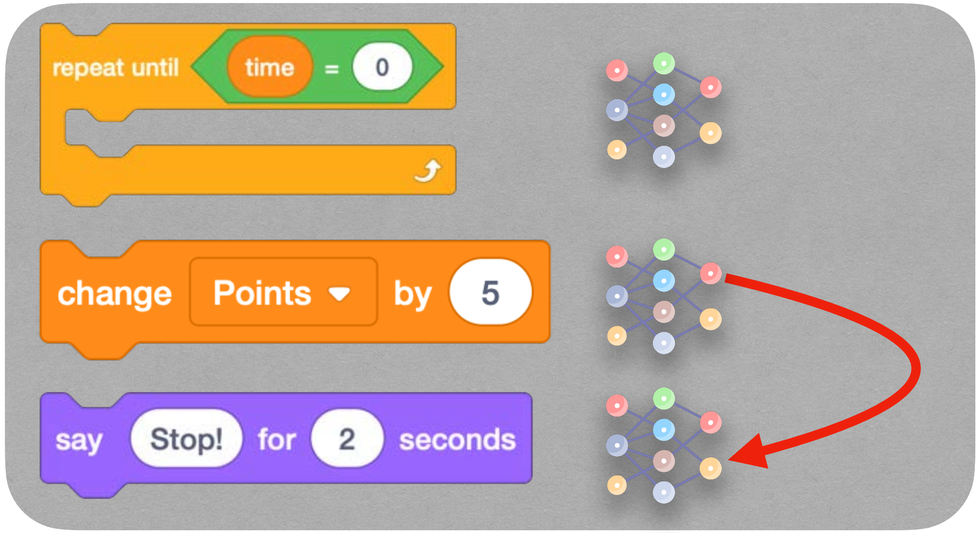


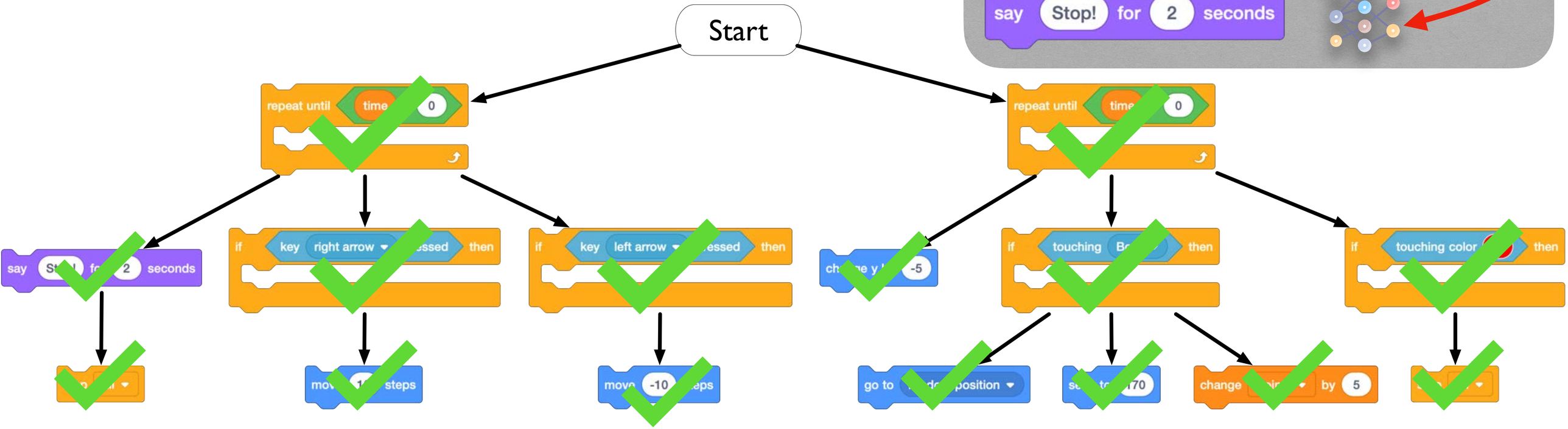


Selecting Target Statements

- Select direct children of covered control nodes
  - ➡Deep nodes require meaningful gameplay
  - ➡Build upon previously optimised networks

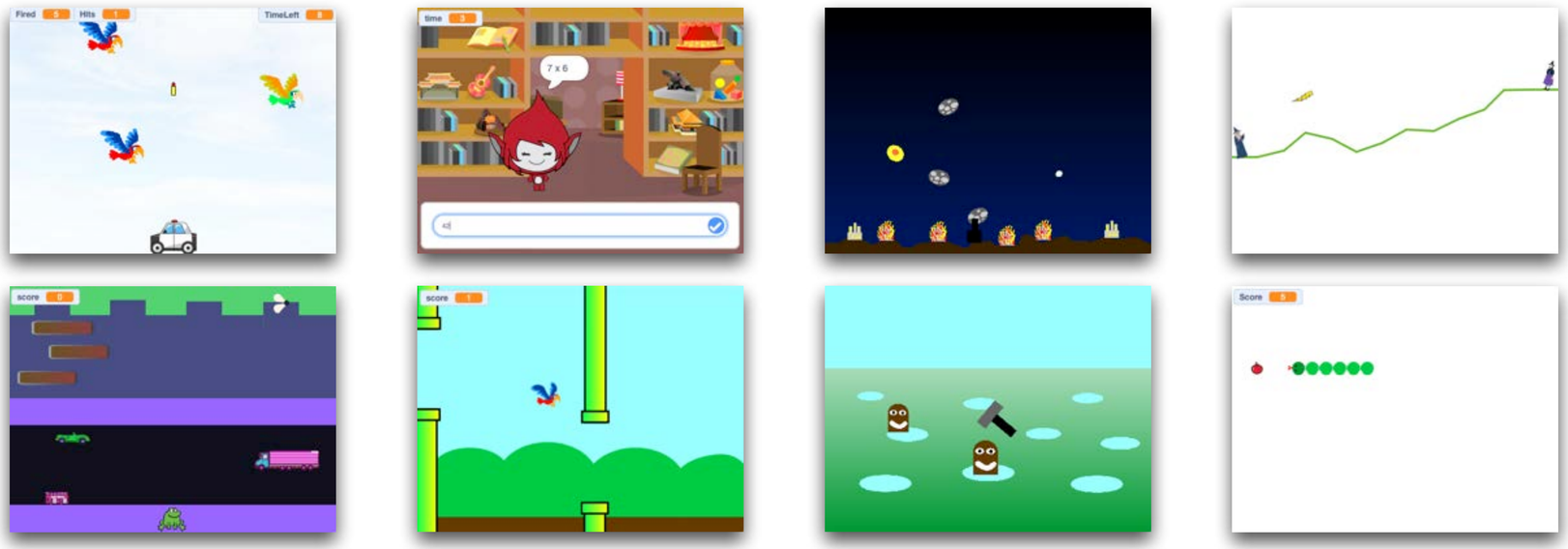
Dynamic Test Suite





Evaluation of Neatest

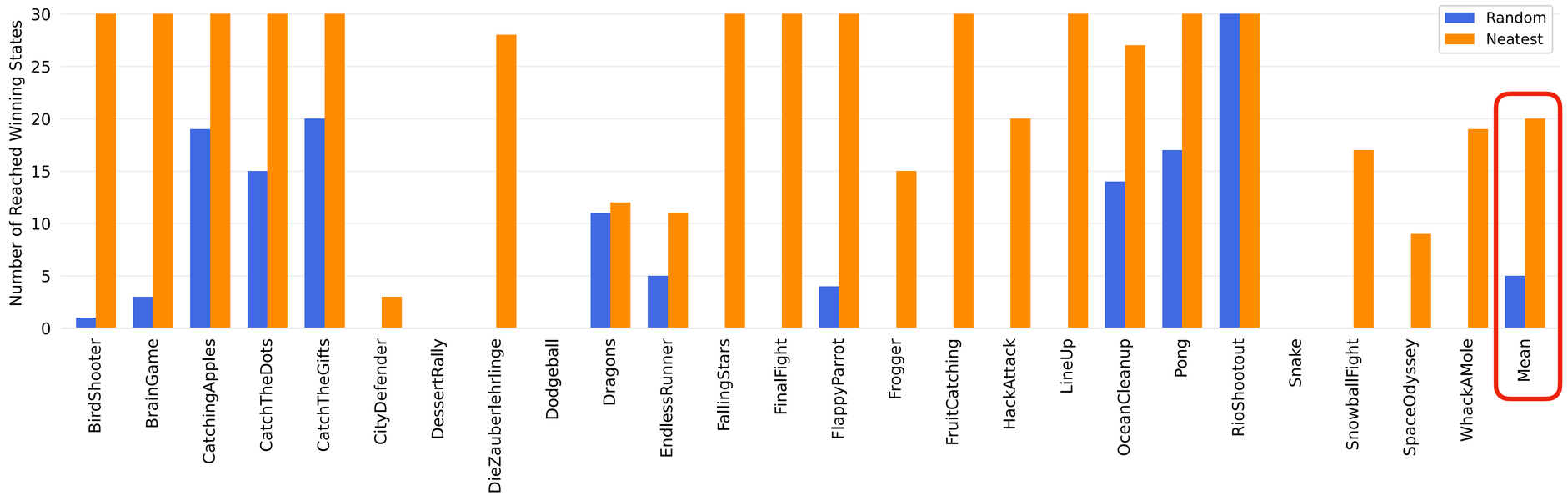
- Dataset of 25 Scratch games with unique challenges and gameplay mechanics



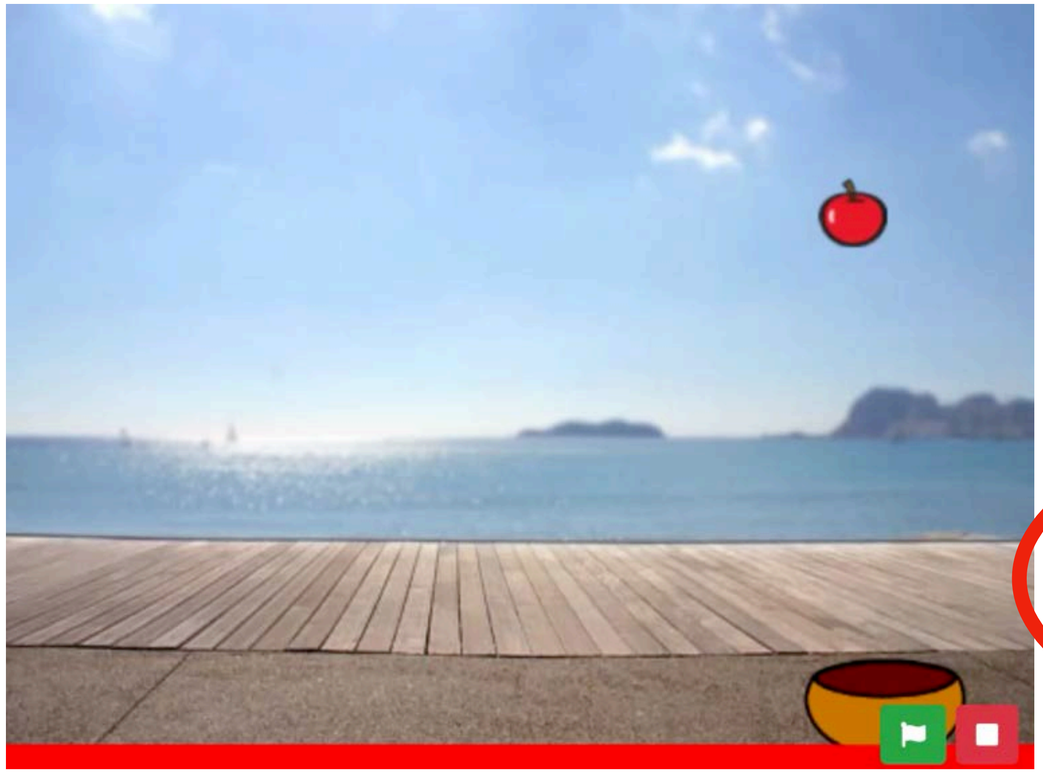
Evaluation of Neatest

- Compares Neatest with random test generation baseline
- Statements are covered if generated test passes the robustness check 10 times

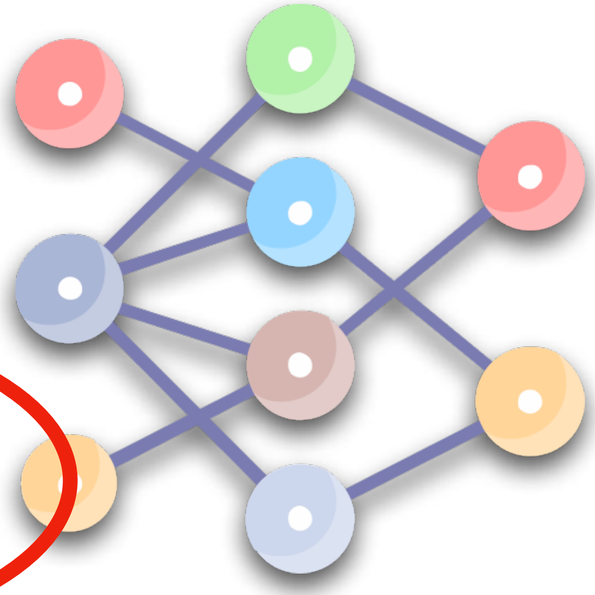
➡ Neatest wins games on average 20/30 times



Neatest



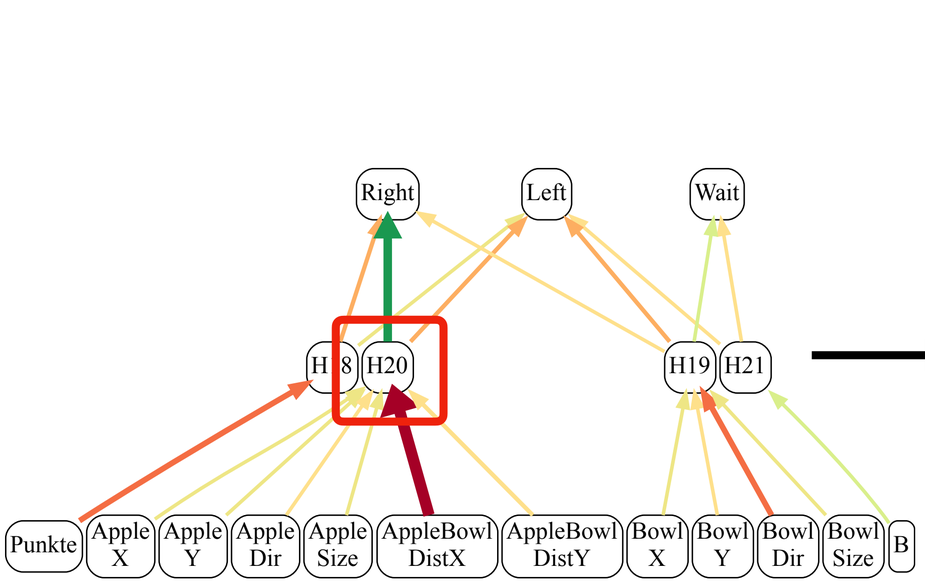
Dynamic Test Suites

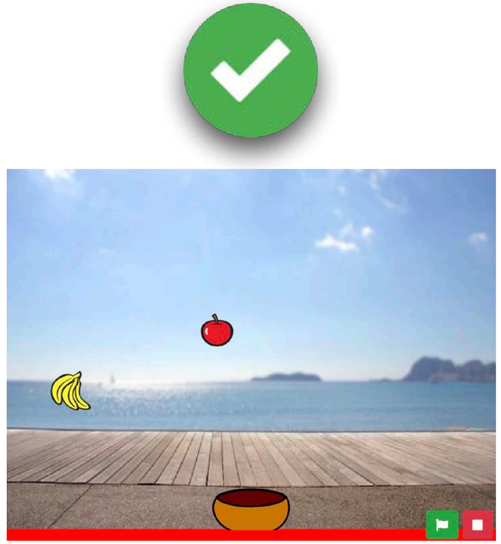


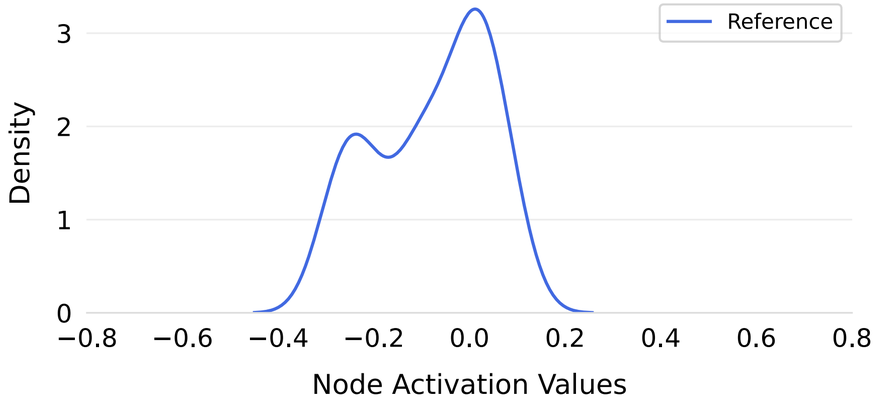
Learn to play

Validate behaviour

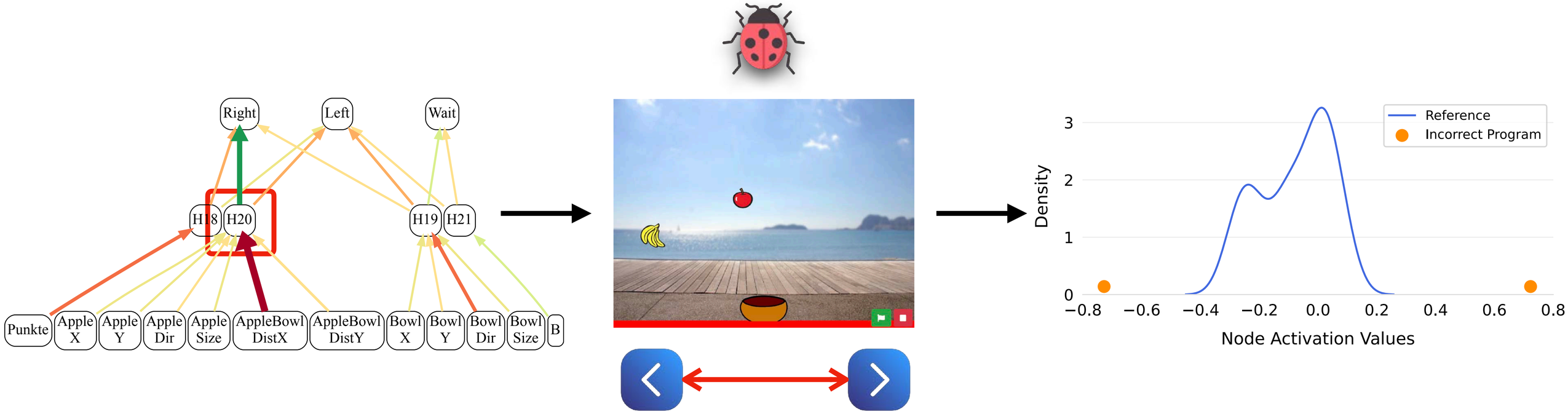
Test Oracle Based on Surprise Adequacy



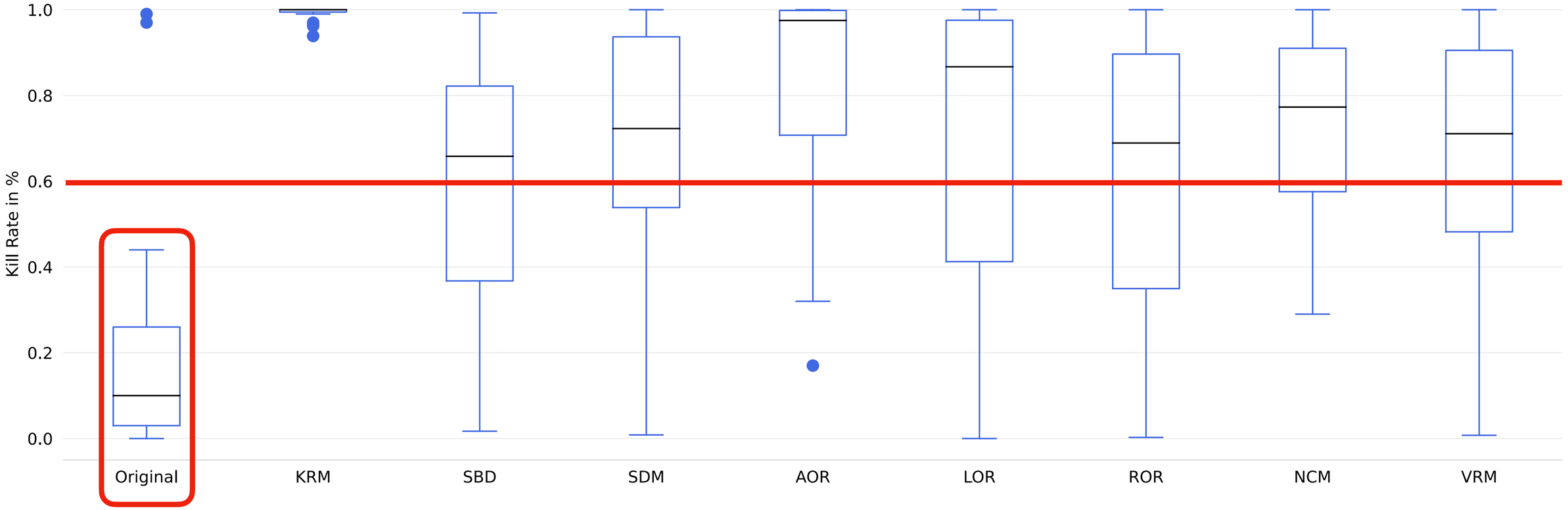




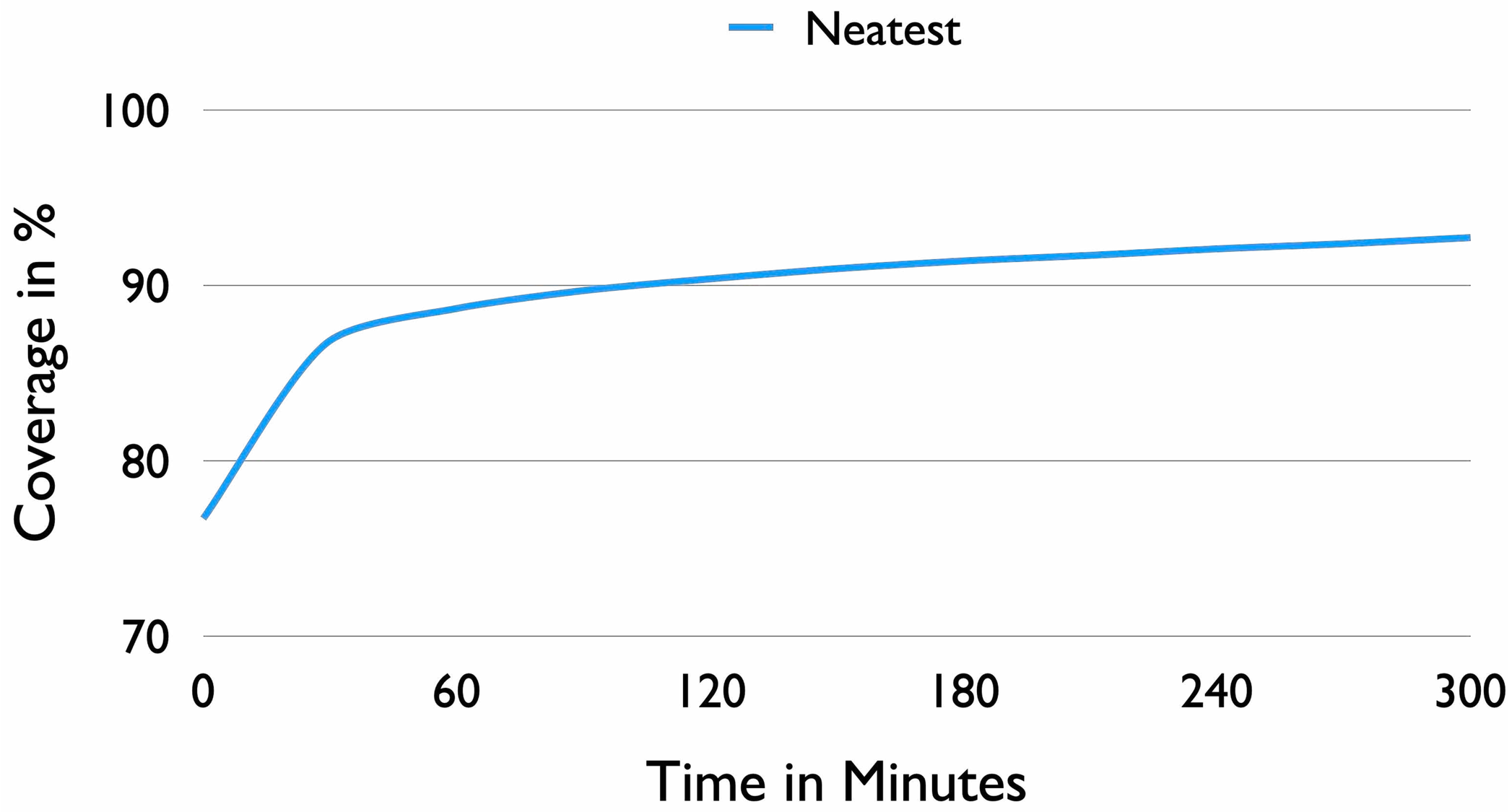
Test Oracle Based on Surprise Adequacy



Surprise-Adequacy Oracle Detects Faults Reliably

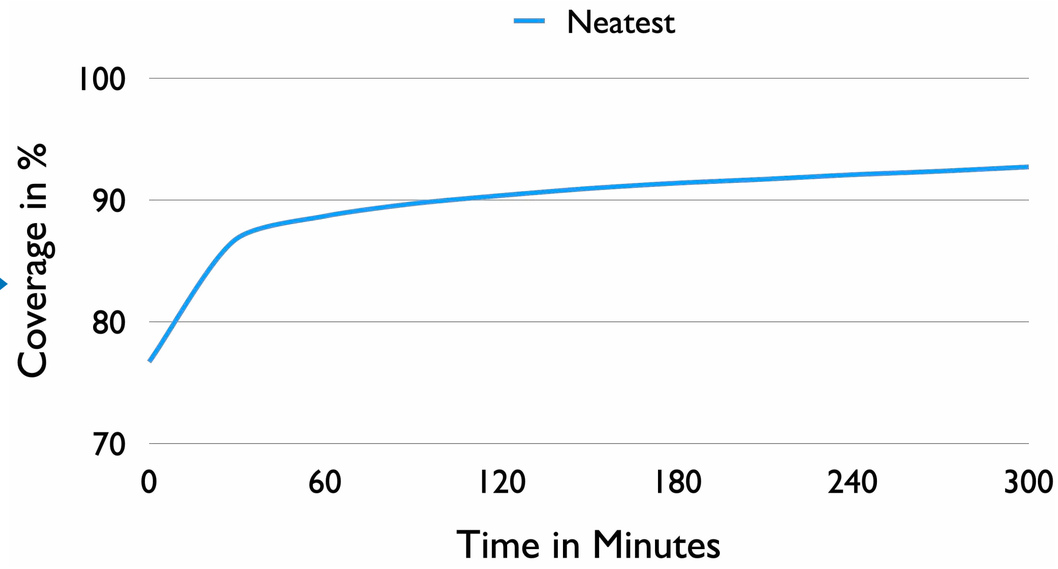


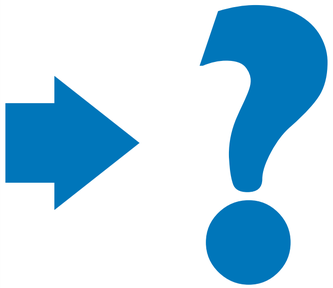
Slow Progress of Neatest

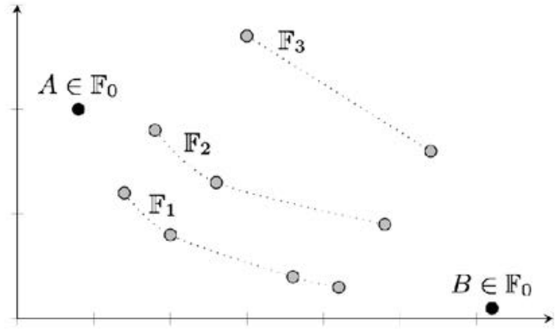


Ongoing Work: Overcoming Limitations of Neatest

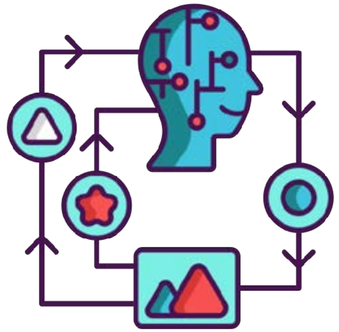
Neatest



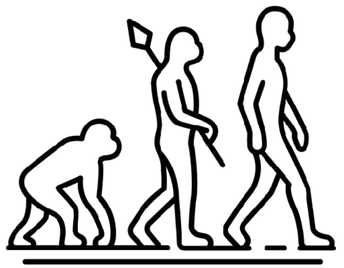




Search-based Testing

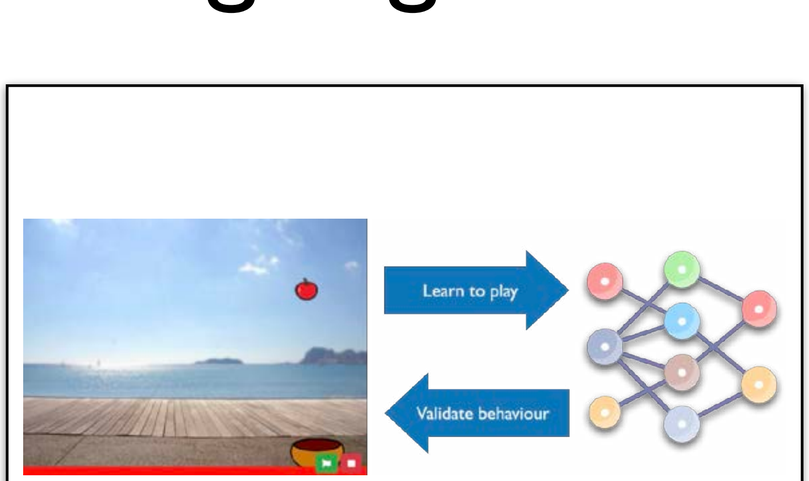


Reinforcement Learning

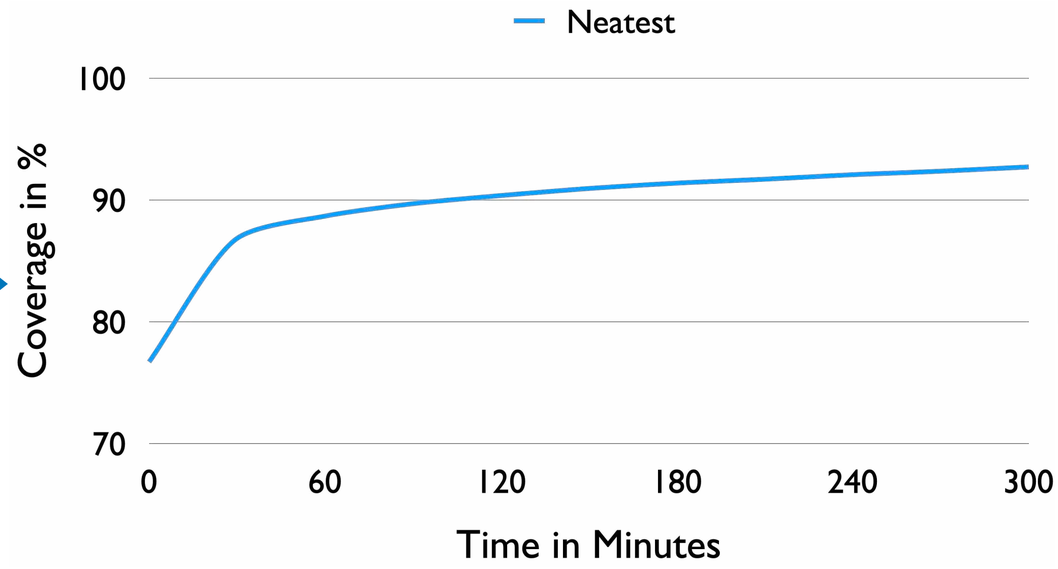


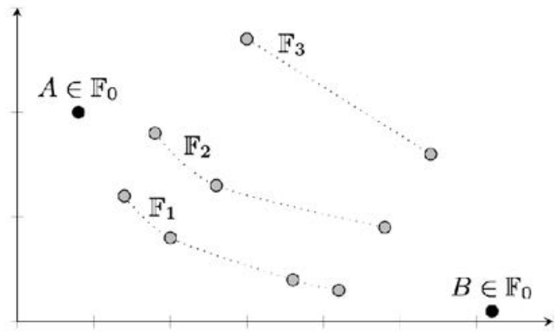
Illumination Search

Ongoing Work: Overcoming Limitations of Neatest



Neatest



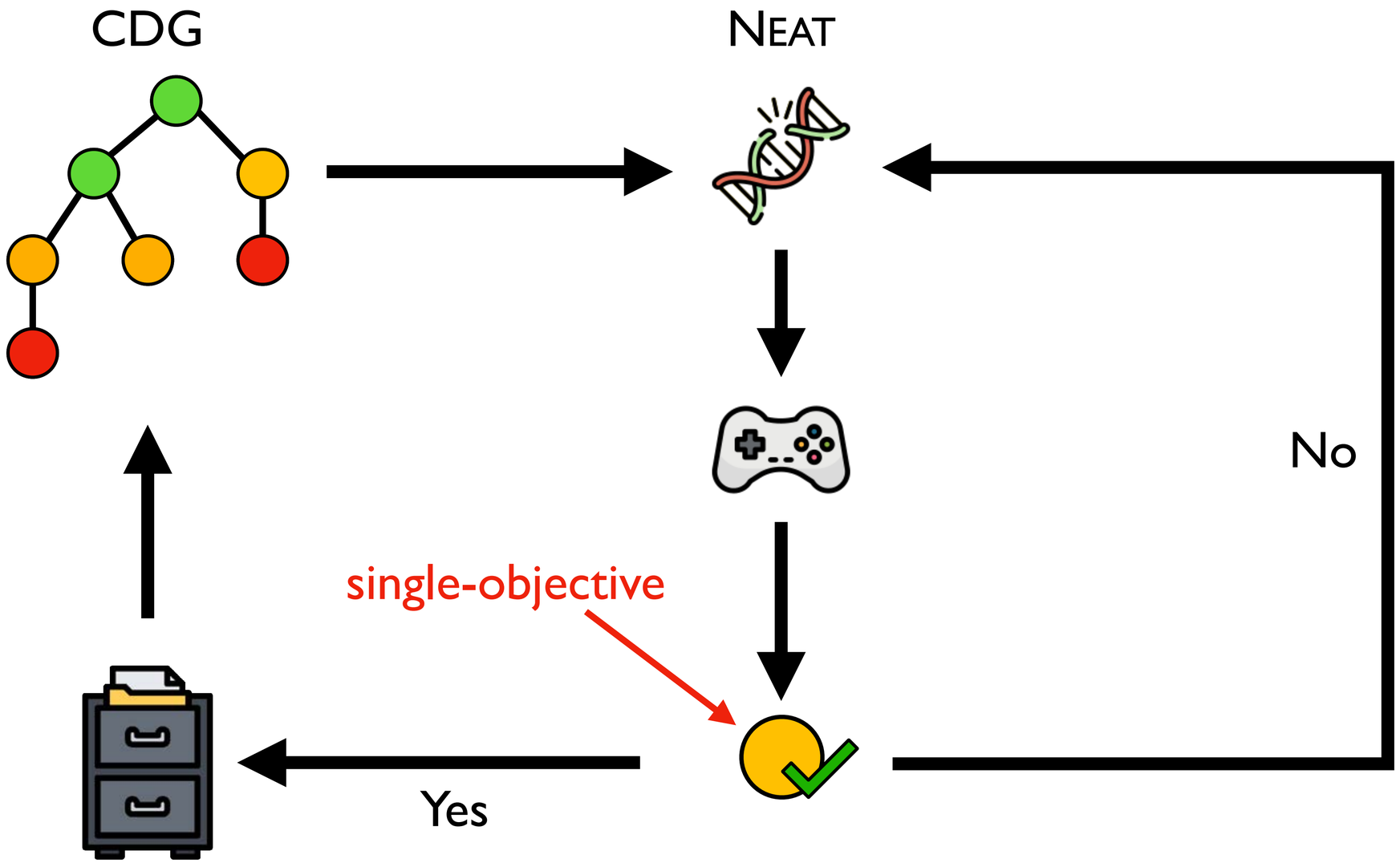


Search-based Testing

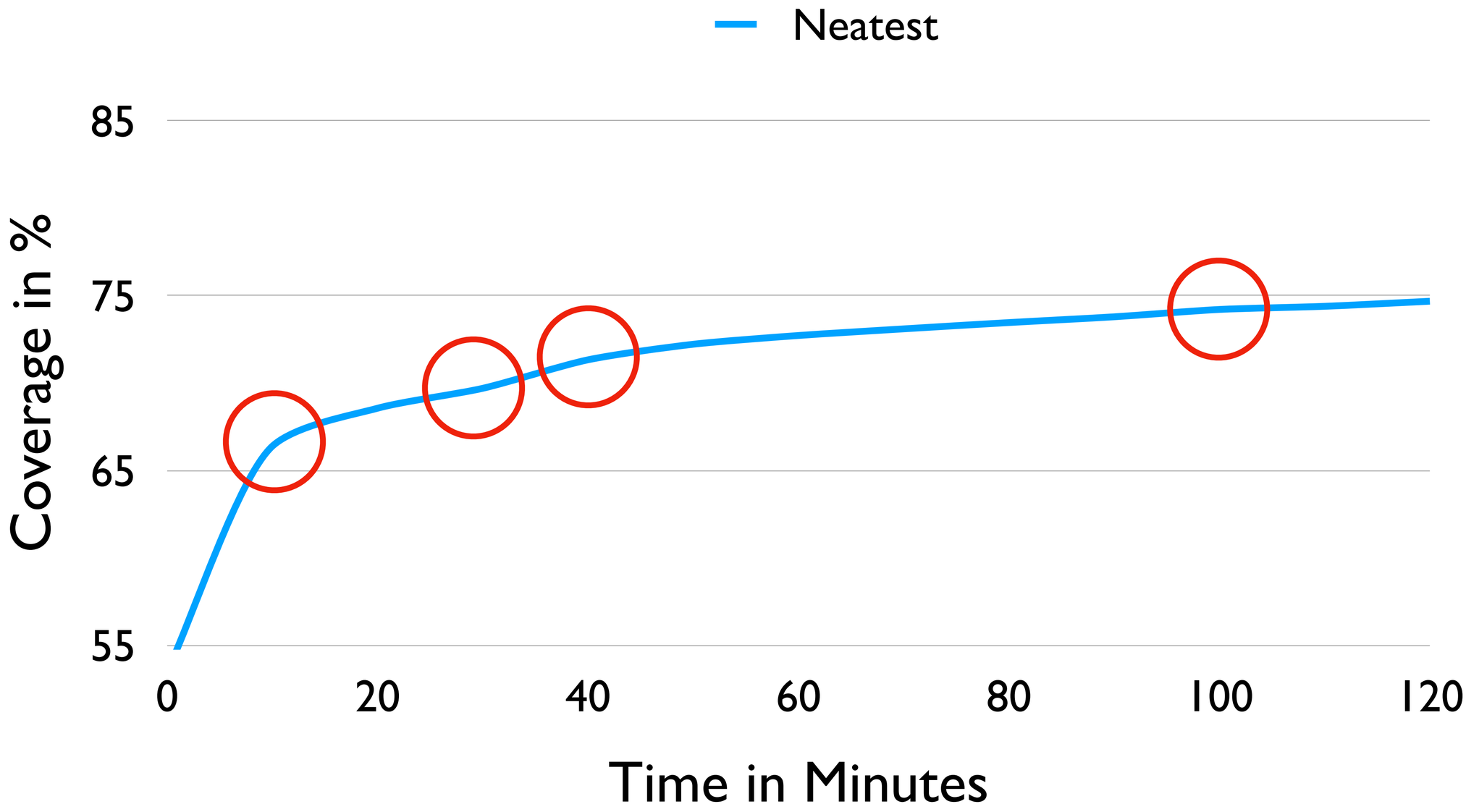
Reinforcement Learning

Illumination Search

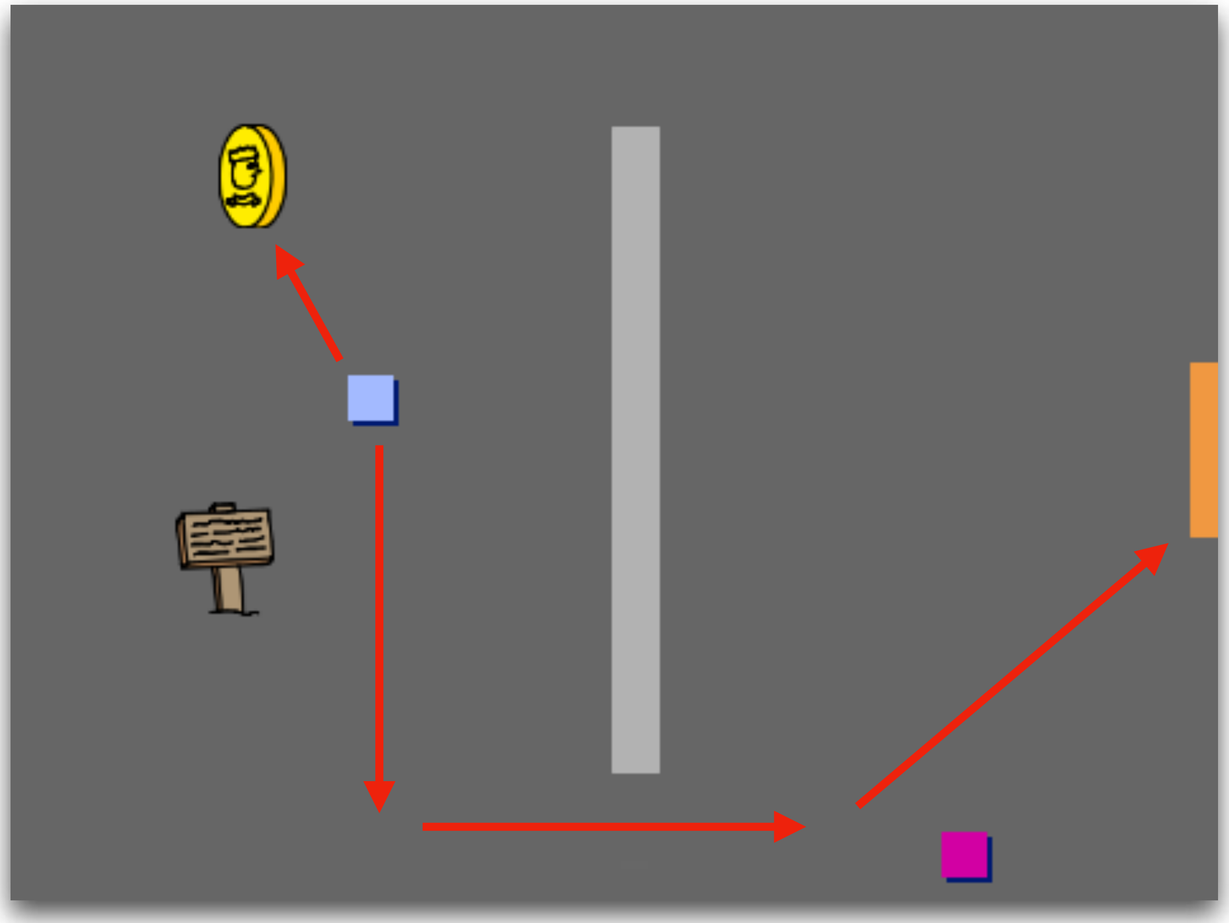
Neatest is a Single-Objective Algorithm



Single-Objective Algorithms are Inefficient



Single-Objective Algorithms are Inefficient



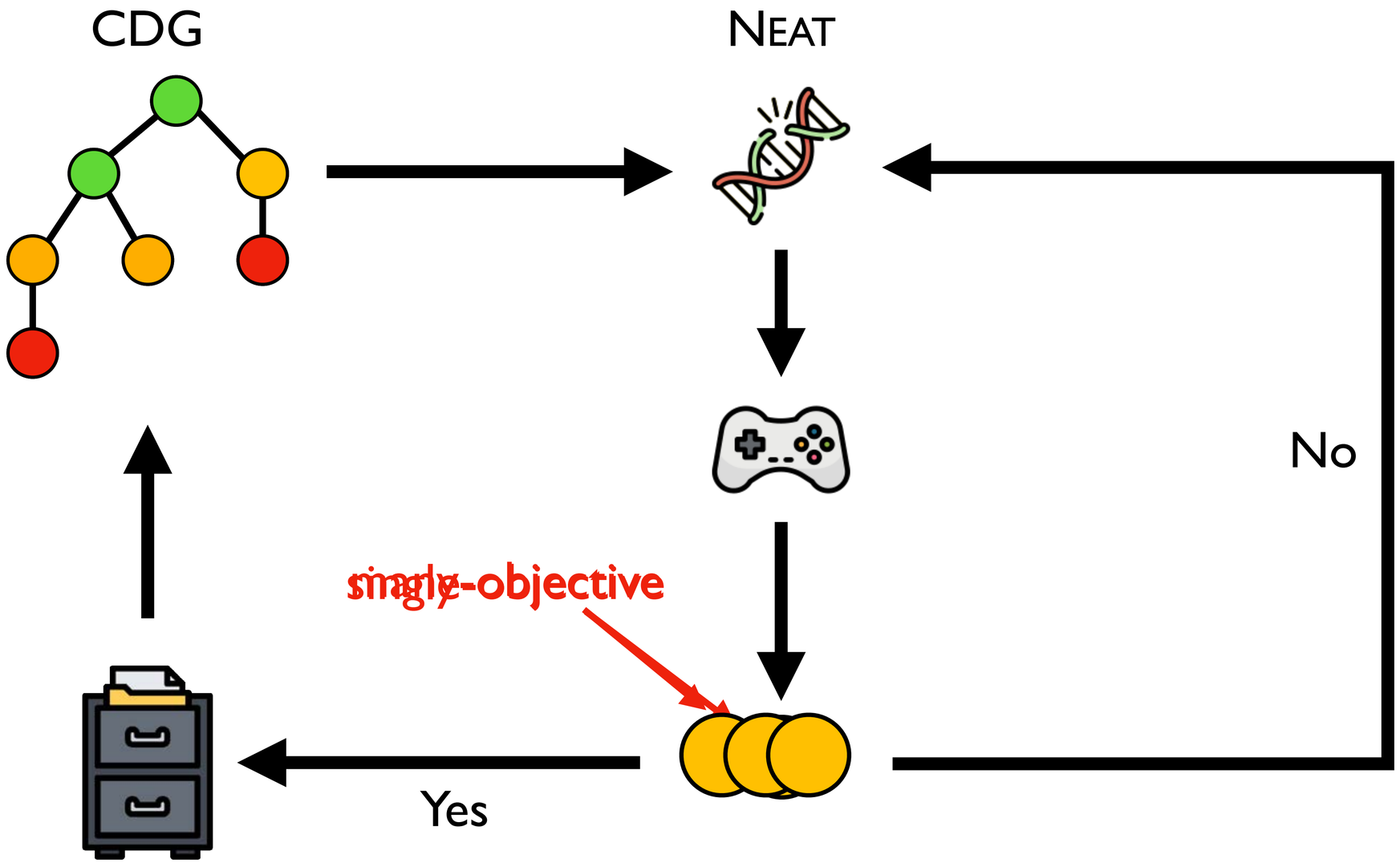
Script: Next Level

```
// …
if touching orange colour:
    // go to next level
```

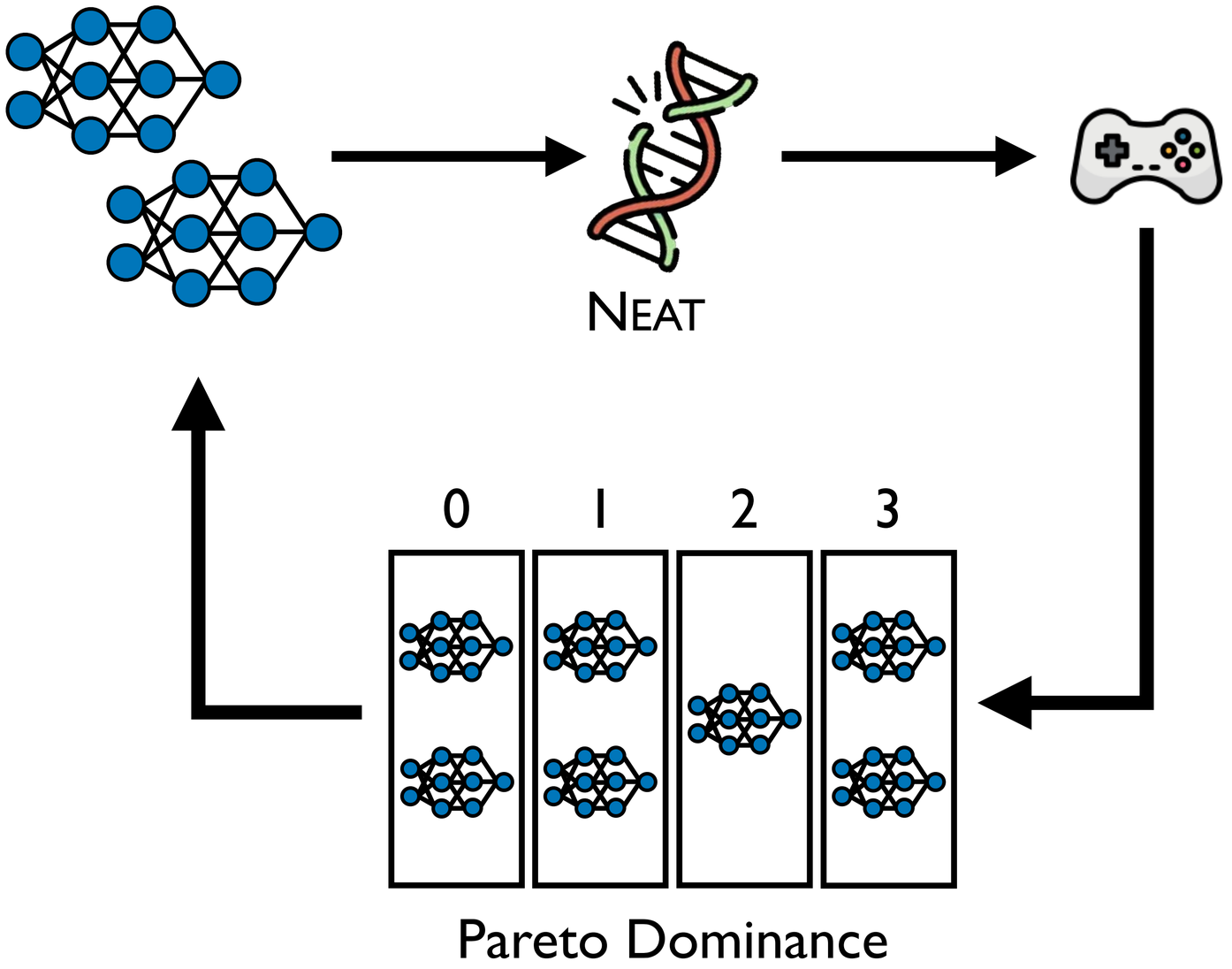
Script: Coin

```
// …
if touching coin:
    // increase score
```

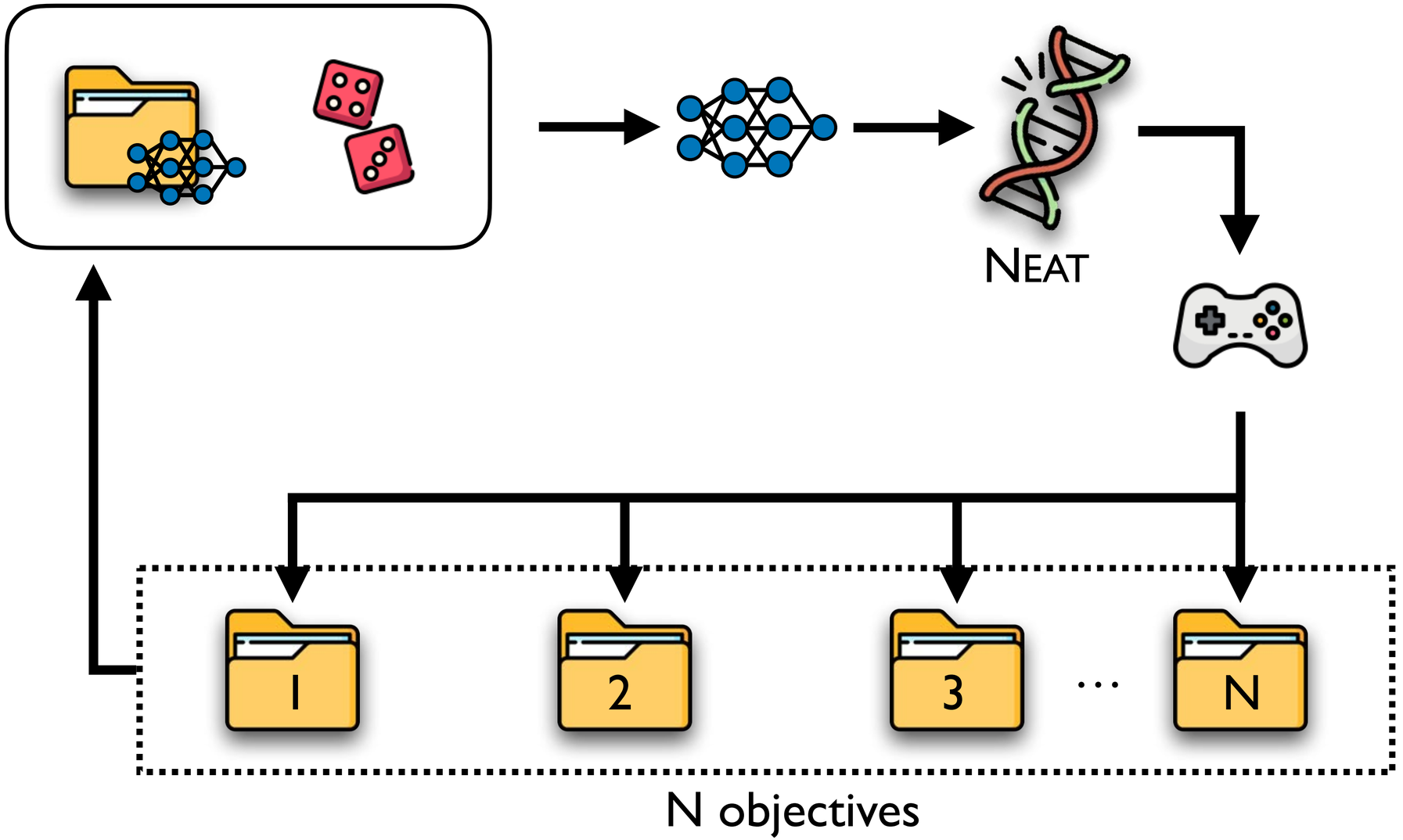
Cast Neatest to a Many-Objective Algorithm



MOSA-Neatest

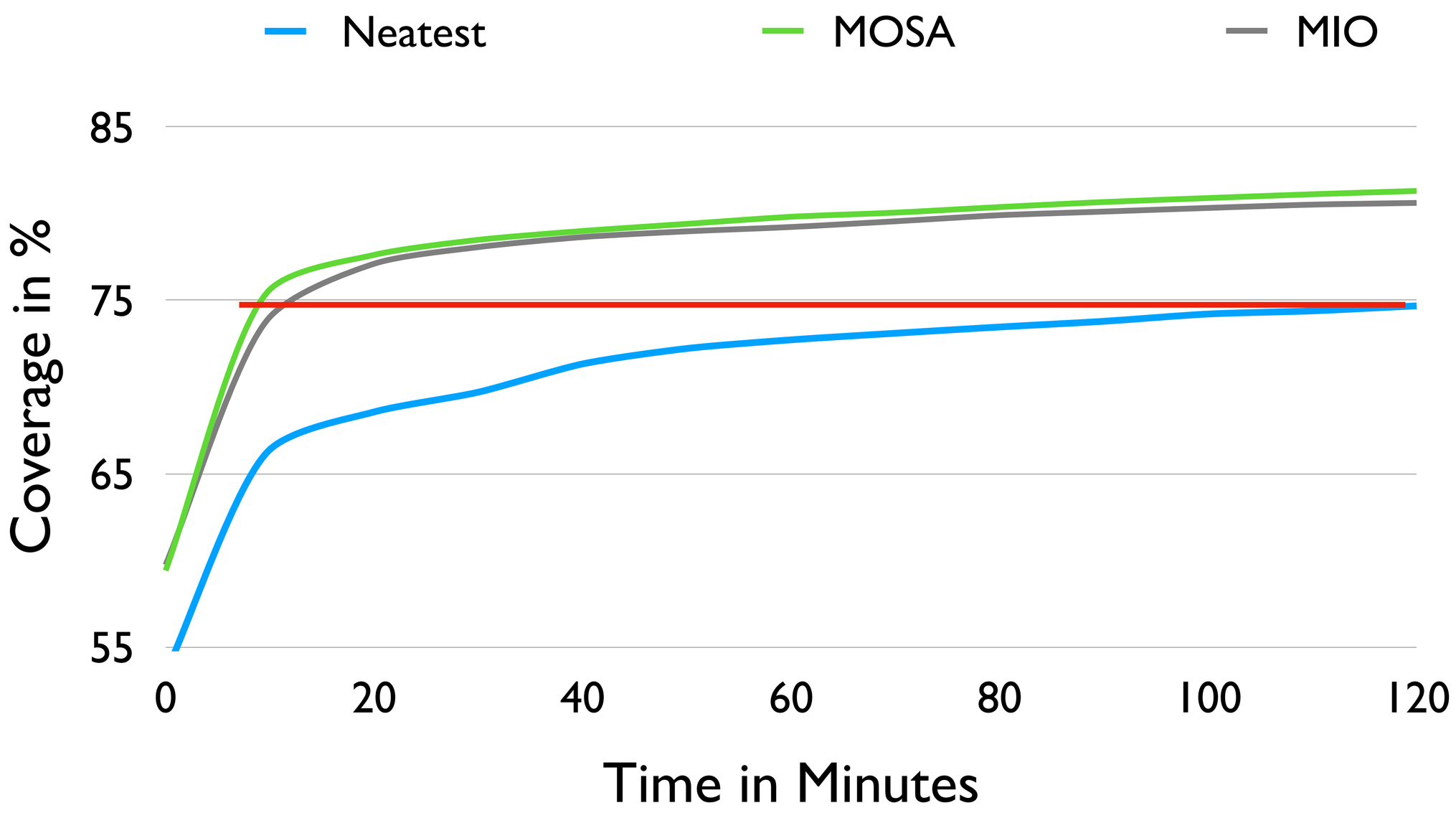


MIO-Neatest



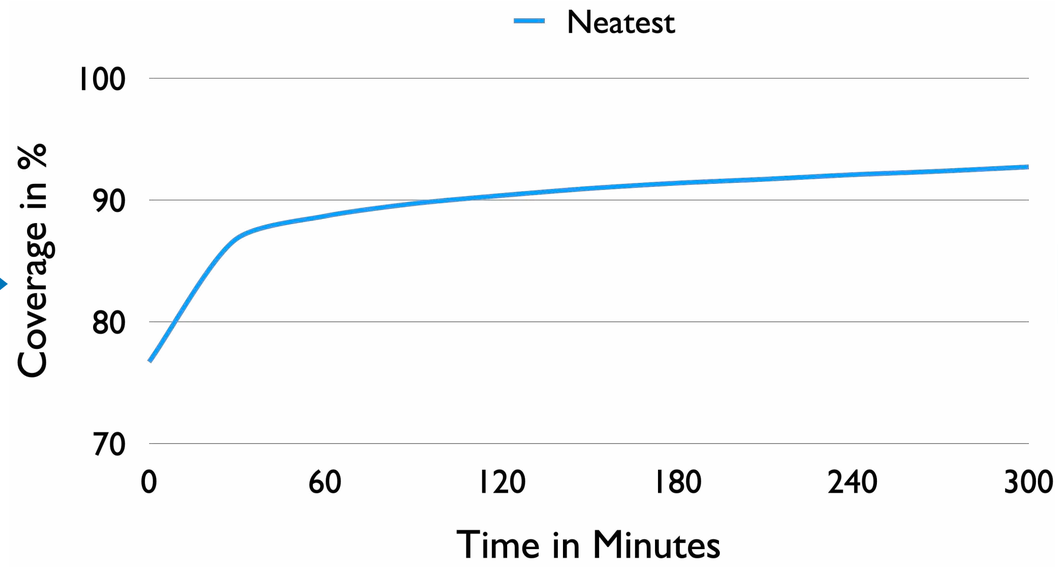
Evaluation of MO-Neatest on 20 Games

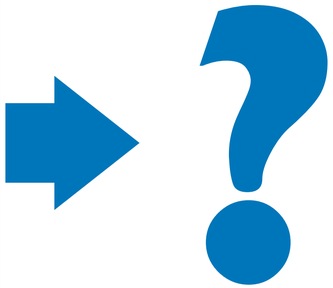
MO-Search Improves Neatest

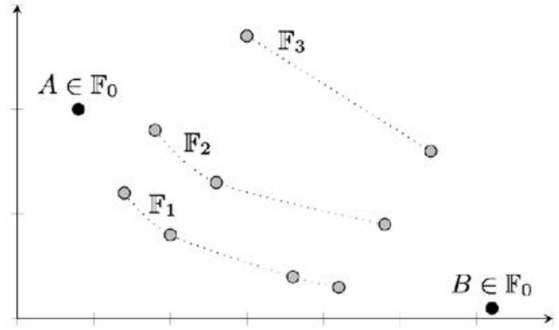


Ongoing Work: Overcoming Limitations of Neatest

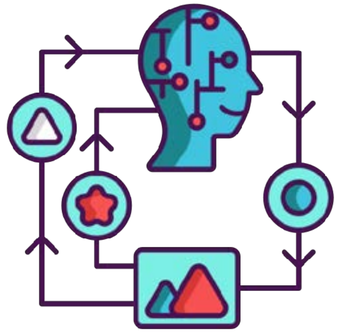
Neatest



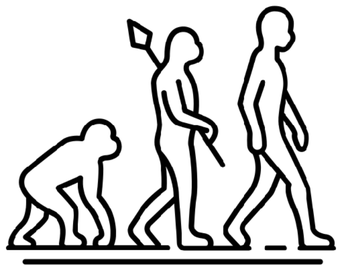




Search-based Testing

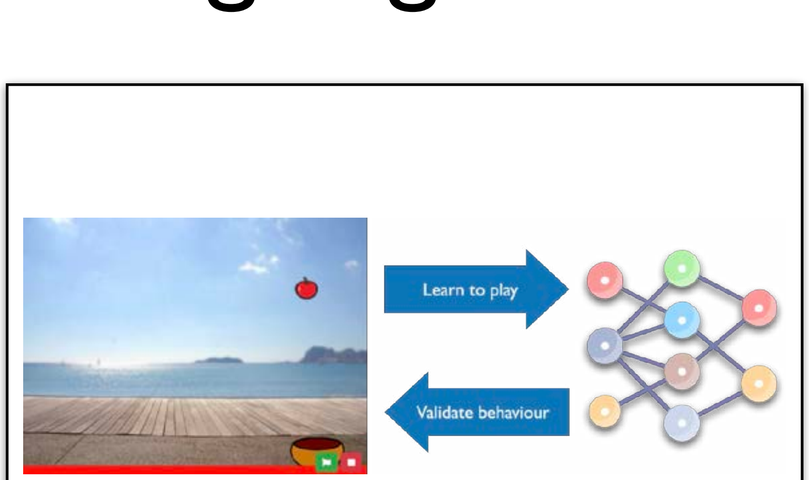


Reinforcement Learning

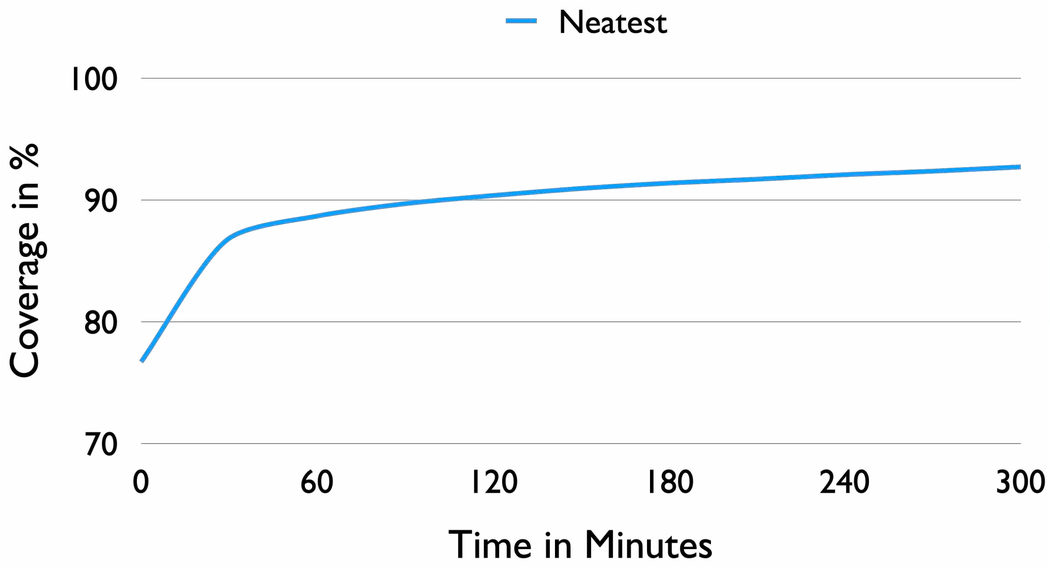


Illumination Search

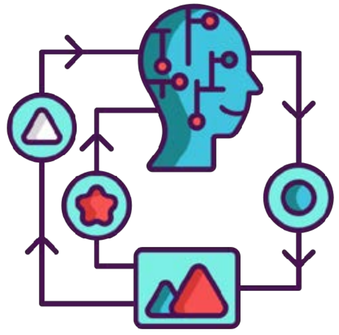
Ongoing Work: Overcoming Limitations of Neatest



Neatest



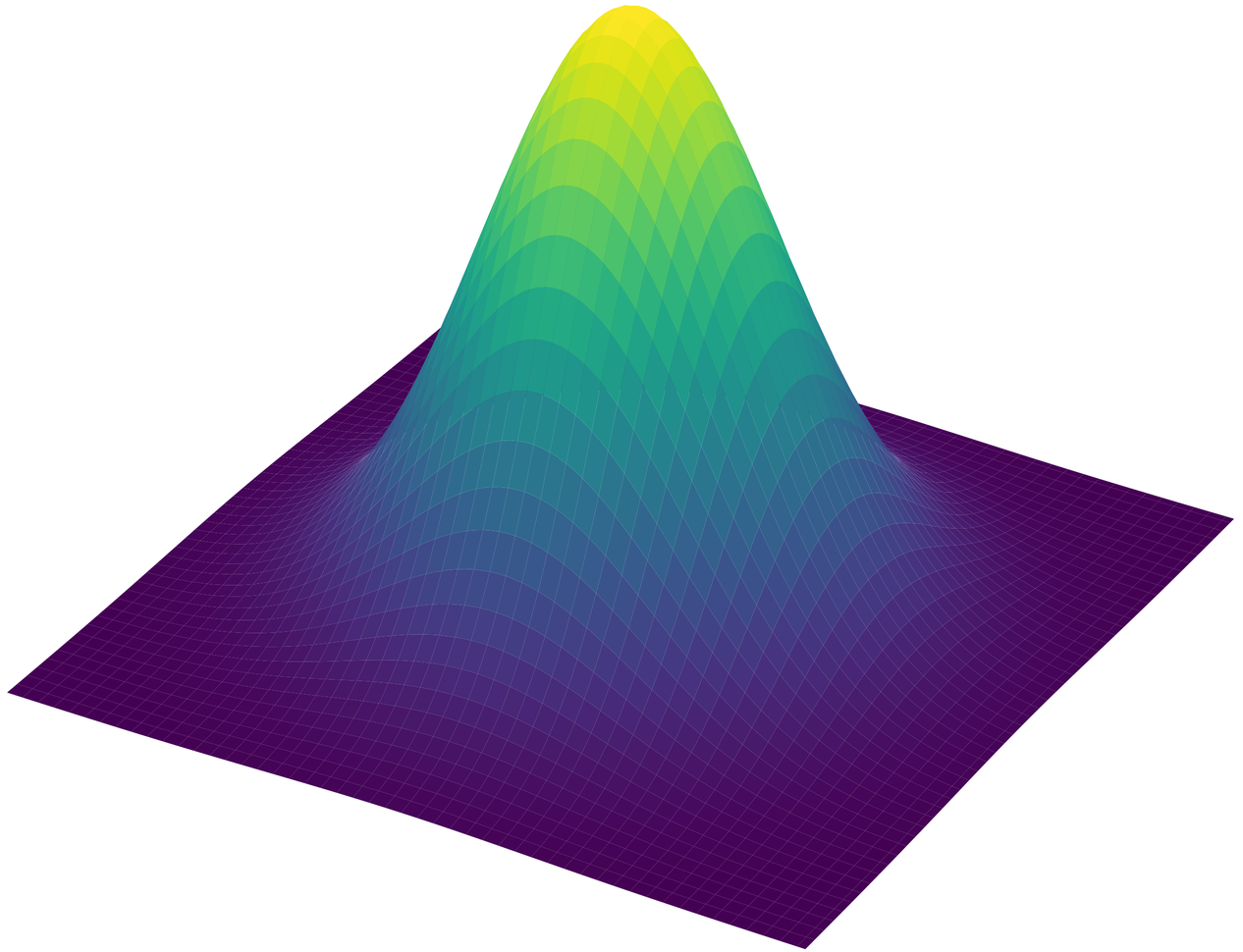
Search-based Testing



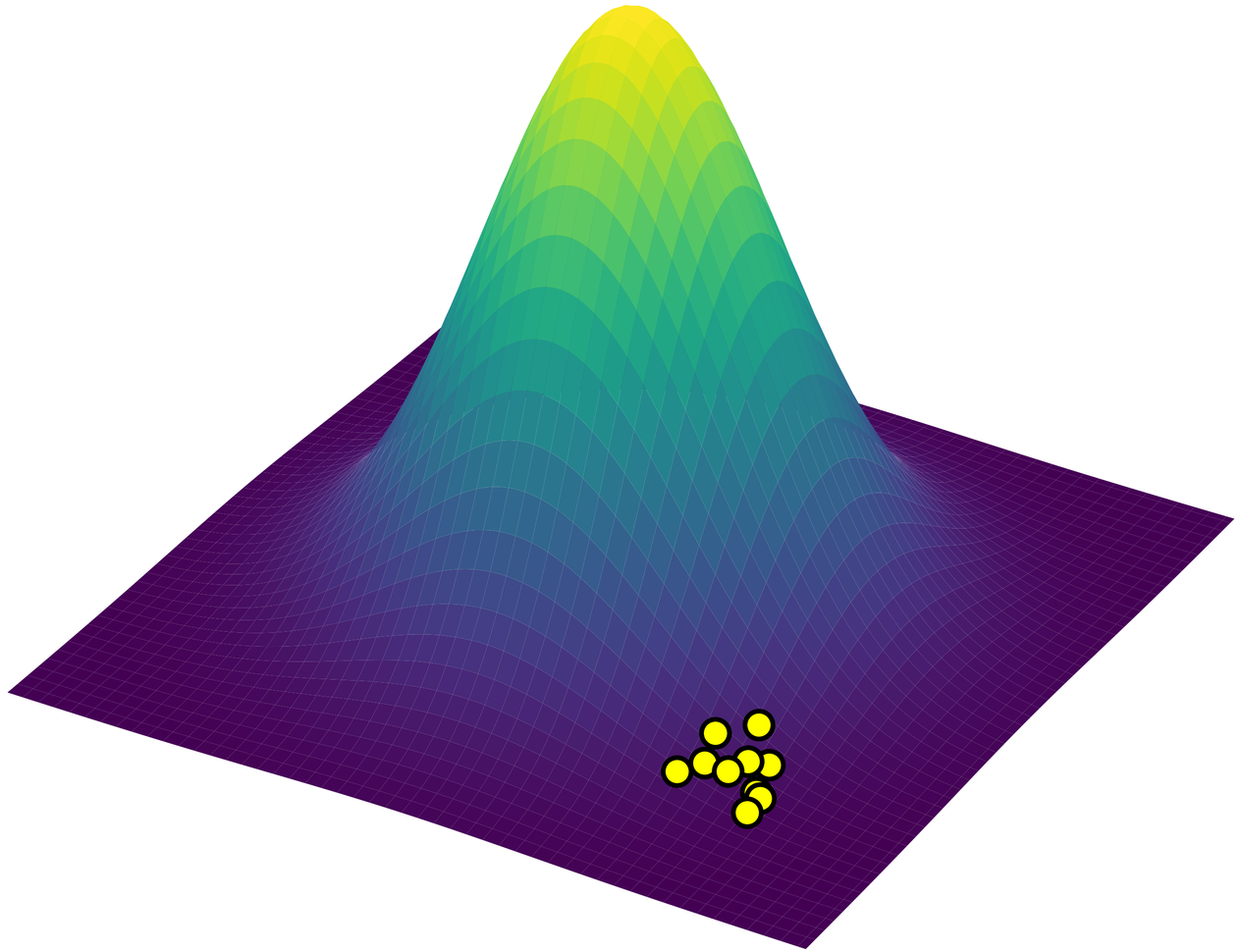
Reinforcement Learning

Illumination Search

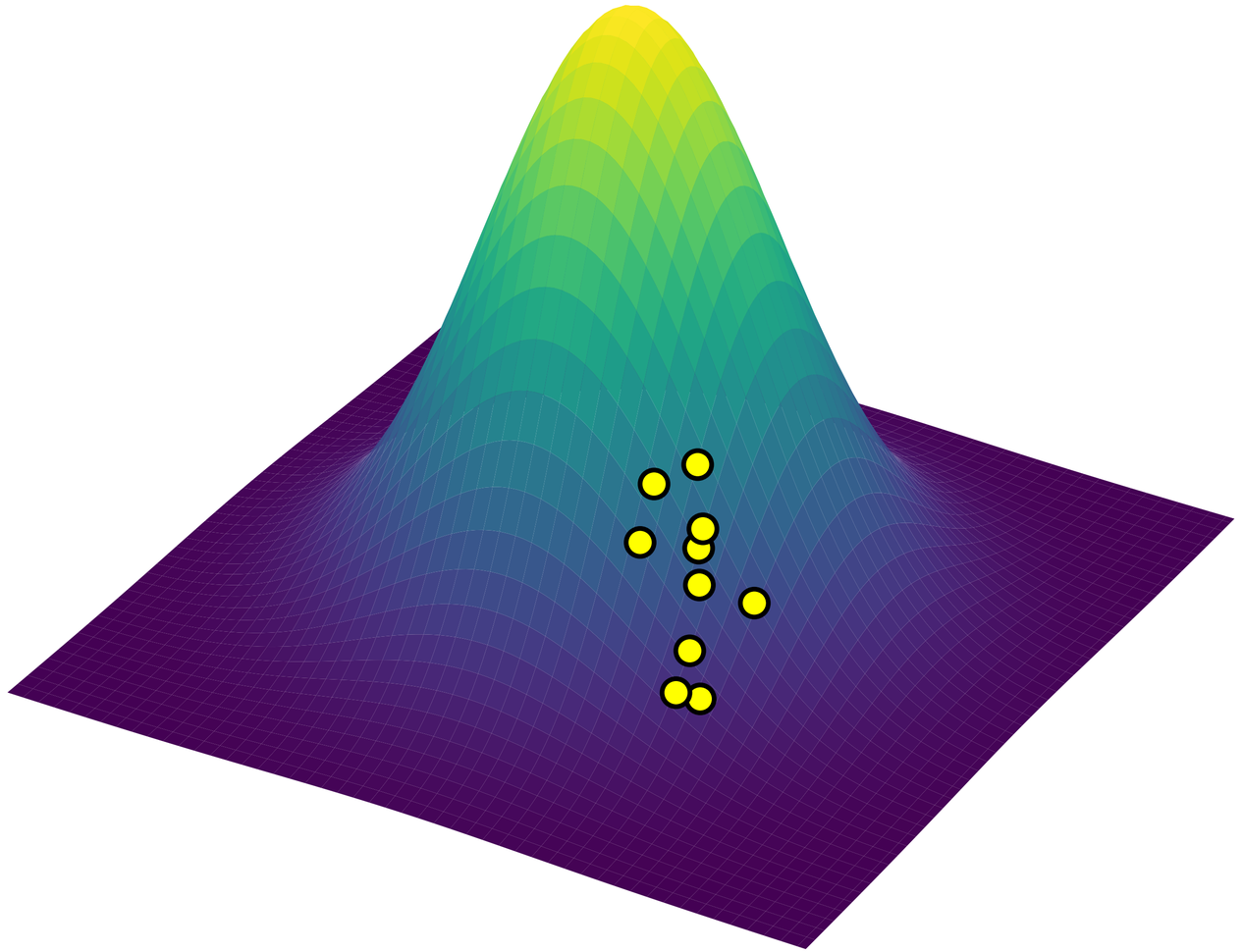
Slow Progress of Stochastic Weight Mutation



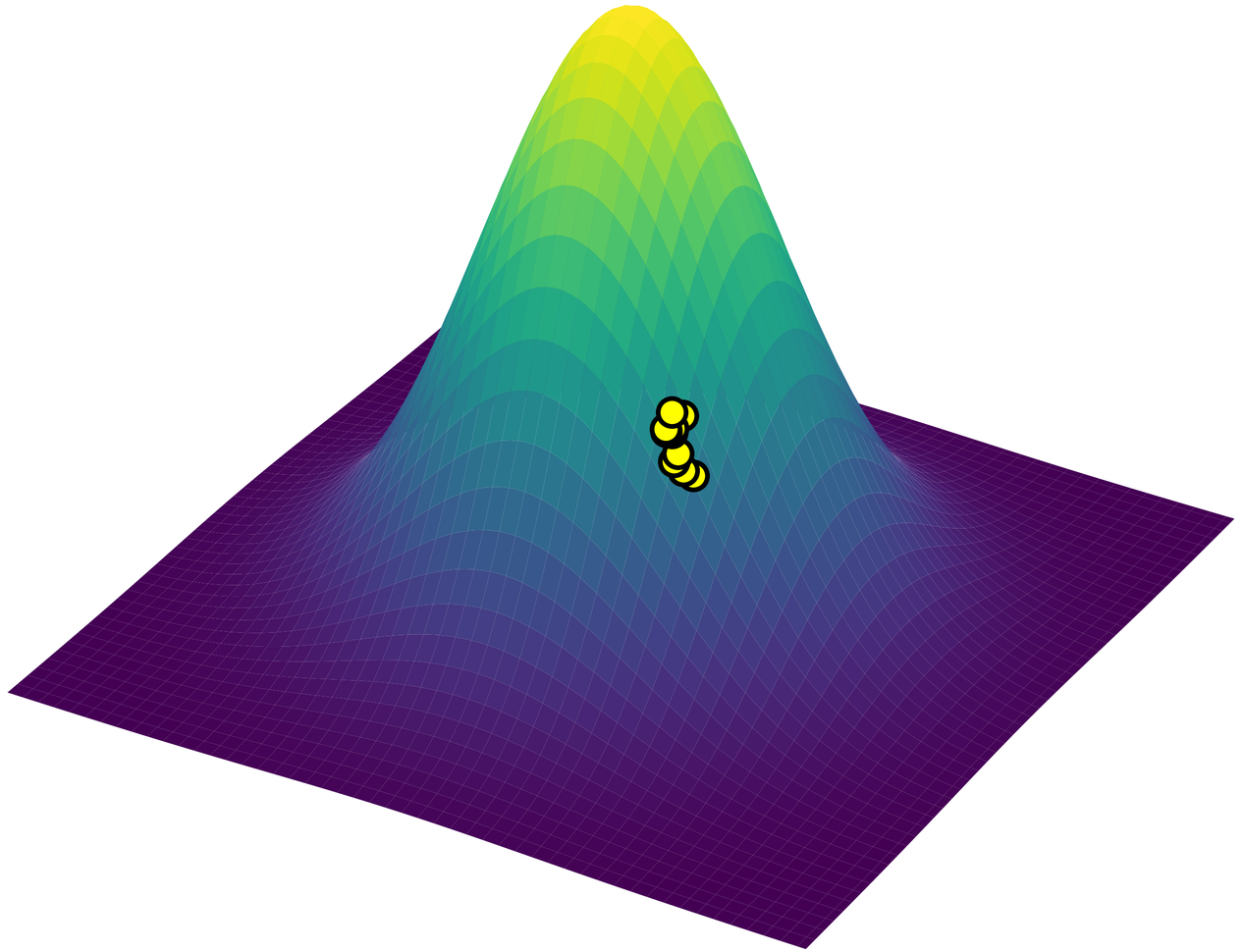
Slow Progress of Stochastic Weight Mutation



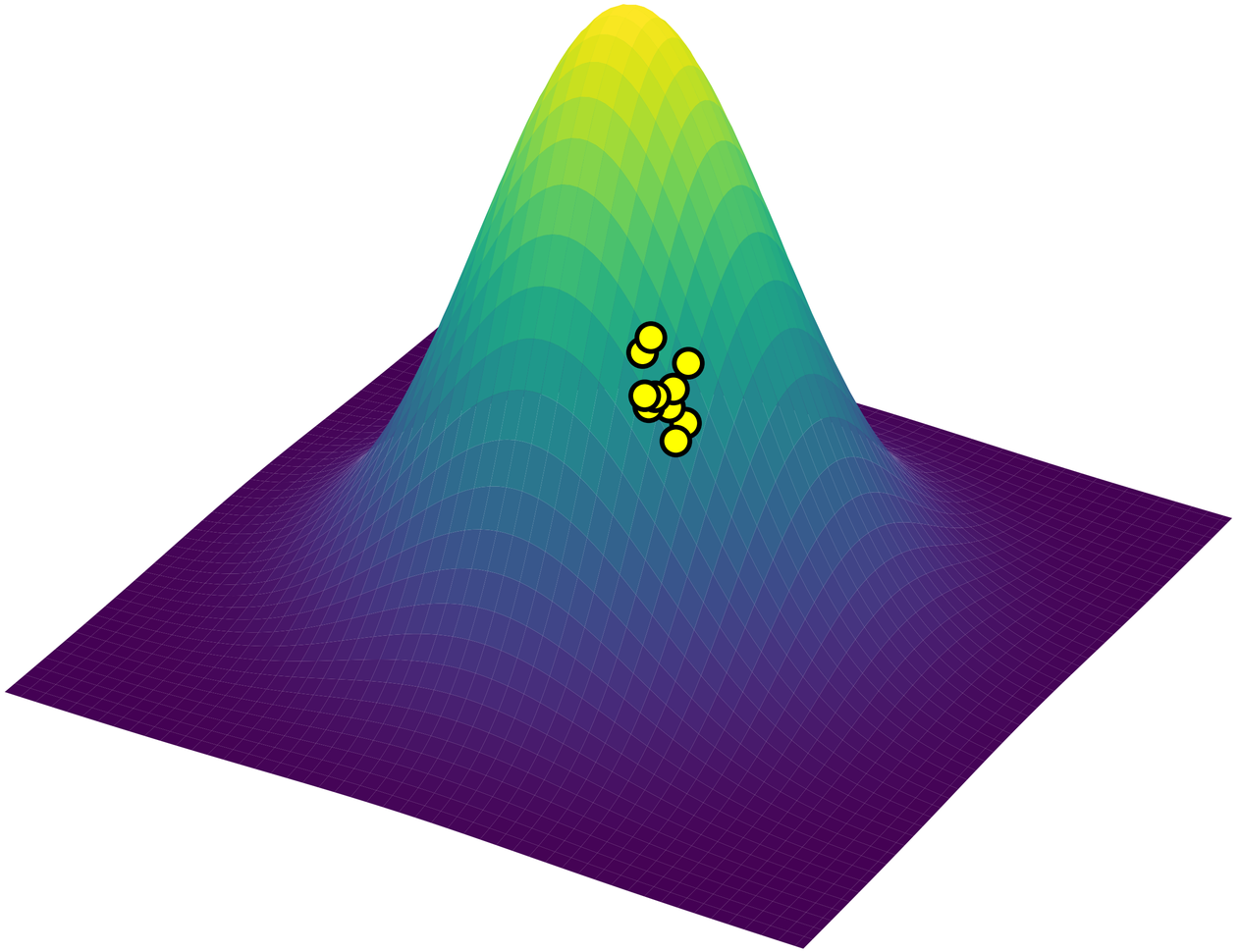
Slow Progress of Stochastic Weight Mutation



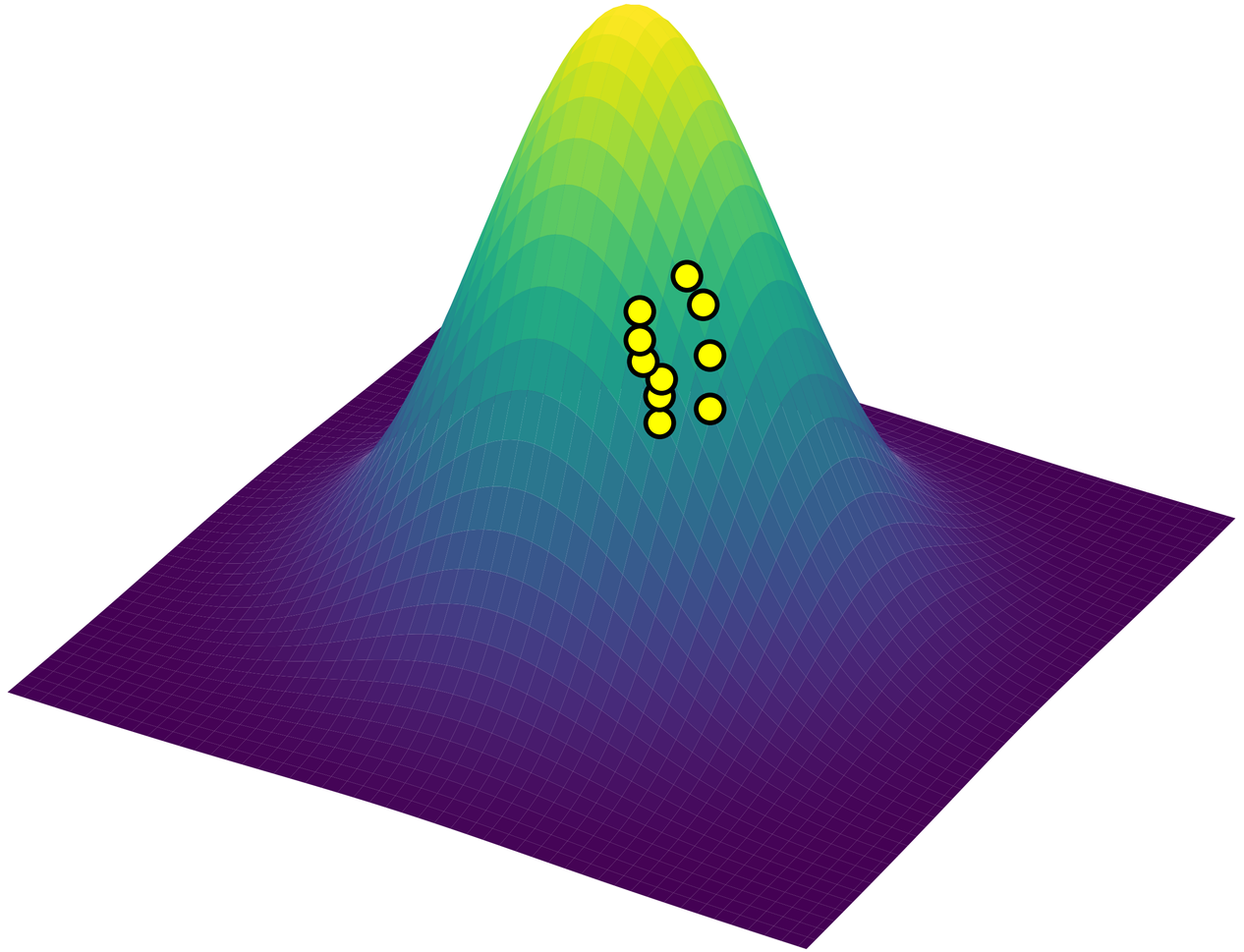
Slow Progress of Stochastic Weight Mutation



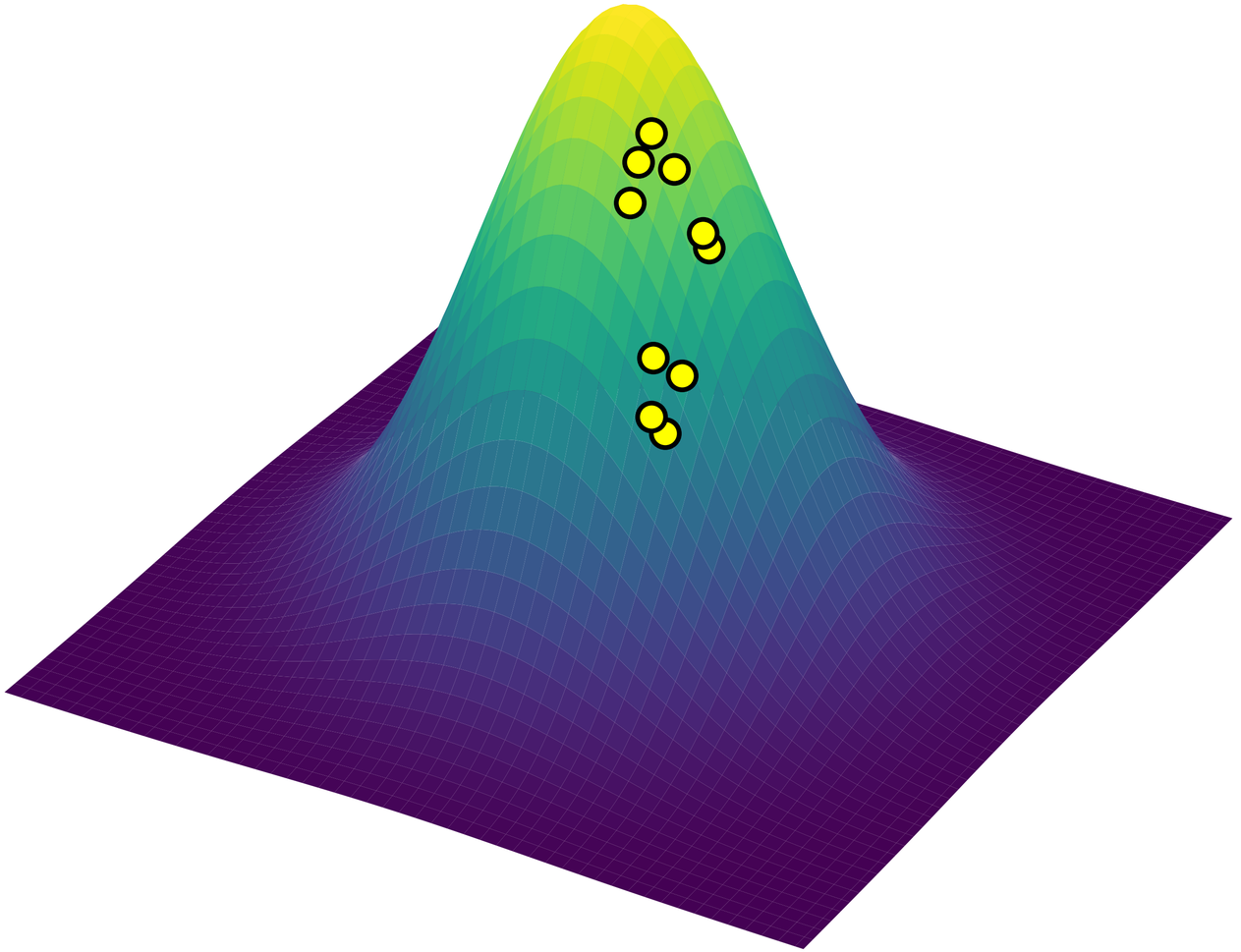
Slow Progress of Stochastic Weight Mutation



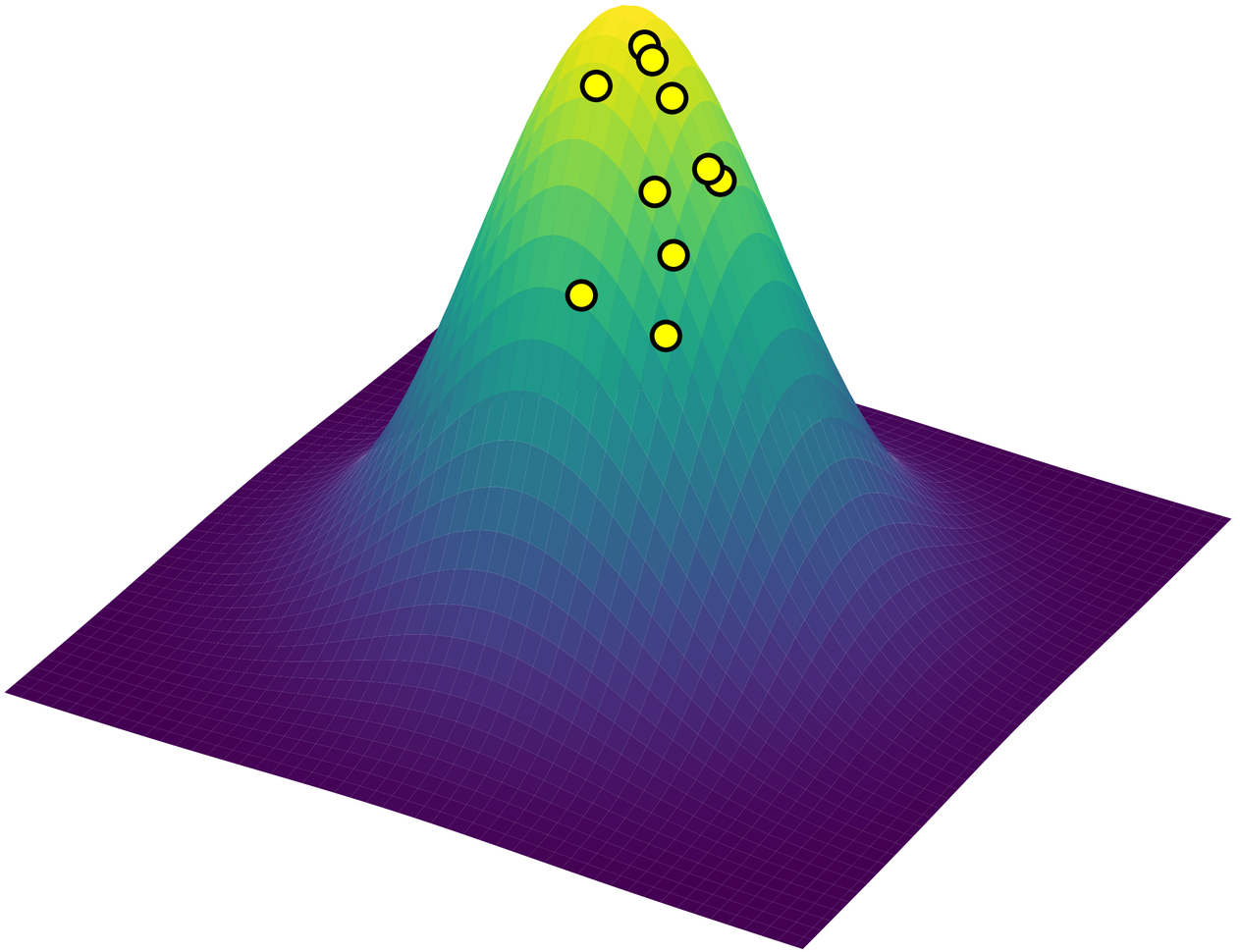
Slow Progress of Stochastic Weight Mutation



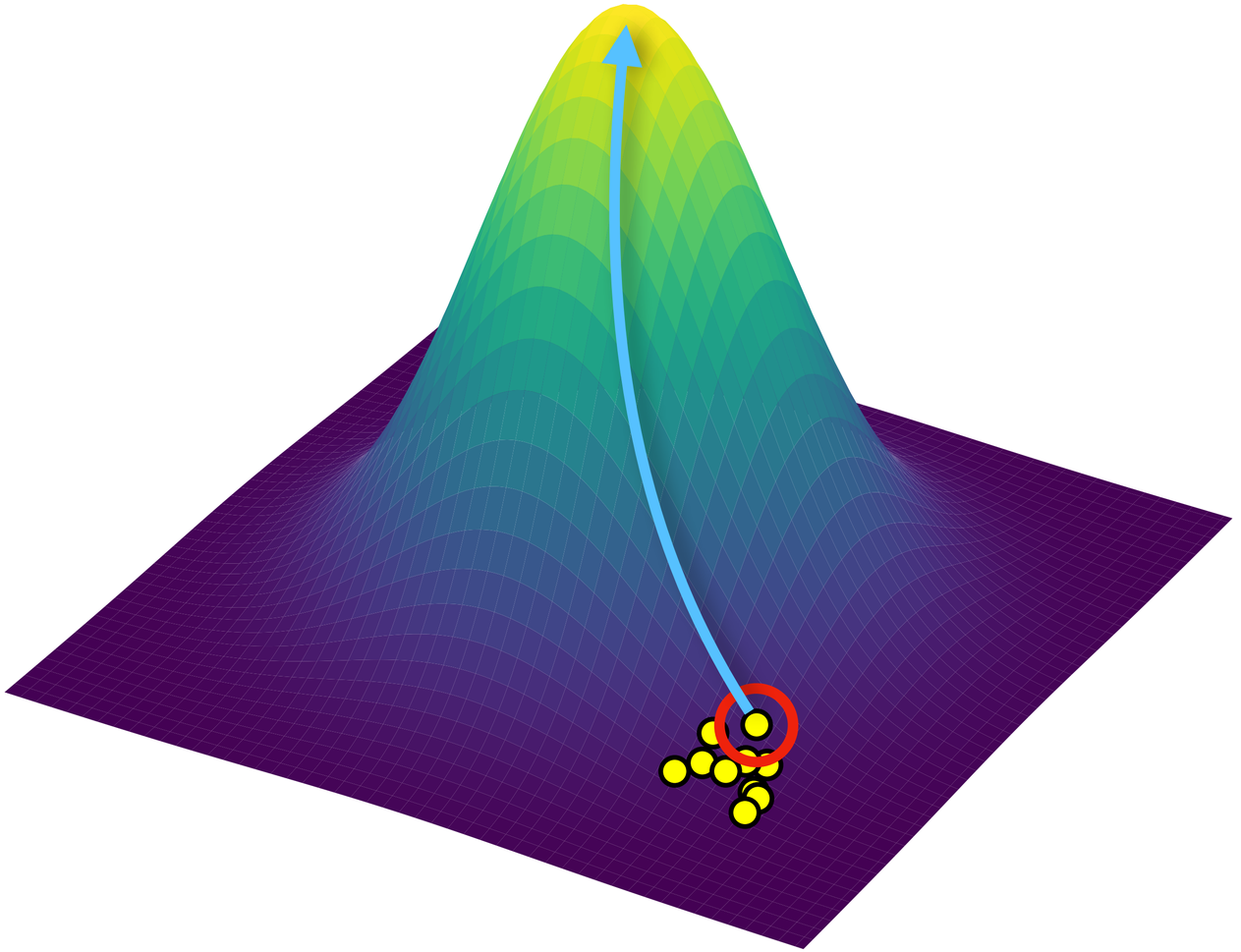
Slow Progress of Stochastic Weight Mutation



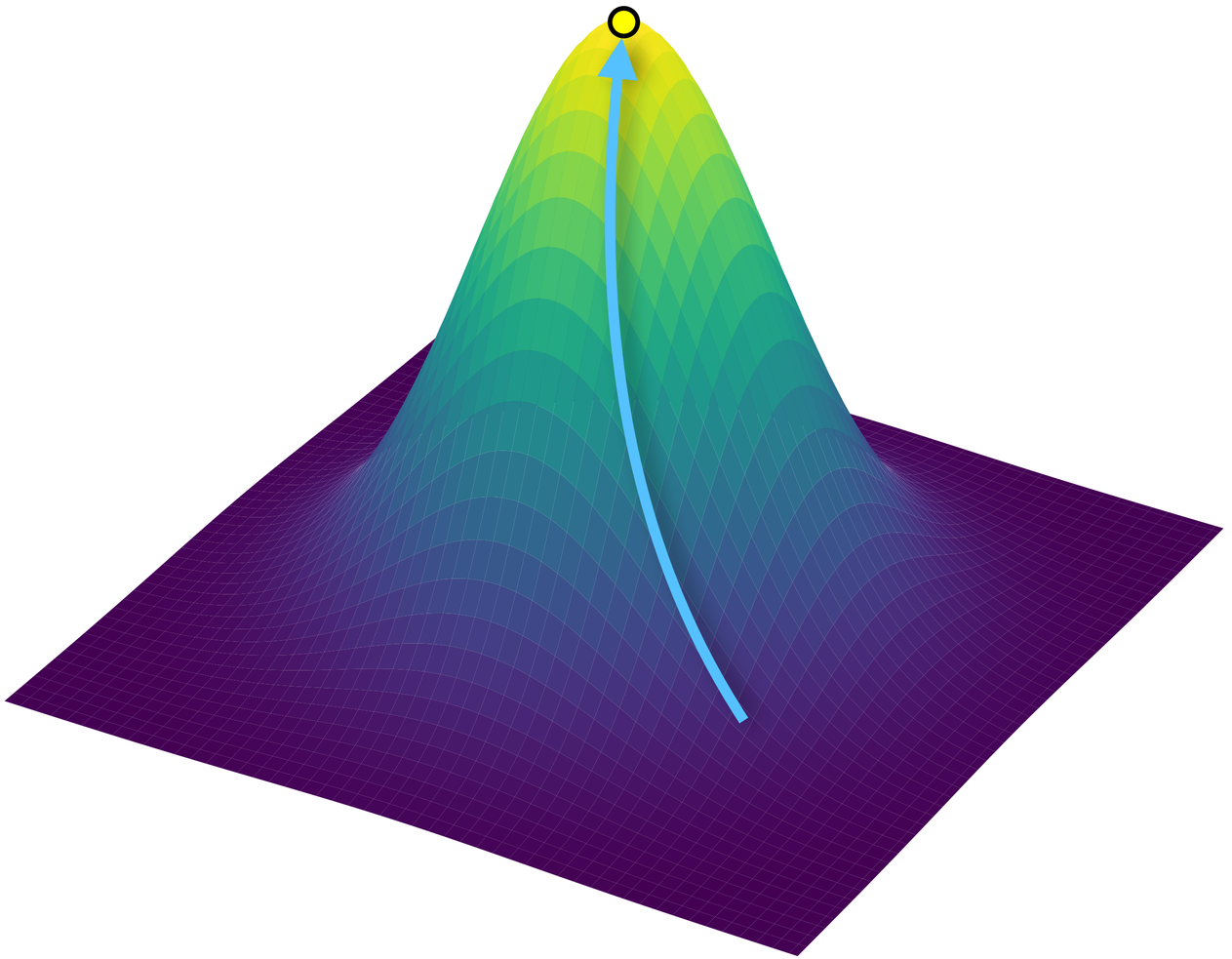
Slow Progress of Stochastic Weight Mutation



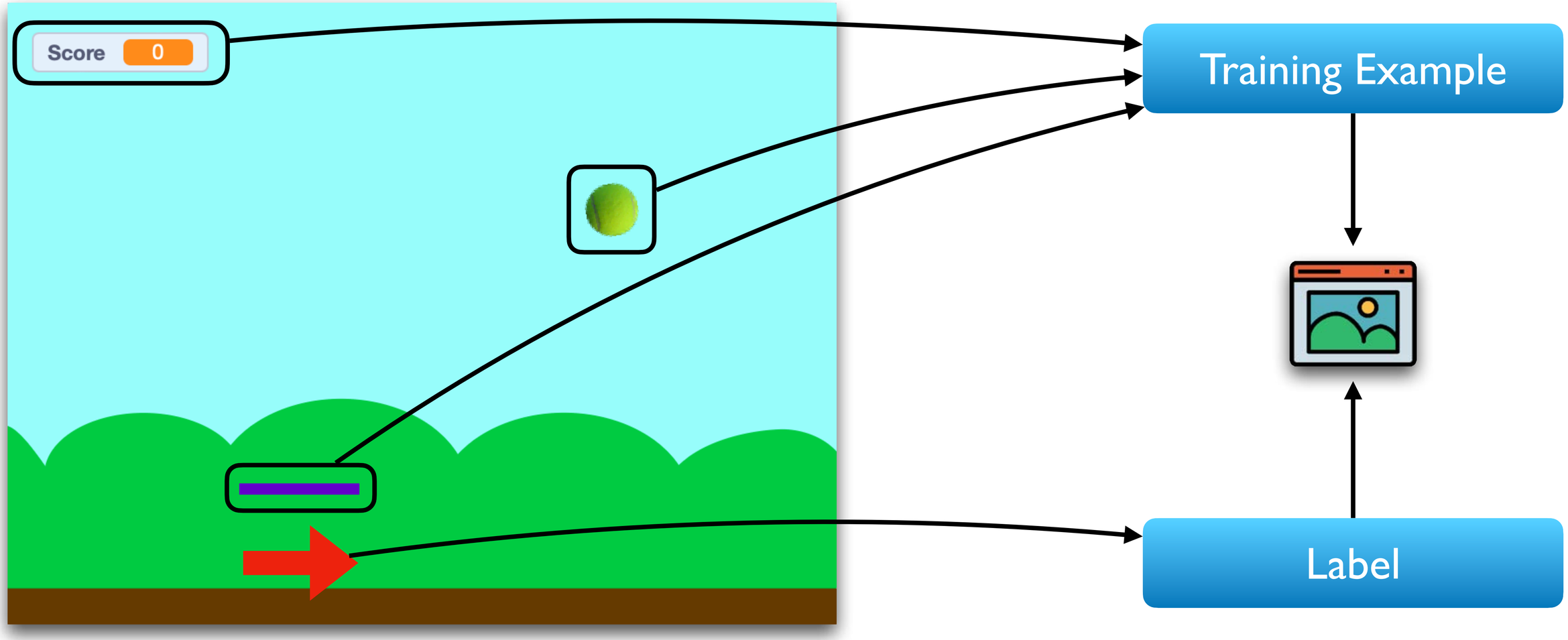
Gradient-Descent as Systematic Optimiser



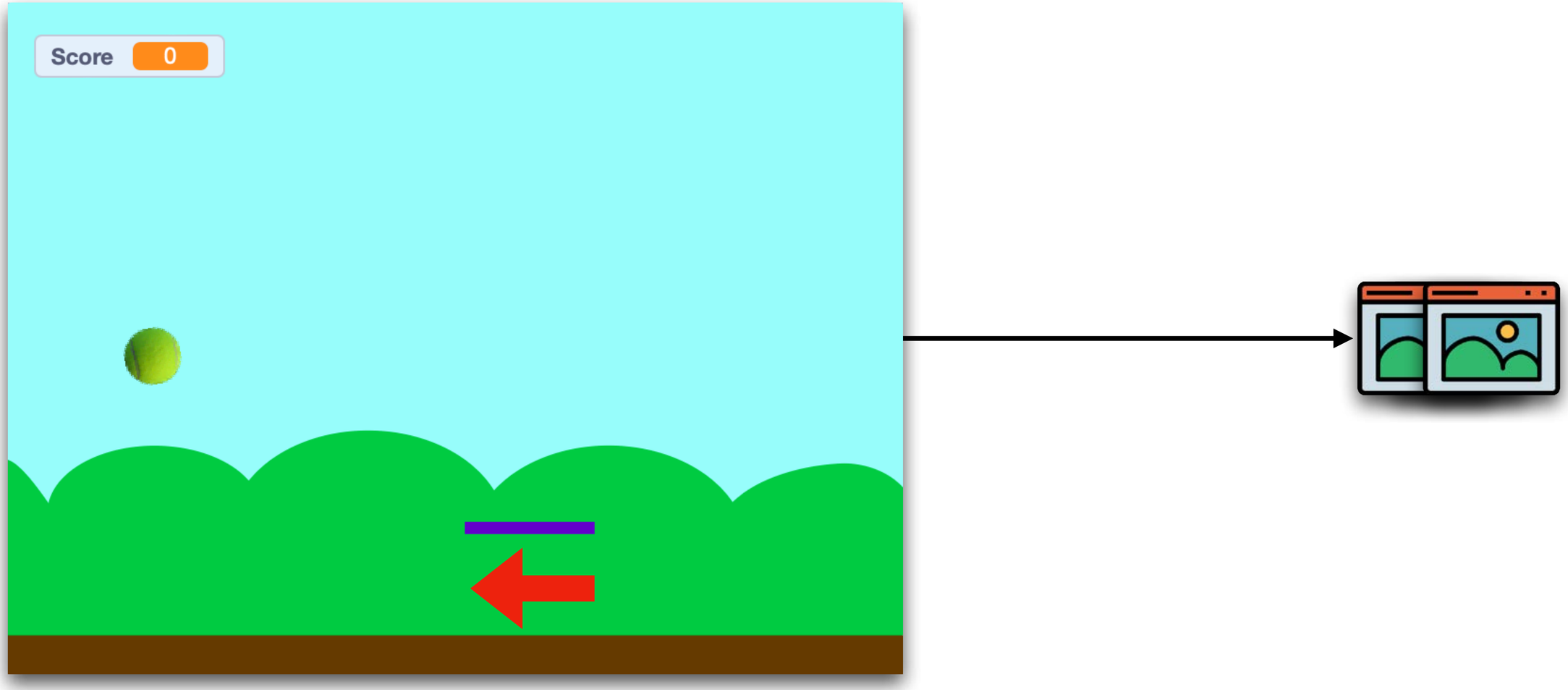
Gradient-Descent as Systematic Optimiser



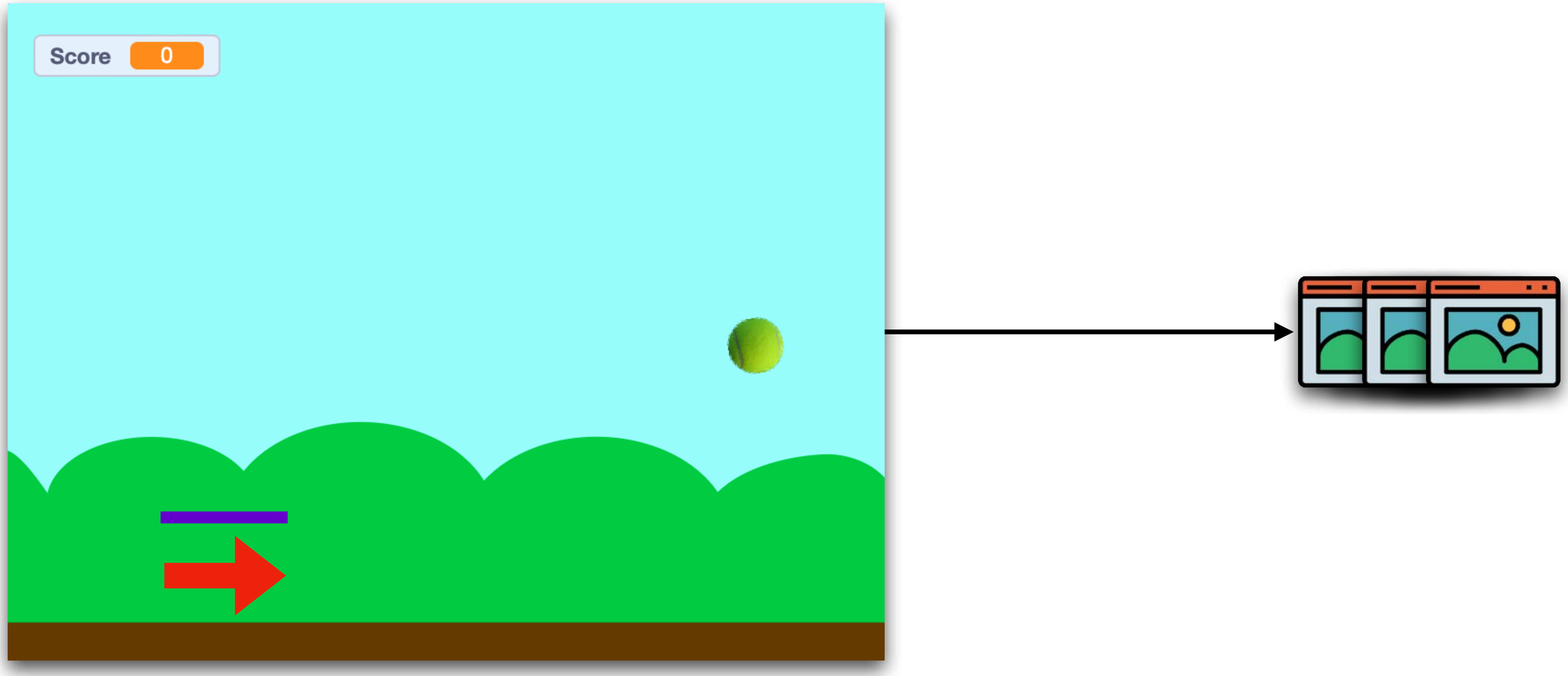
Human Gameplay Traces as Training Set



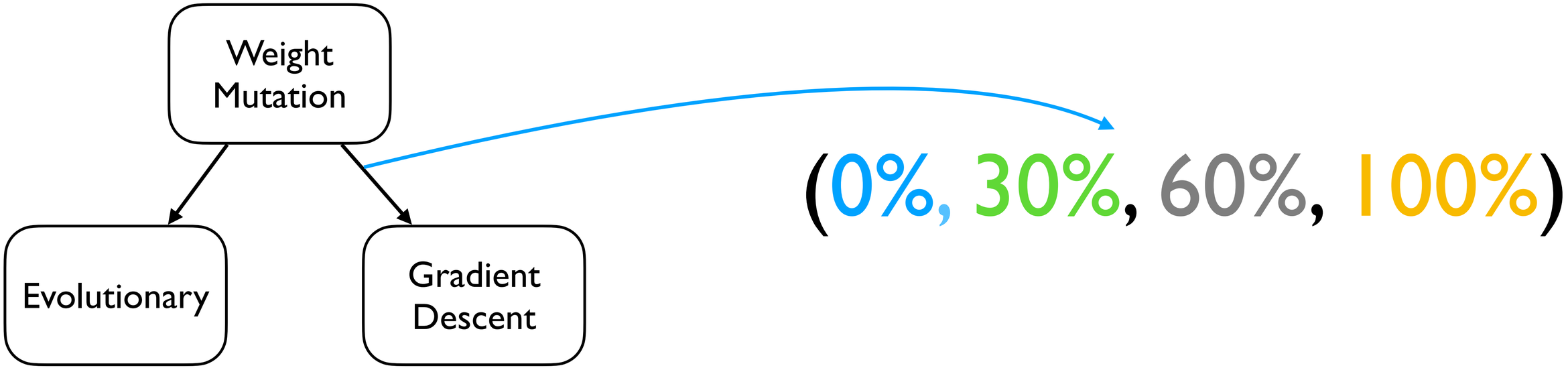
Human Gameplay Traces as Training Set



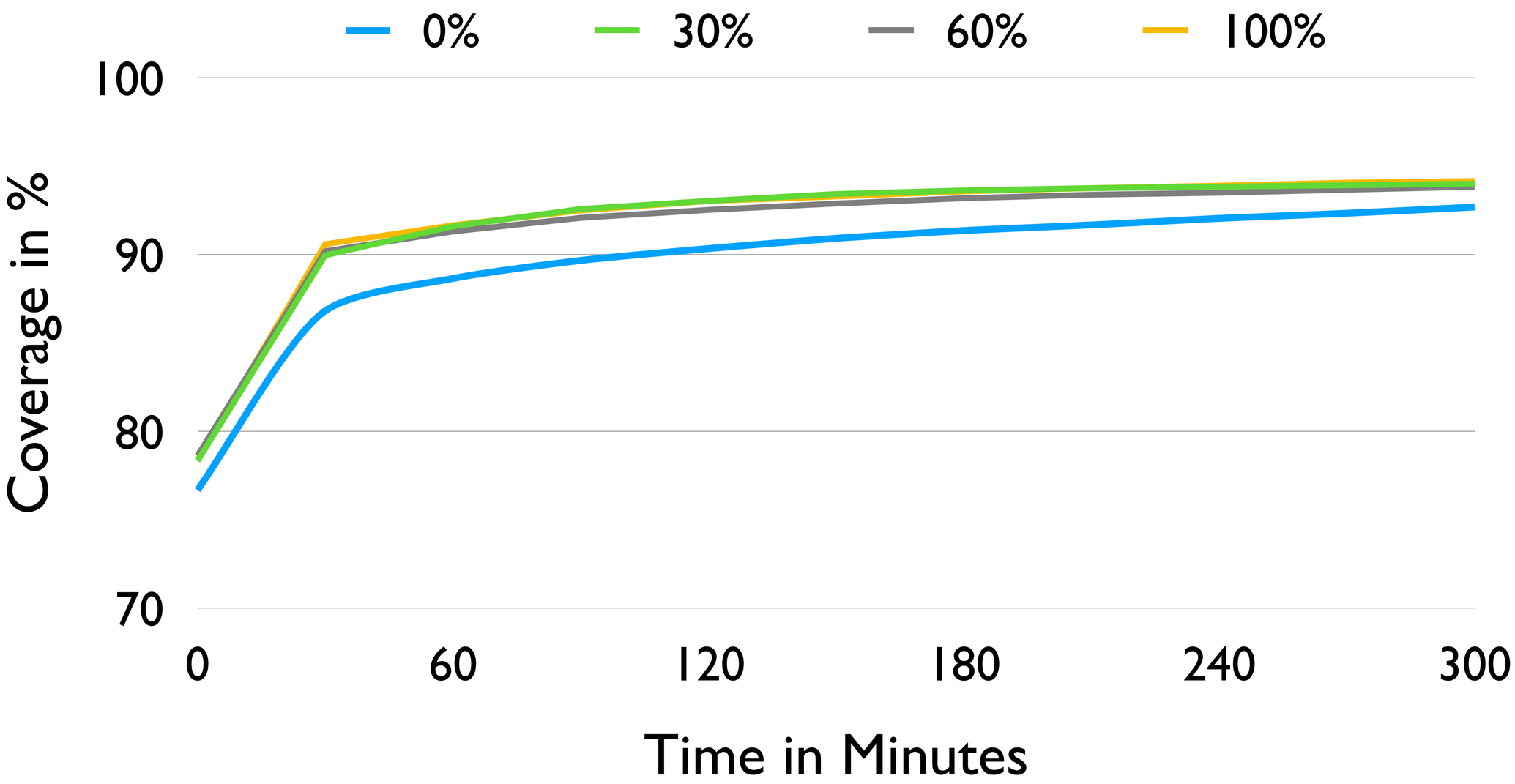
Human Gameplay Traces as Training Set



Apply Gradient-Descent with Varying Probabilities

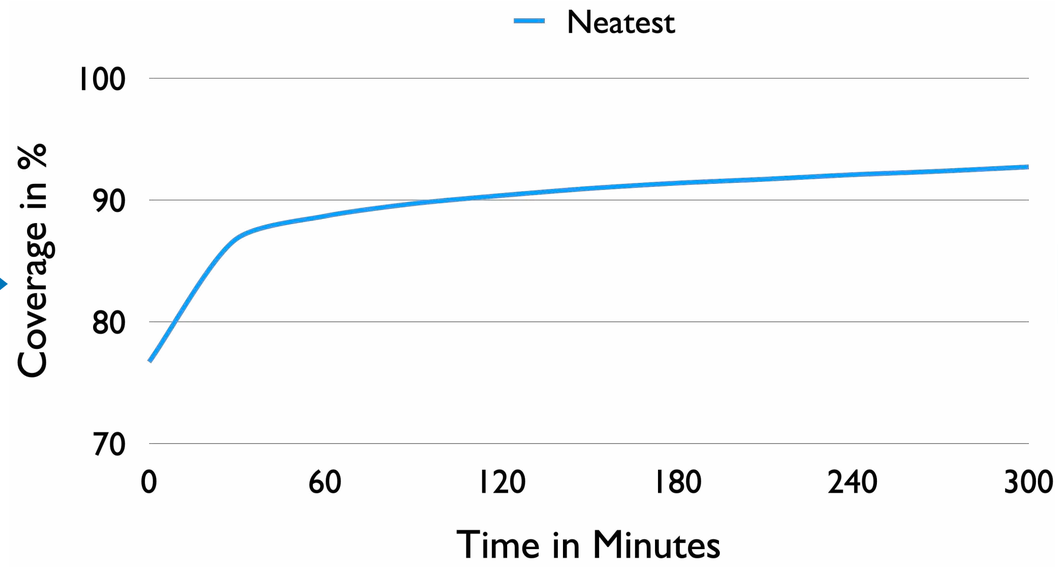


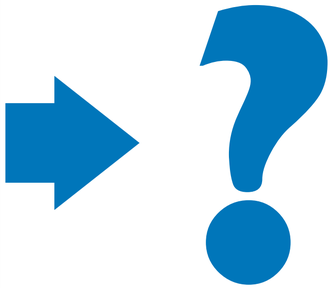
Significant Speedups through Gradient-Descent

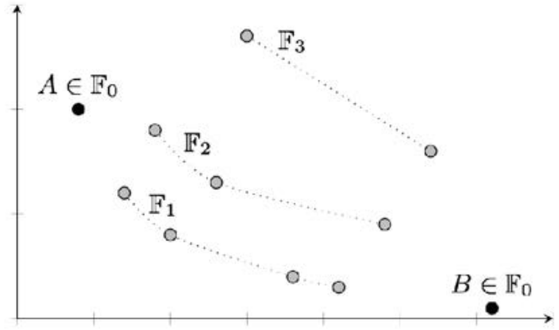


Ongoing Work: Overcoming Limitations of Neatest

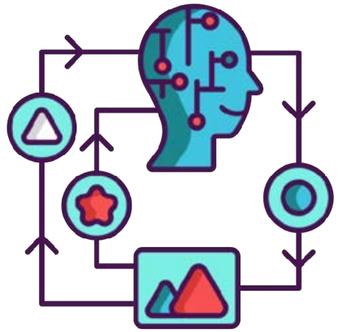
Neatest



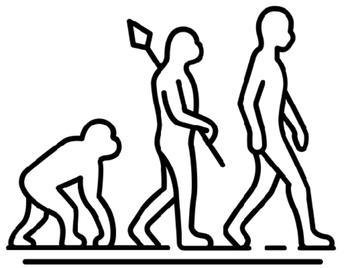




Search-based Testing

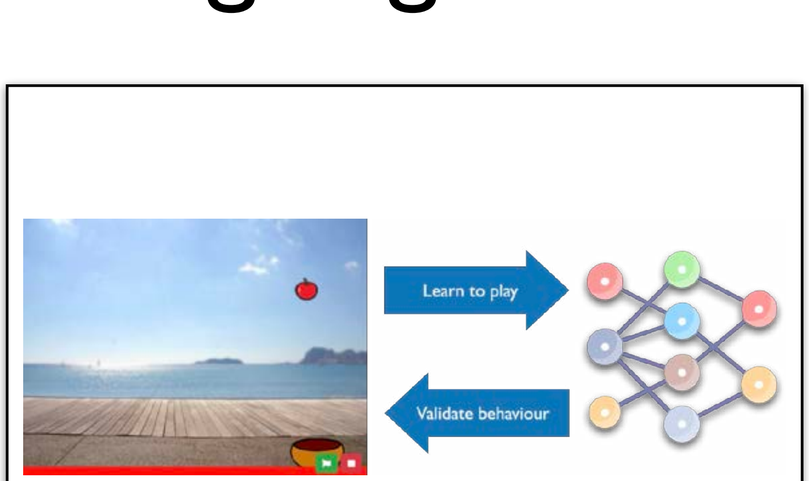


Reinforcement Learning

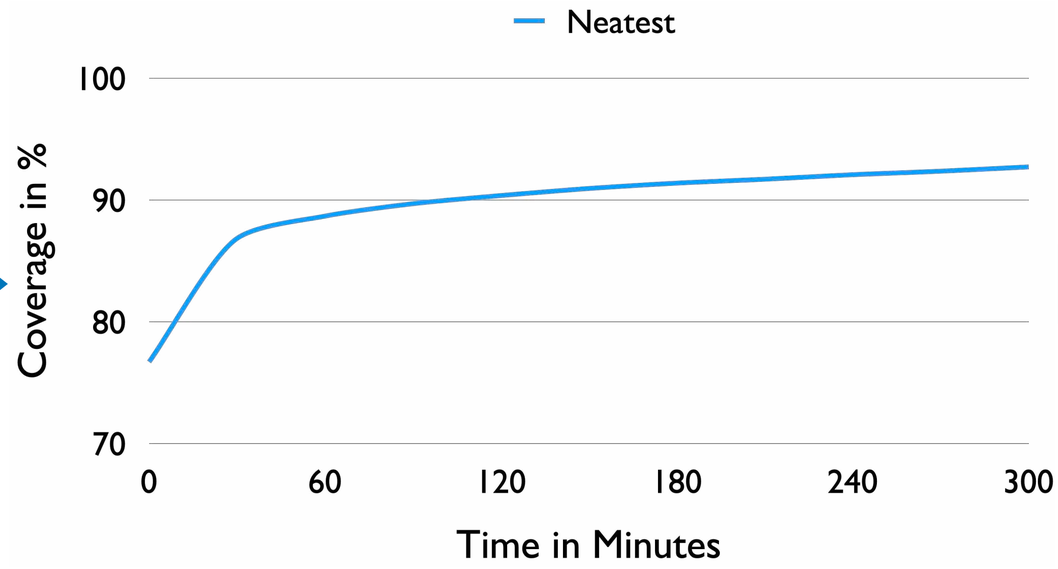


Illumination Search

Ongoing Work: Overcoming Limitations of Neatest

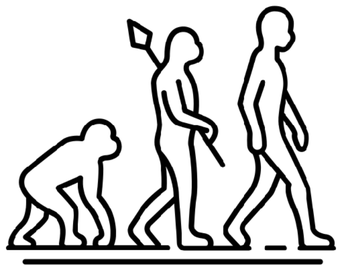


Neatest



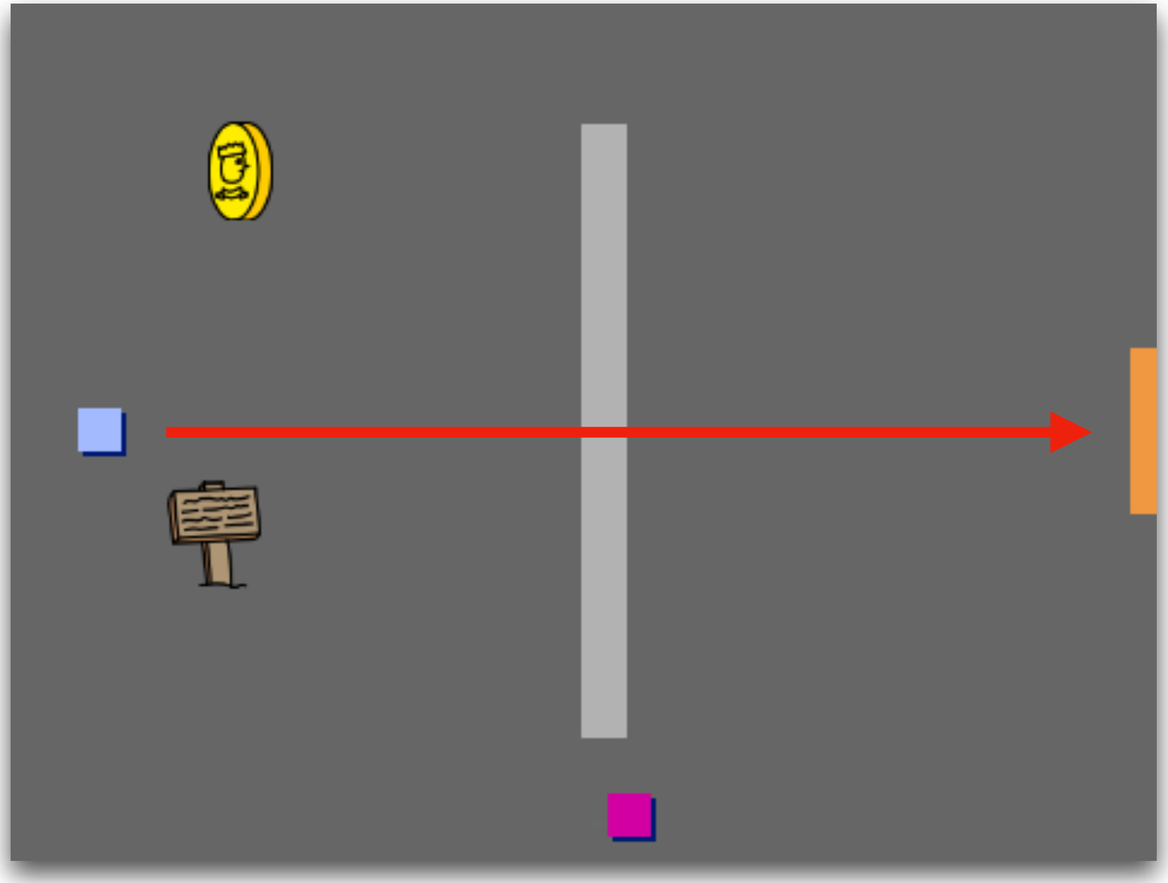
Search-based Testing

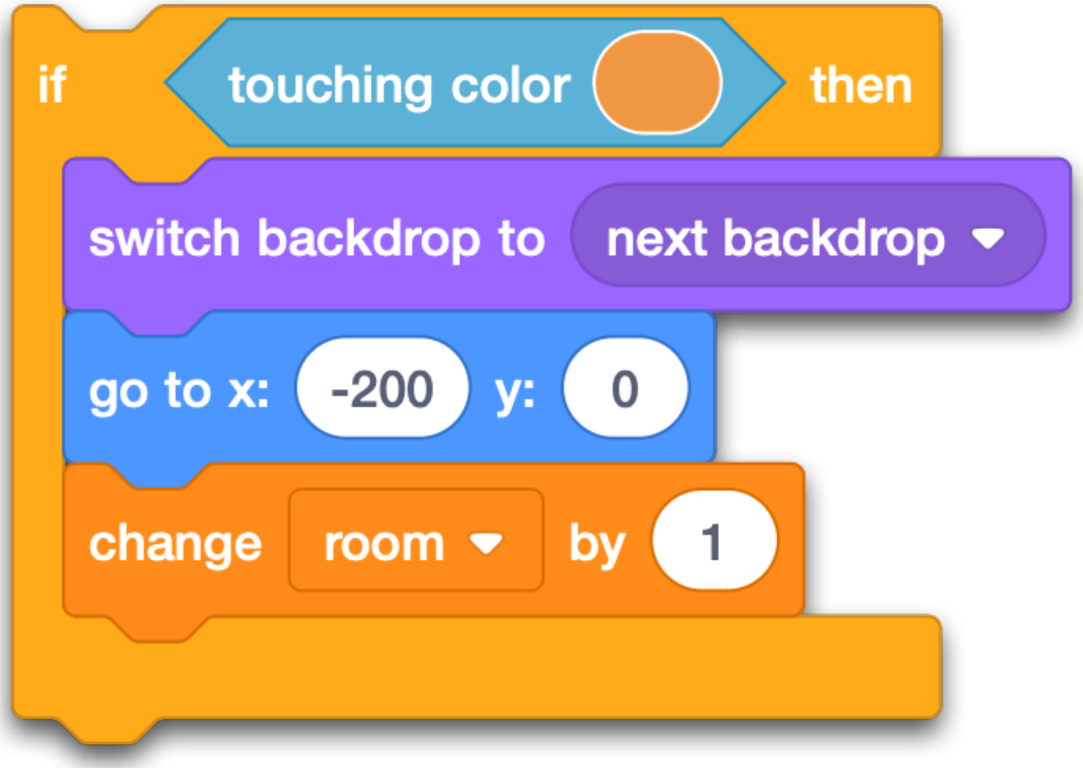
Reinforcement Learning



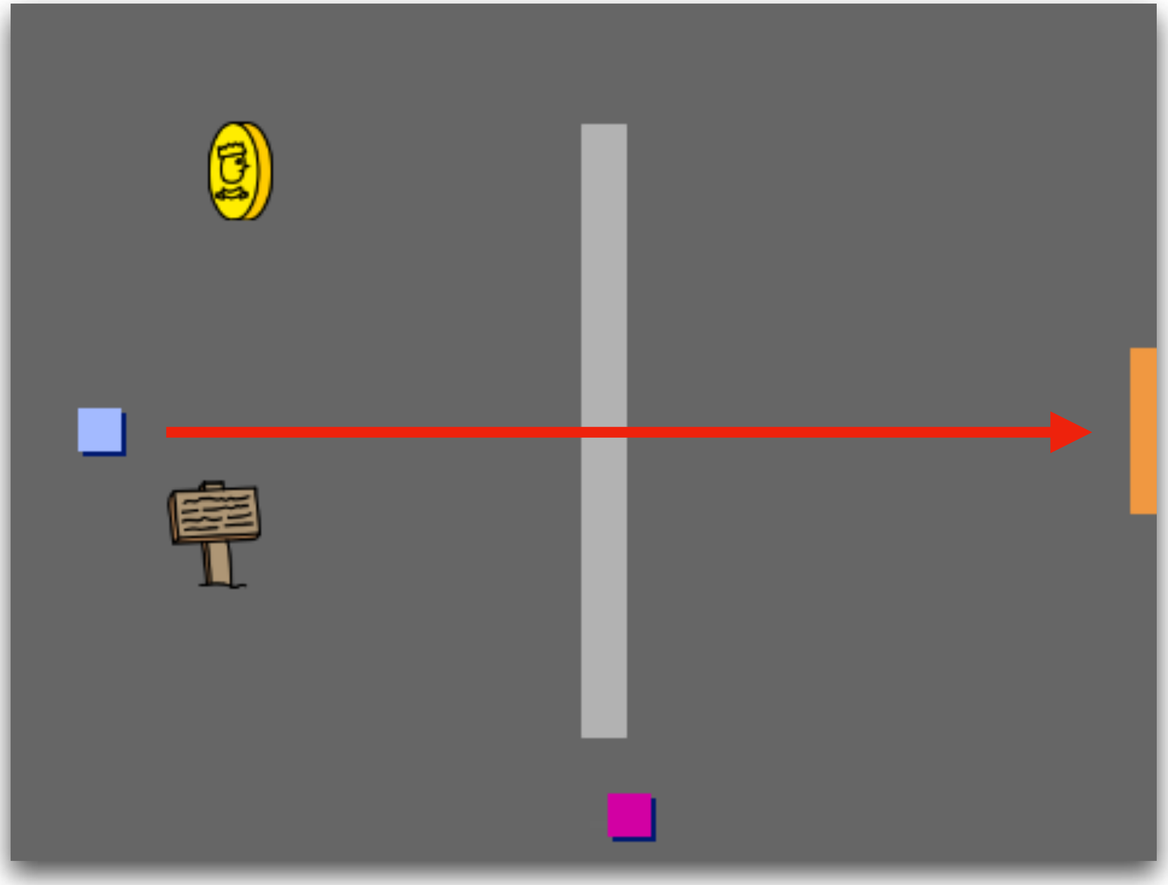
Illumination Search

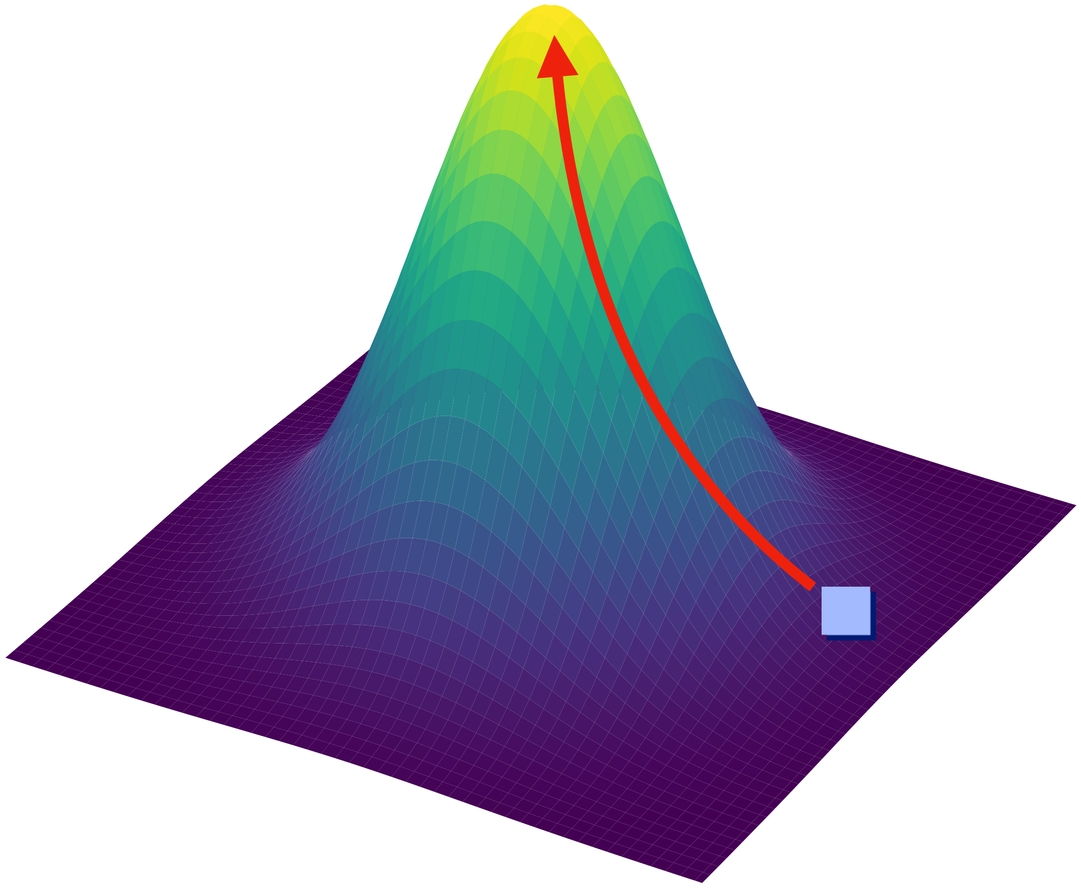
Fitness Based on Distance Towards Target Statement



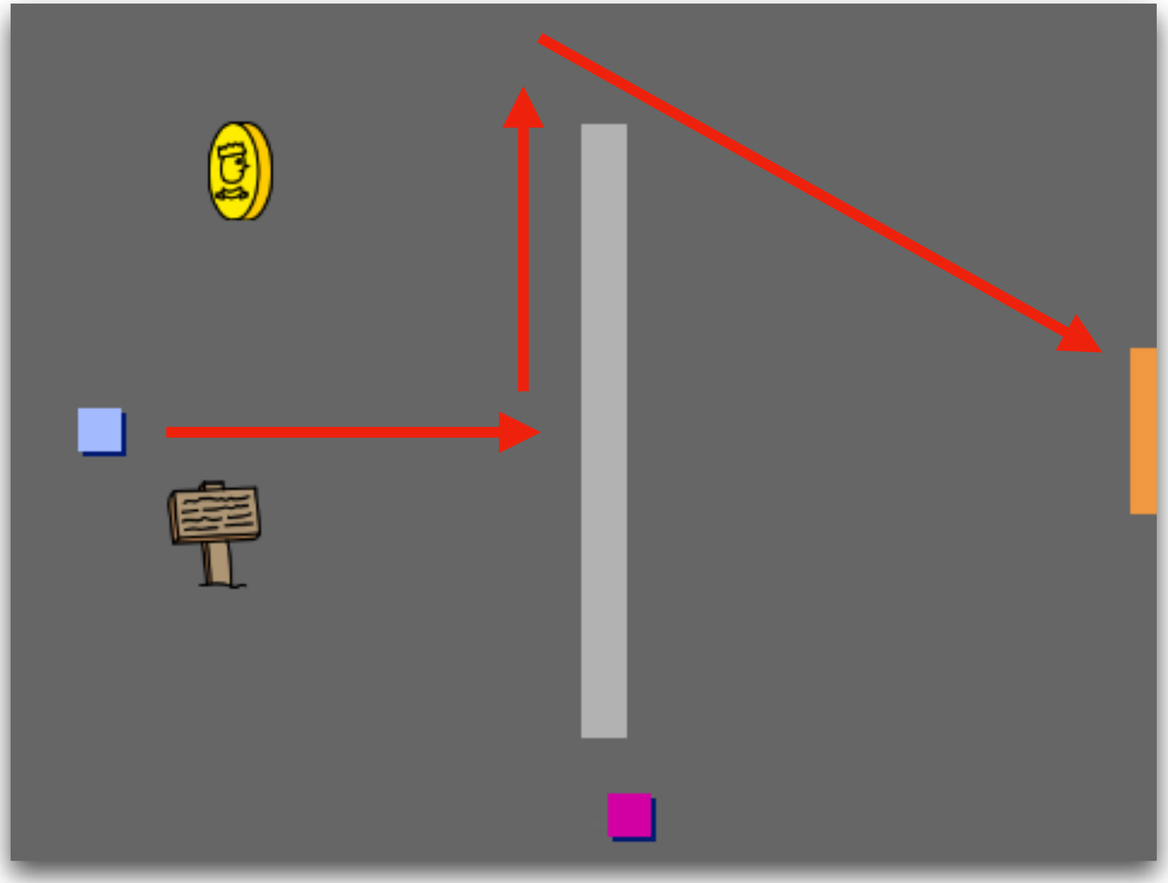


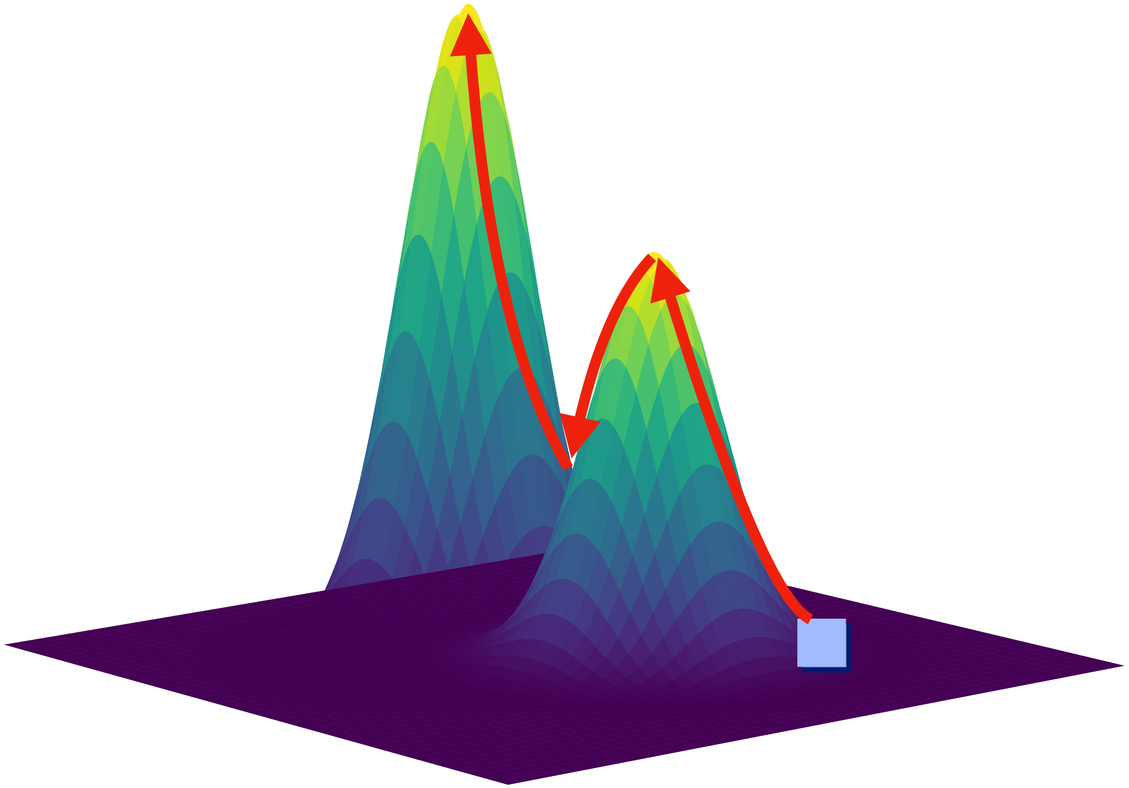
Fitness Based on Distance Towards Target Statement



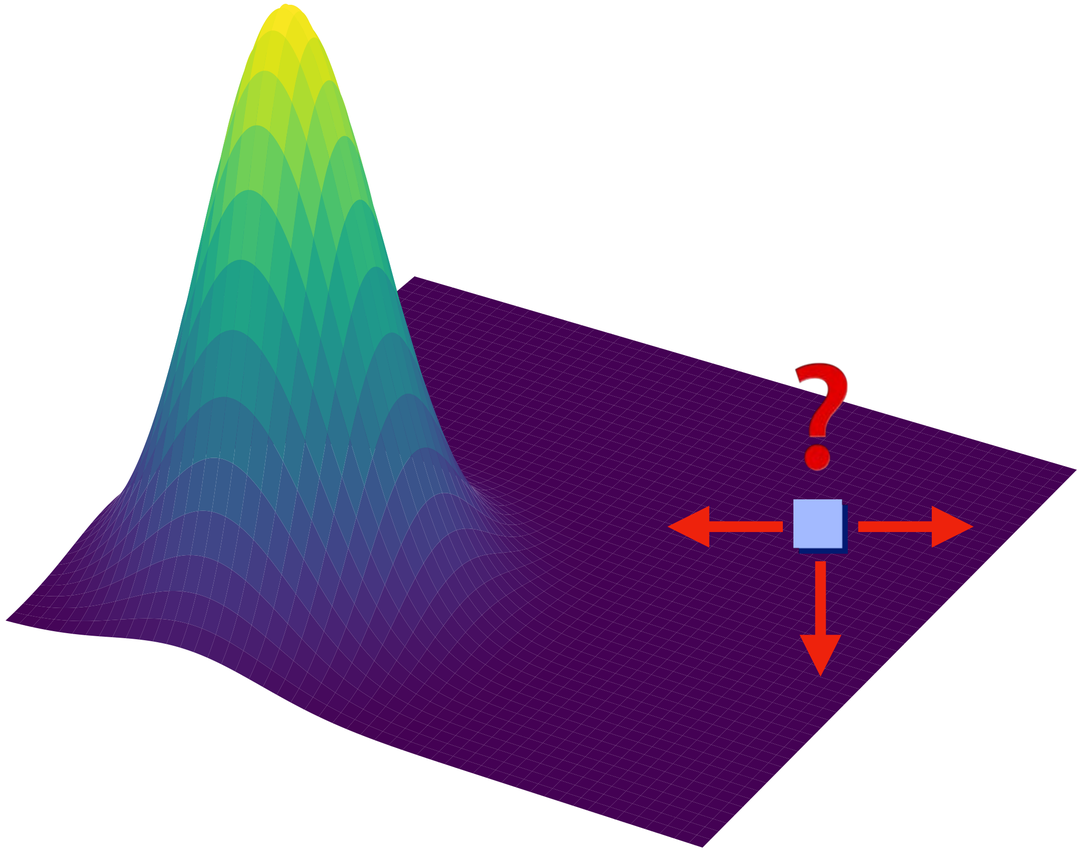


Deceiving Fitness Landscapes in Mazes

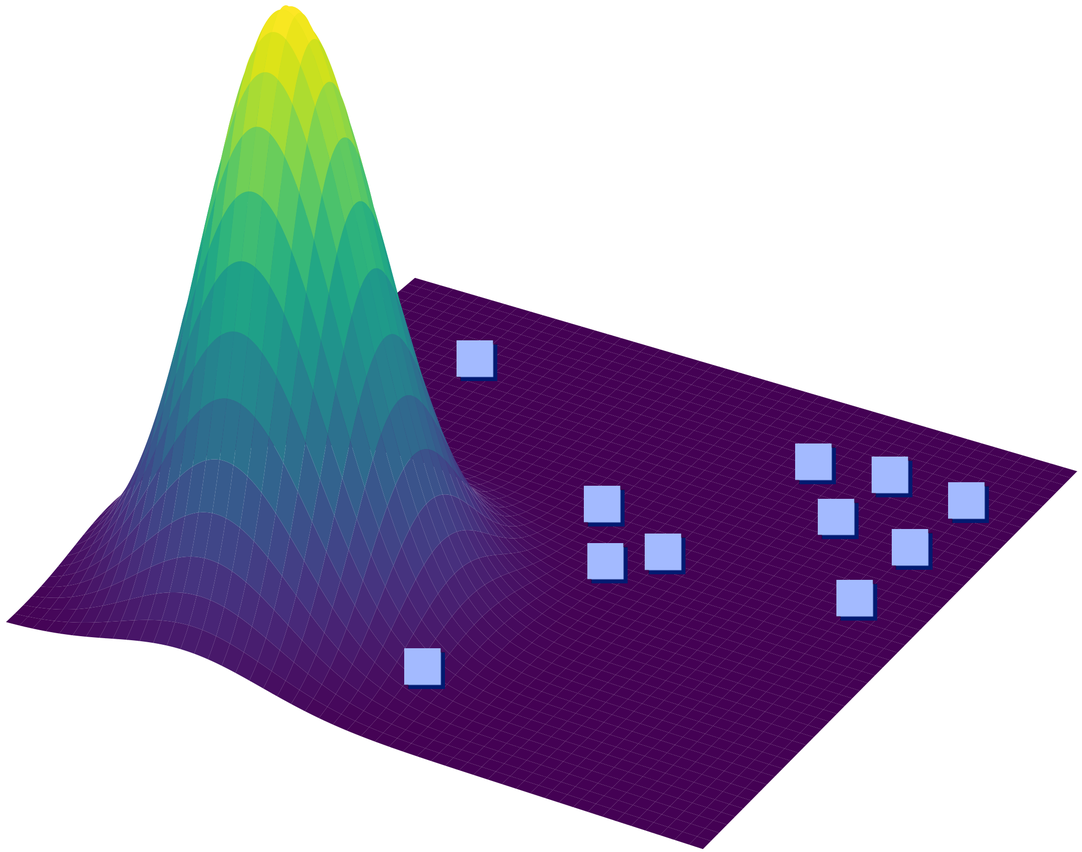




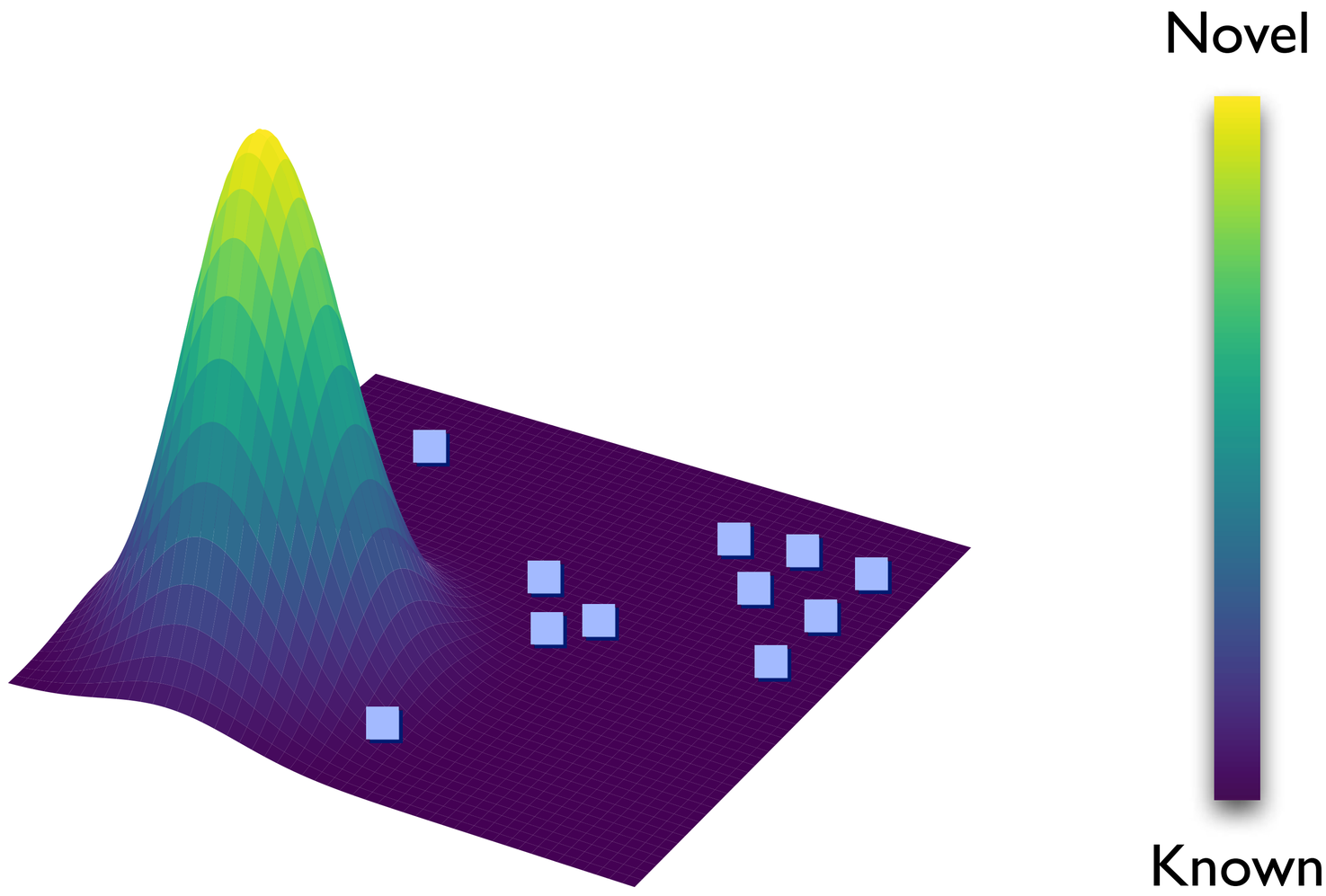
Fitness Landscapes Without Guidance



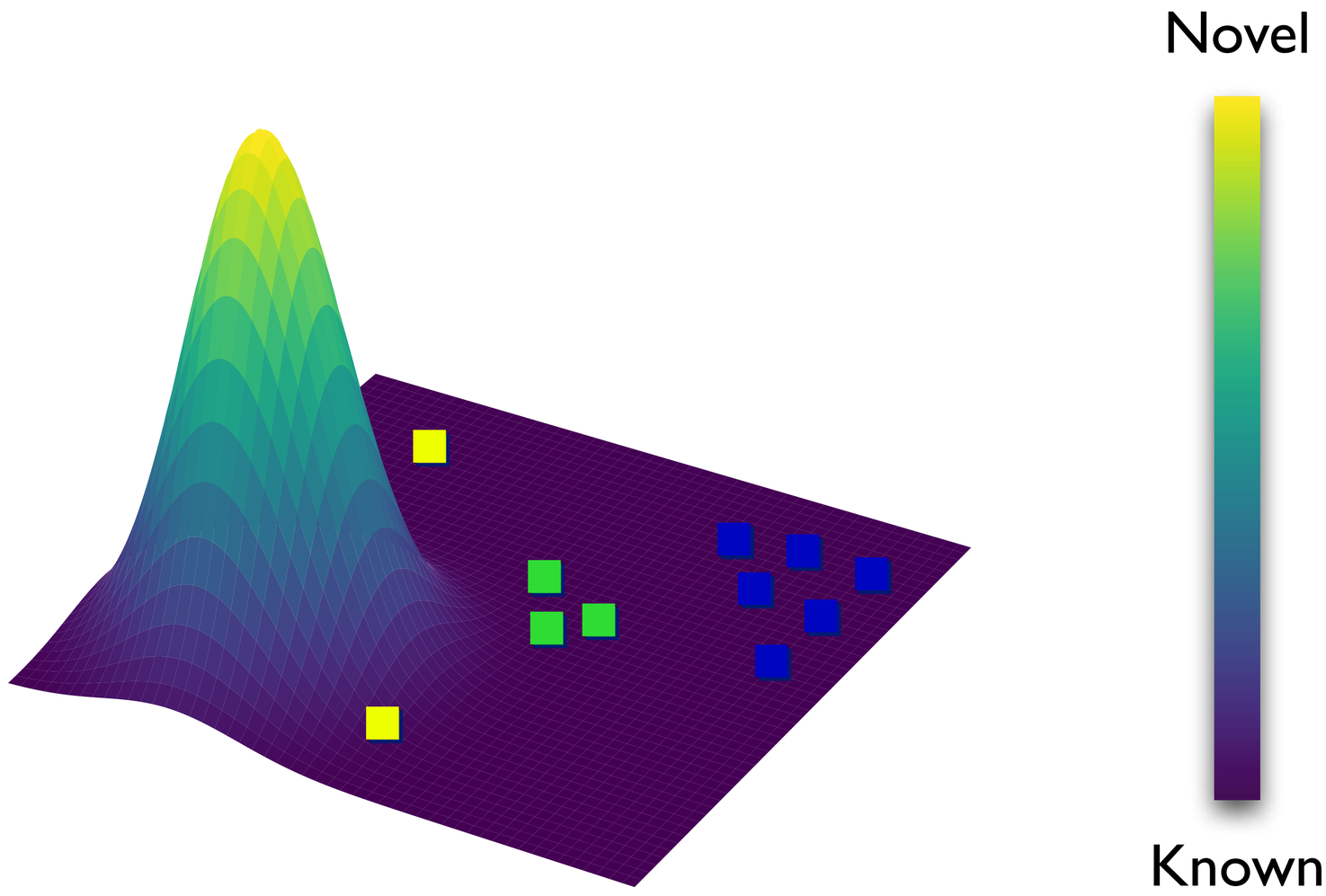
Illumination Algorithms: Integrating Novelty into the Search



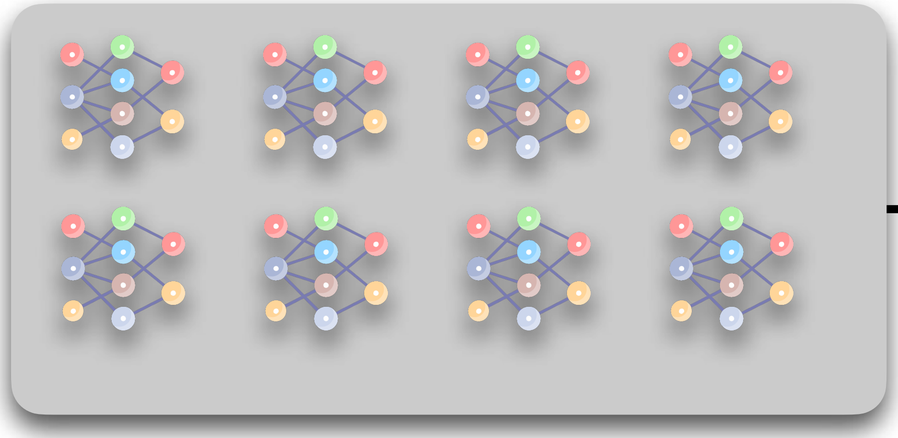
Illumination Algorithms: Integrating Novelty into the Search



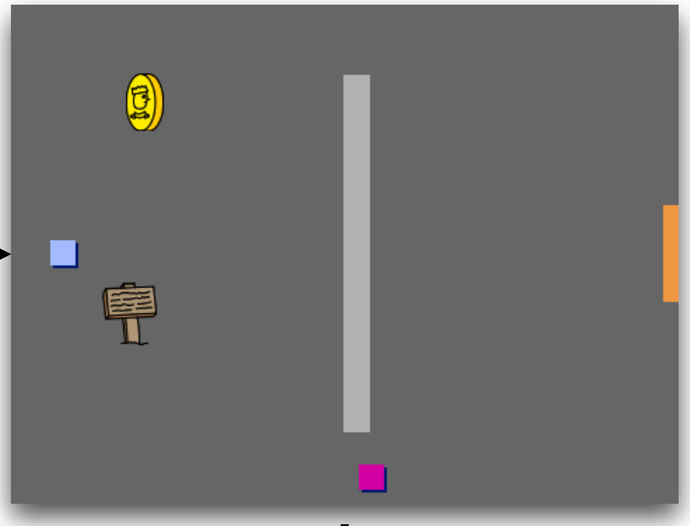
Illumination Algorithms: Integrating Novelty into the Search



Deriving Novelty from Observed Behaviours



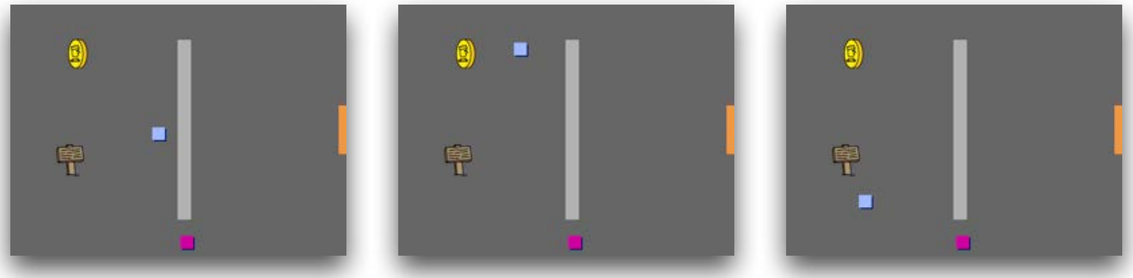
Execute Networks



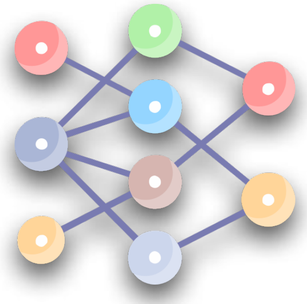
Behaviour Archive

[0.4,0.5,0.25,0.6]
[0.3,0.85,0,0.7]
[0.3,0.25,0.5,0.65]

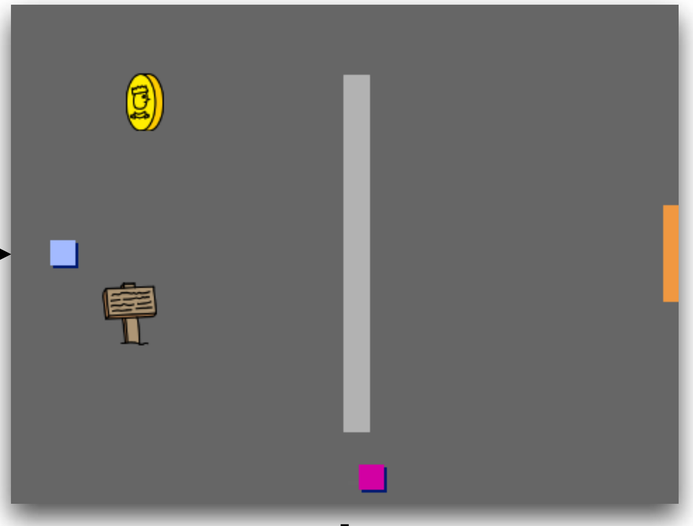
$[x_P, y_P, \alpha, d_{PO}]$



Comparing Behaviours Using Cosine Similarity



Execute Network



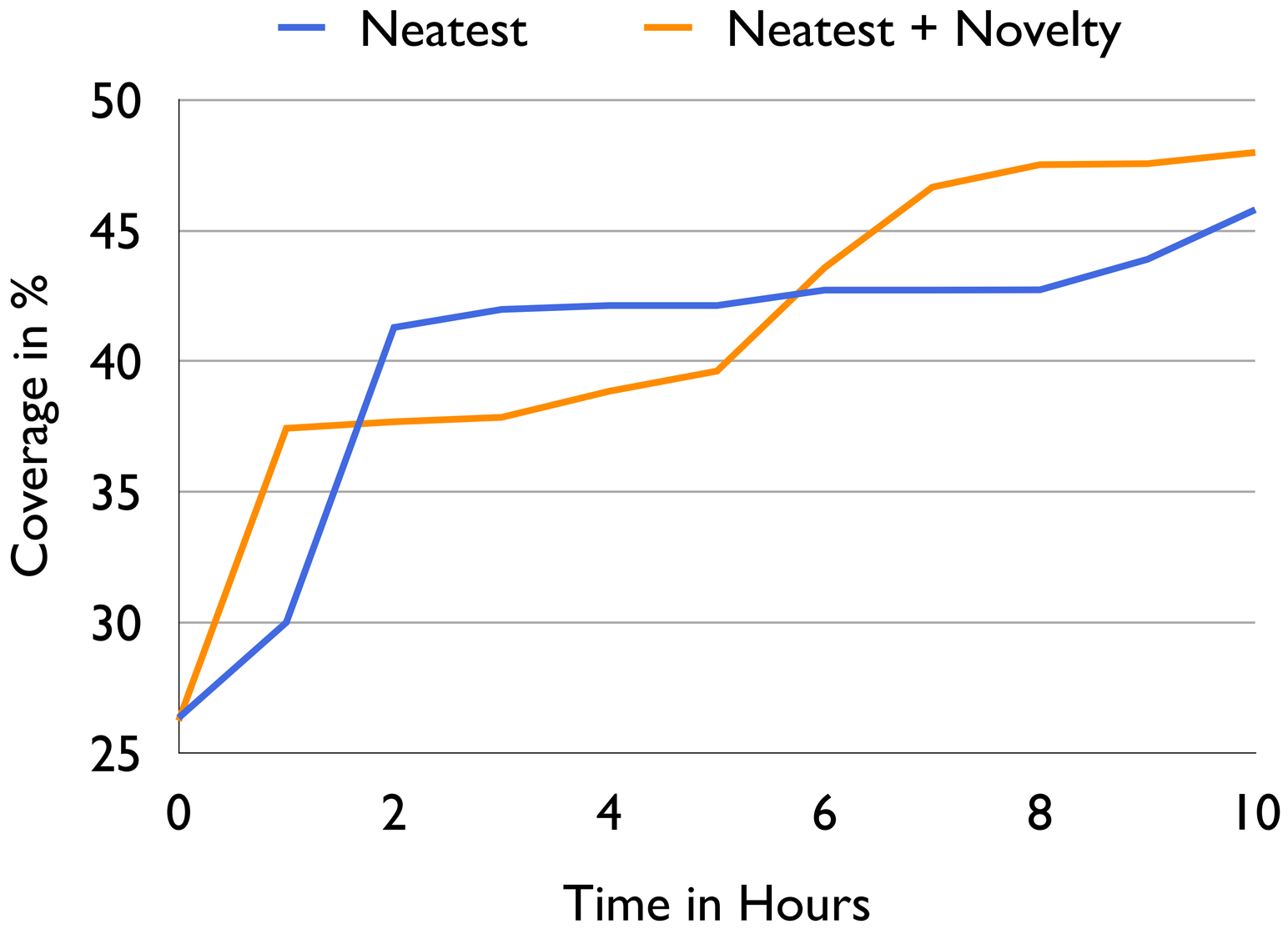
Behaviour Archive

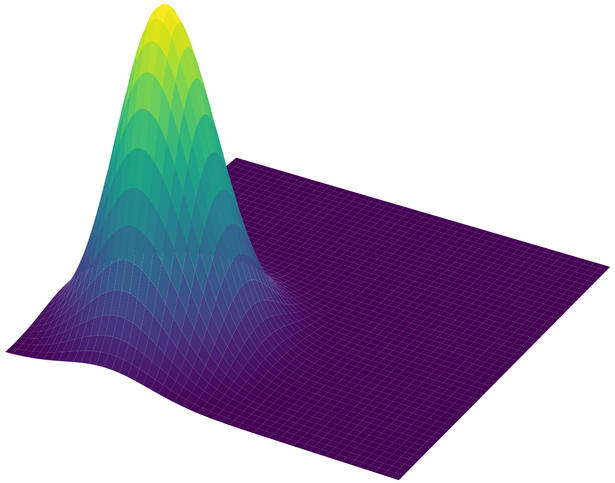
[0.4,0.5,0.25,0.6]
[0.3,0.85,0,0.7]
[0.3,0.25,0.5,0.65]

Cosine Similarity

[0.8,0.1,0.7,0.3]

Novelty Adds Guidance to Challenging Landscapes

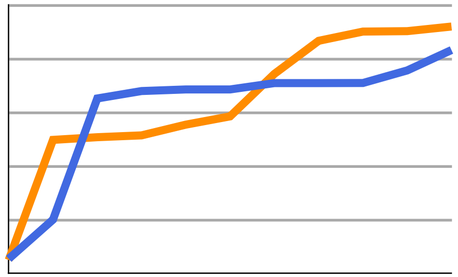




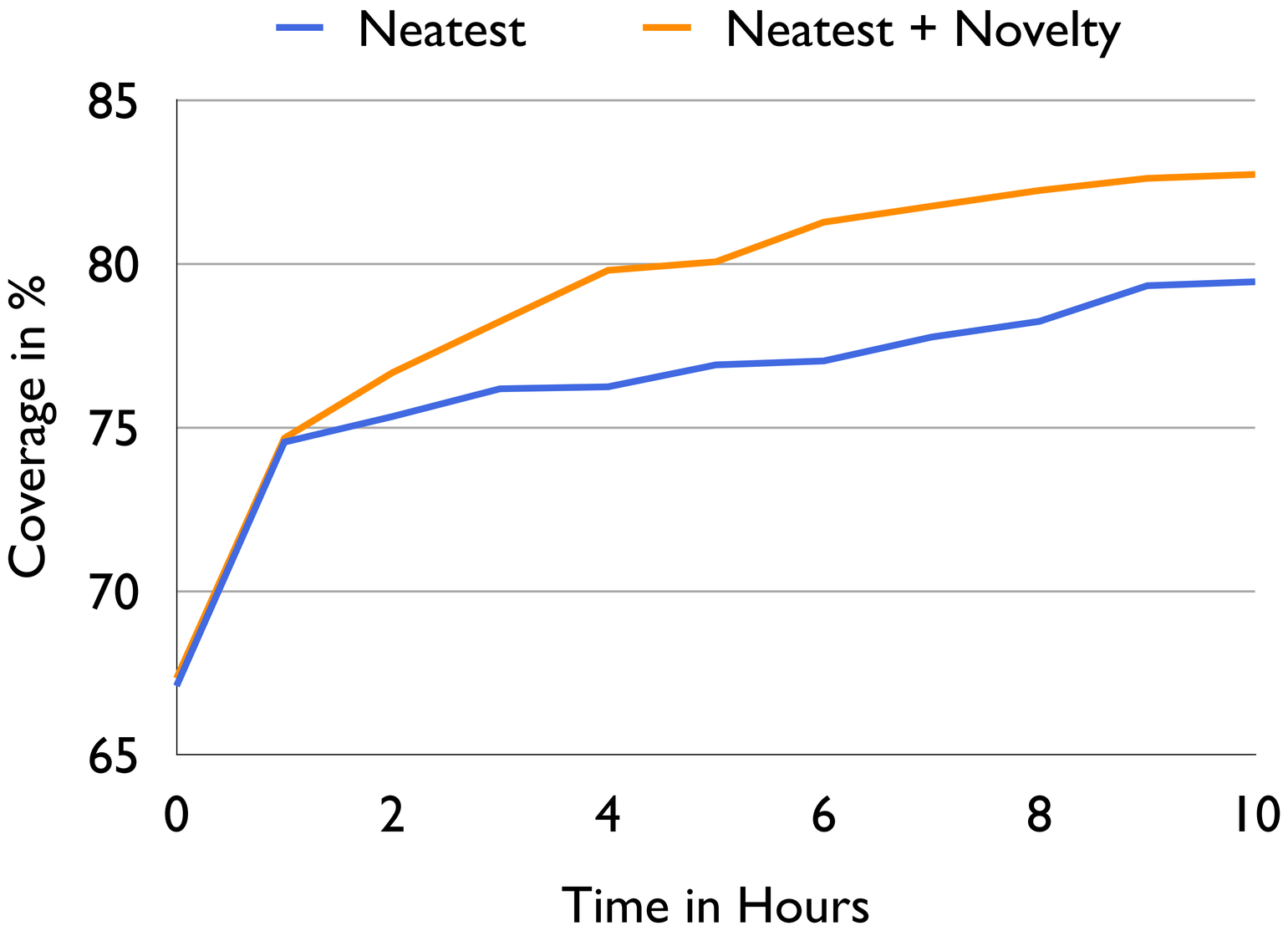
Guidance Helps Discover More Game States

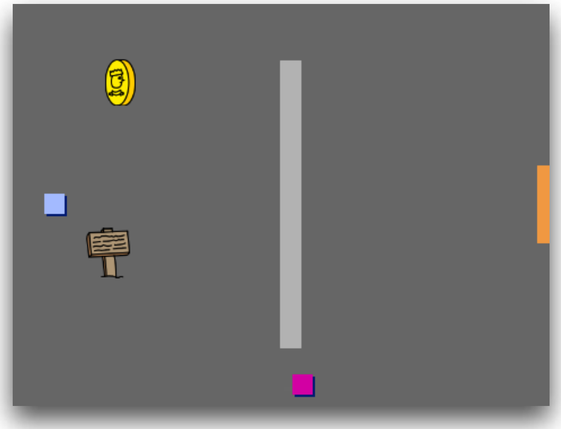
Neatest

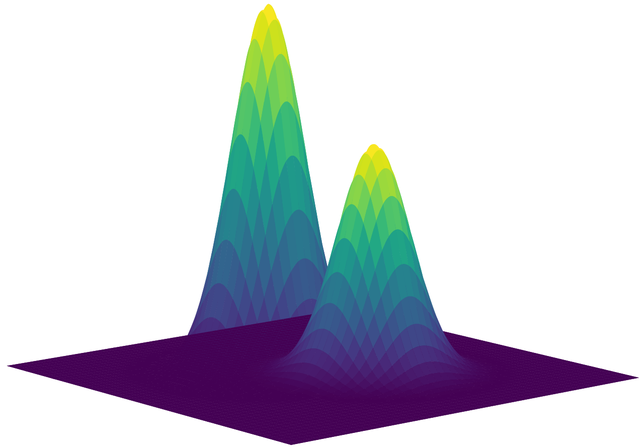
Neatest + Novelty



Novelty Helps Overcome Local Optima

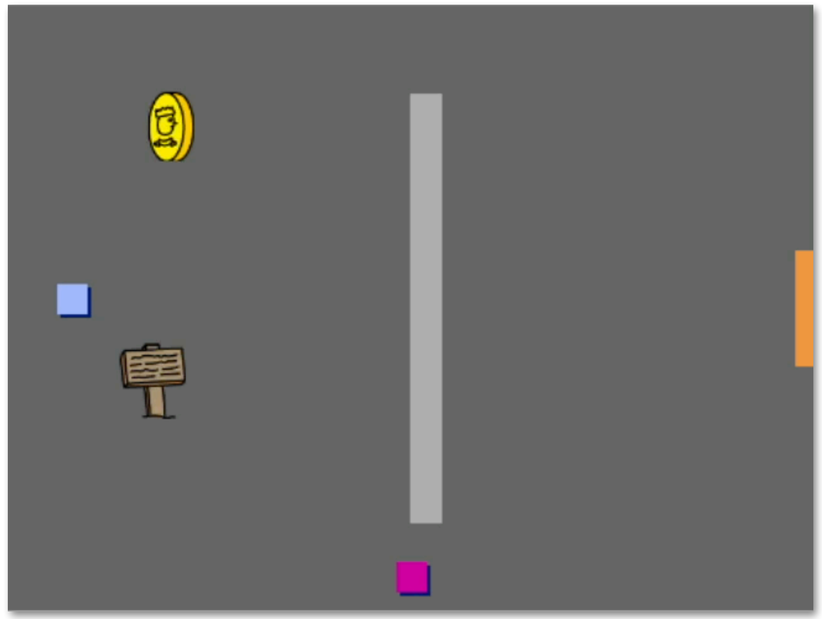




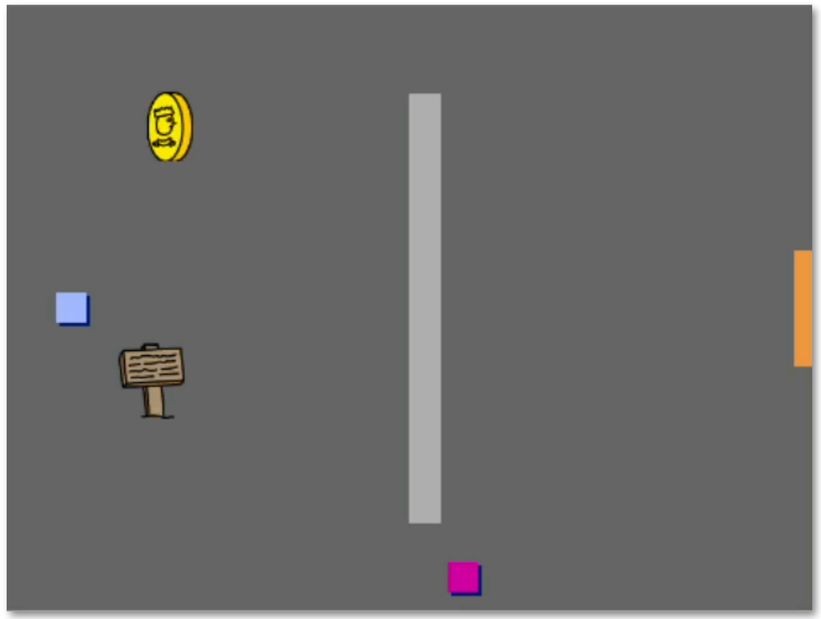


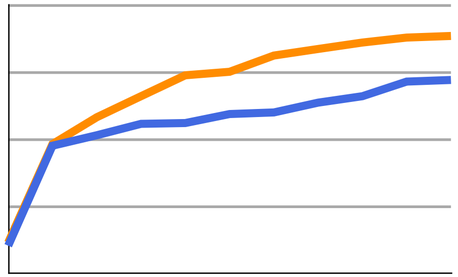
Overcoming Local Optima Helps Reaching More Levels

Neatest



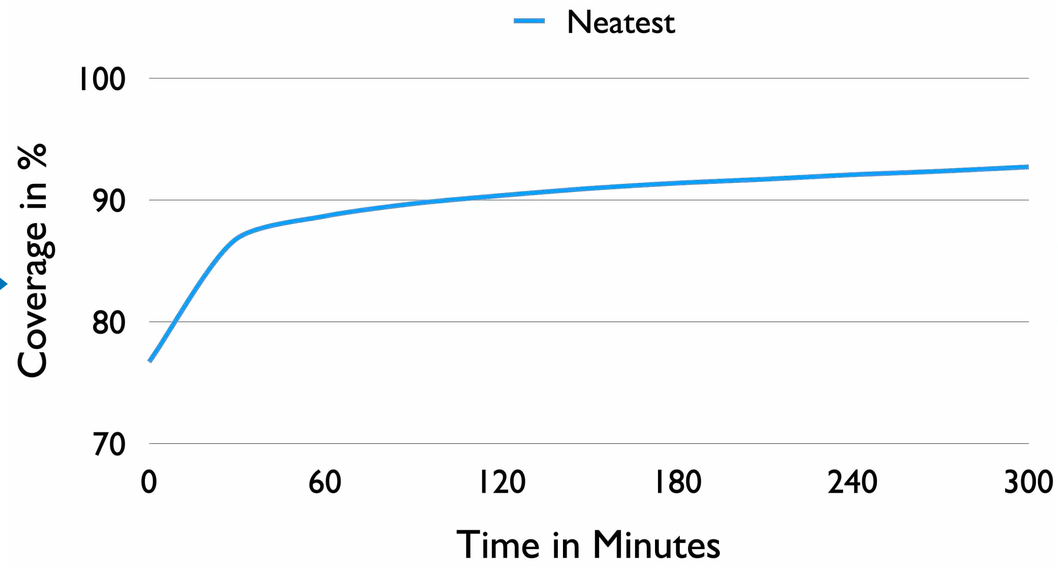
Neatest + Novelty

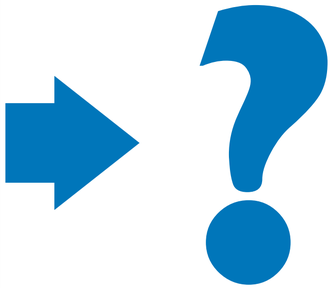


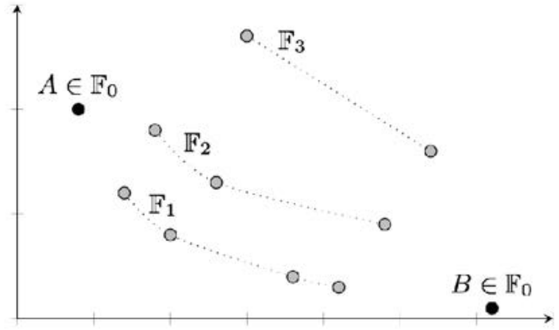


Ongoing Work: Overcoming Limitations of Neatest

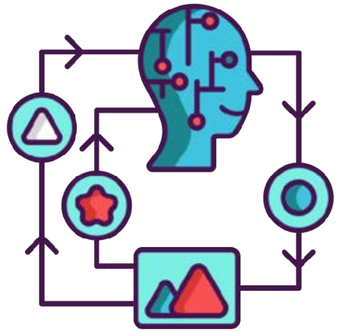
Neatest



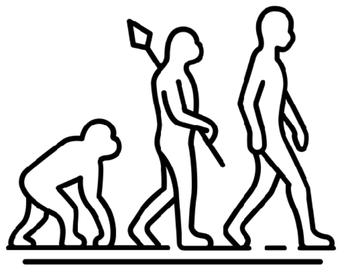




Search-based Testing

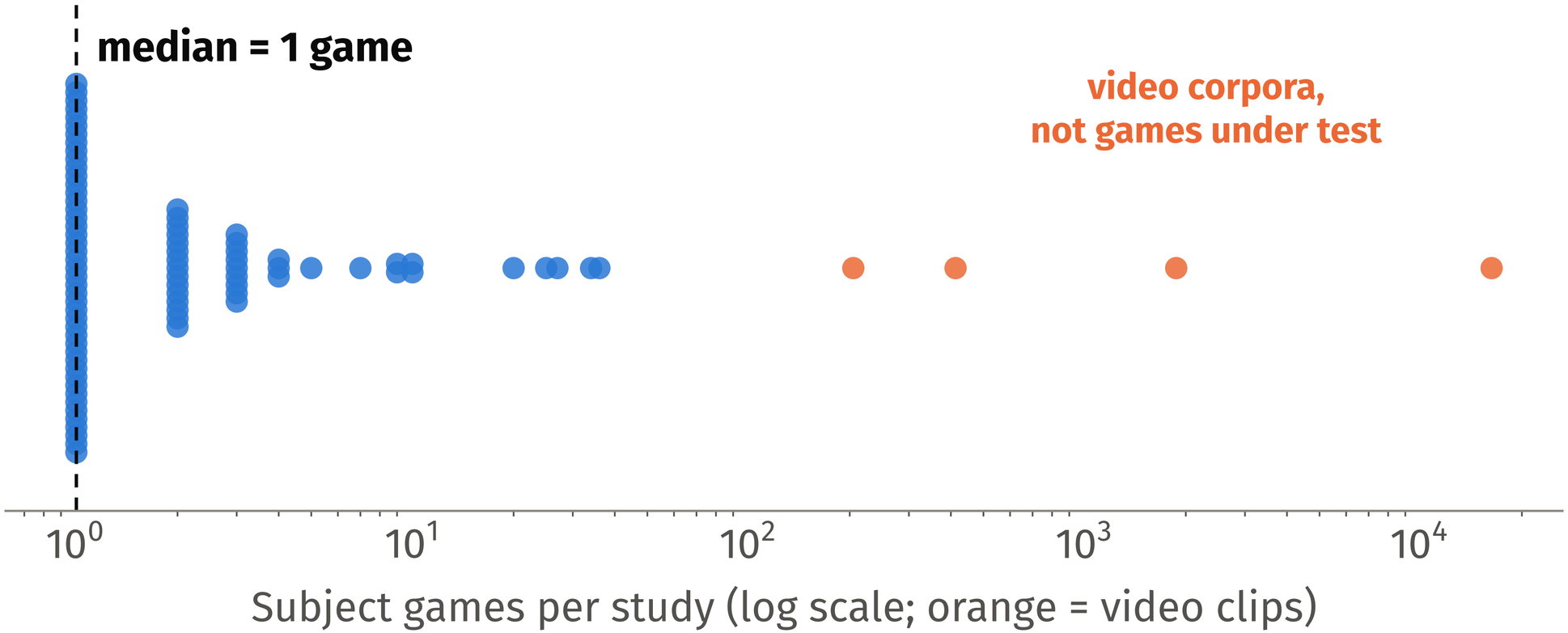


Reinforcement Learning

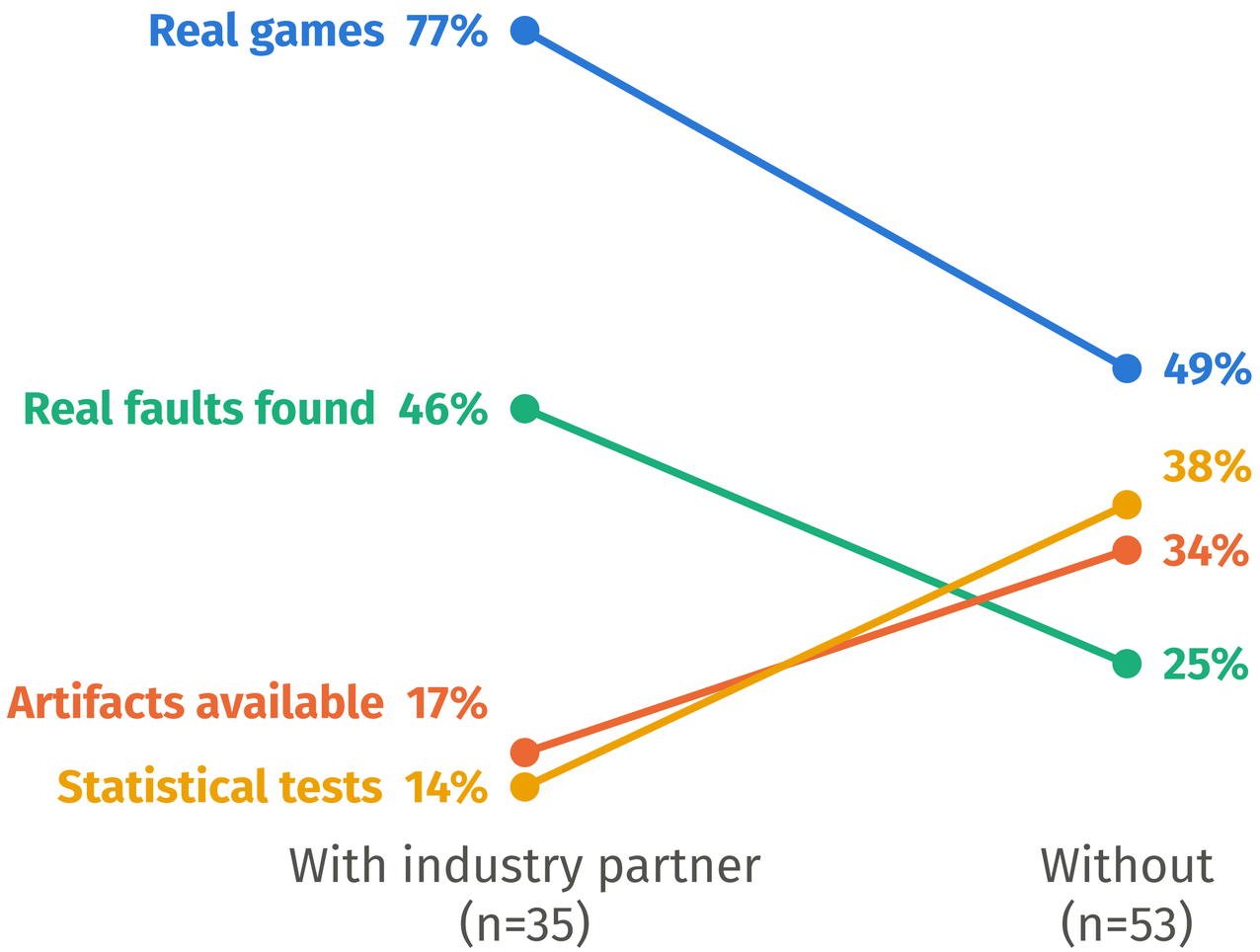


Illumination Search

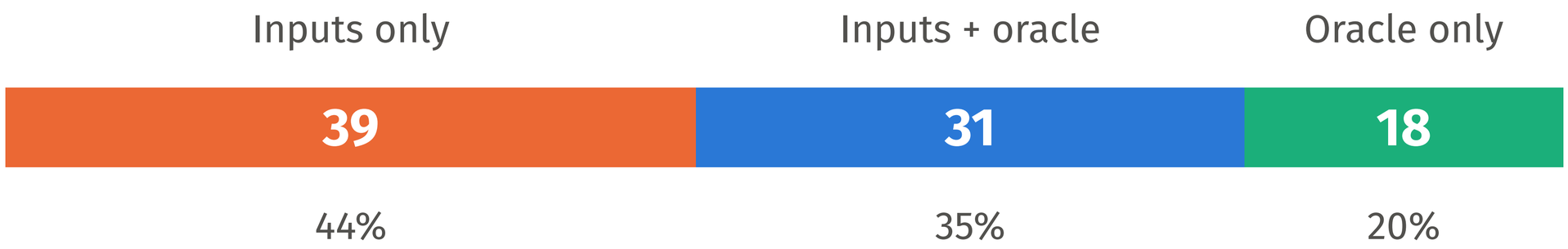
Literature Survey



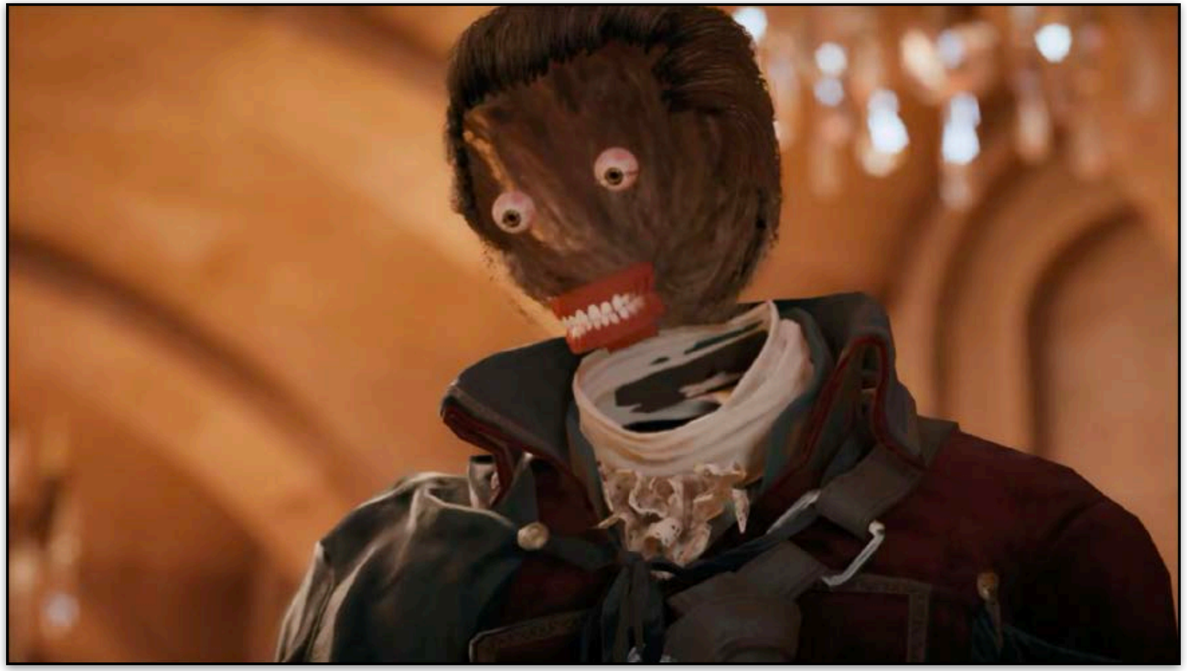
Literature Survey



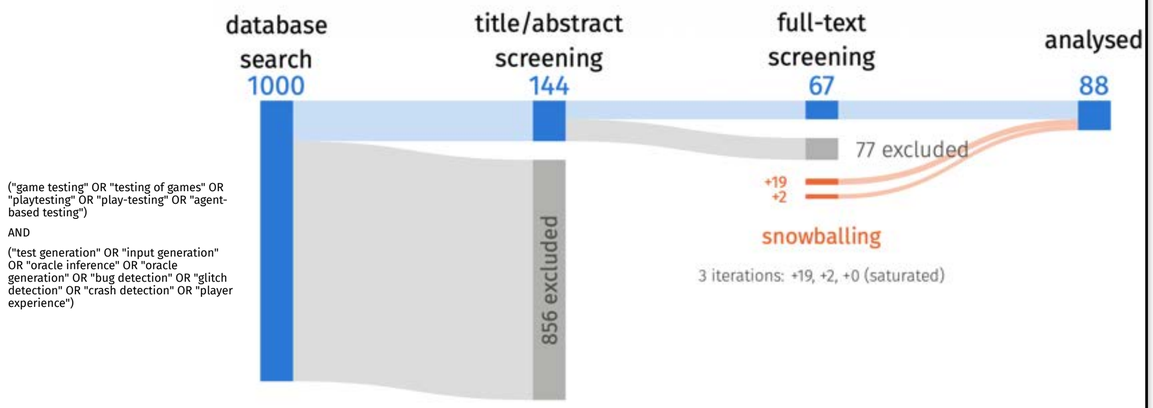
Literature Survey



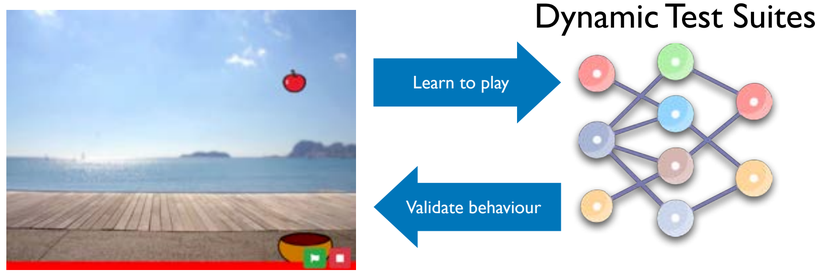
44% of all papers generate inputs with no automated oracle



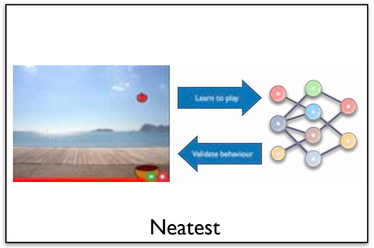
Literature Survey



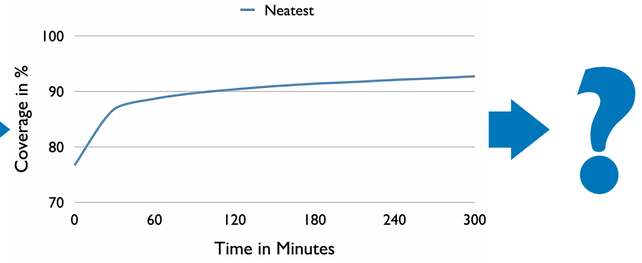
Neurevolution-based Testing with WHISKER

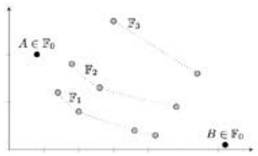


Ongoing Work: Overcoming Limitations of Neatest



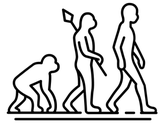
Neatest





Search-based Testing

Reinforcement Learning



Illumination Search