Inspector: Pixel-Based Automated Game Testing via Exploration, Detection, and Investigation

Liu G, Cai M, Zhao L,Qin T,Brown A, Bischoff J, Liu T
2022 IEEE Conference on Games

Current situation

- RL based automated game testing has attracted much attention

- However, those methods rely on game internal state information which has several limitations.
  - Such as player position, player velocity, game reward, etc.,

Goal

- To build a general game testing agent/tool, named Inspector that is not suffer from the above limitations.
- It based on game screenshots.
- Interacting with key objects in games like human players/testers

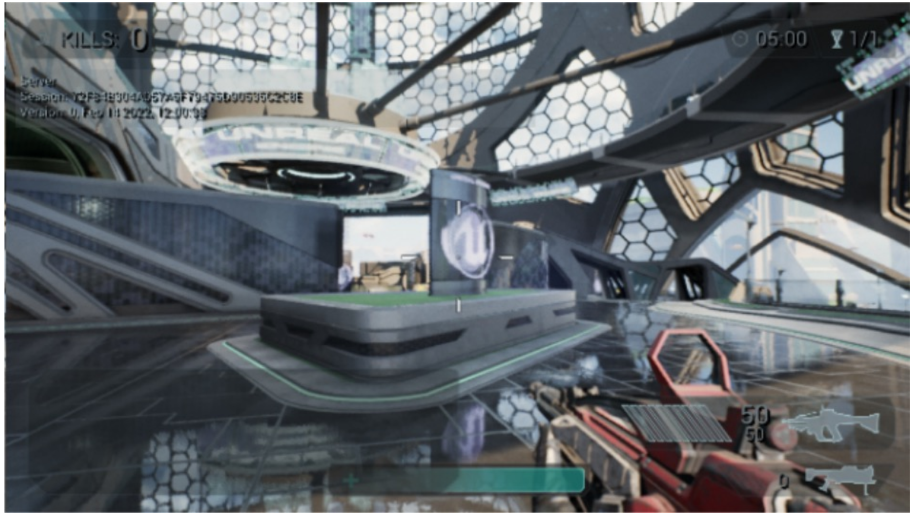
Inspector

Consist of

1) Game space explorer.
2) Key object detector.
3) Human-like object investigator.

Experiments

- Apply the agent to FPS game and Action game. (developed by UE)



Inspector - 1)Game space explorer

- Design a curiosity-driven reward function, based on **Random Network Distillation (RND)**
  - Target network $f : \mathcal{S} \to \mathbb{R}^k$
  - Predictor network $\hat{f} : \mathcal{S} \to \mathbb{R}^k$

$$\psi^* = \min_{\psi \in \Psi} \frac{1}{N} \sum_{i=1}^{N} ||\hat{f}(s_i; \psi) - f(s_i)||^2$$

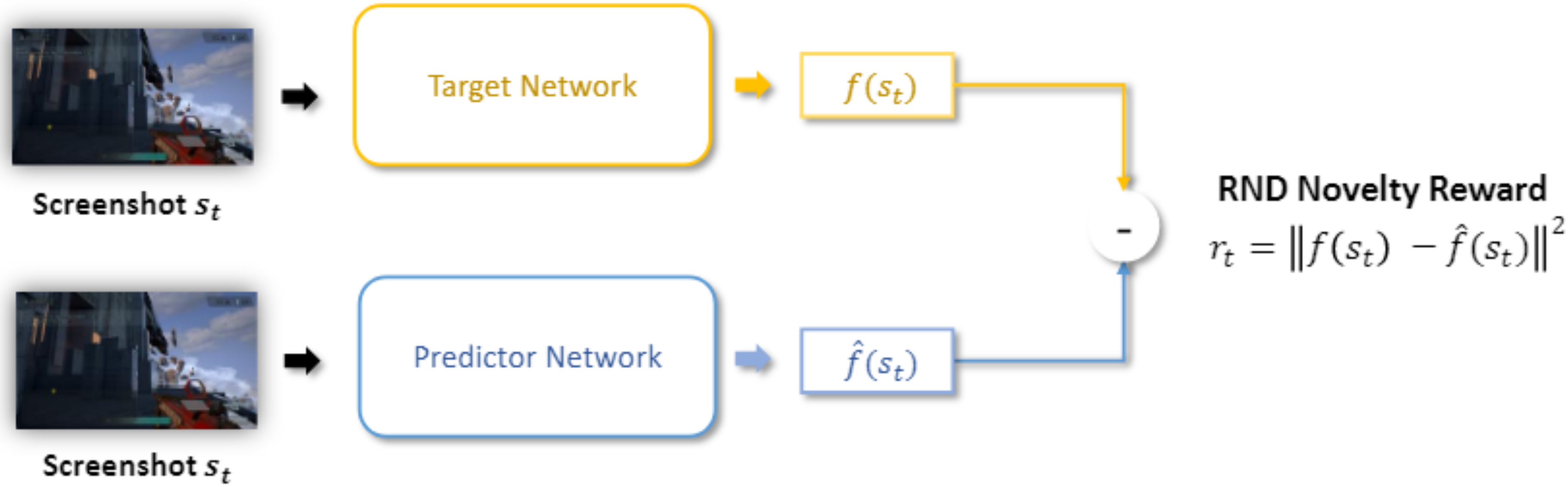
N: number of screenchots

Inspector - 1)Game space explorer

- the Reward function

$$r_t = ||f(s_t) - \hat{f}(s_t)||^2$$

Inspector - 1)Game space explorer



Experiments -Game space explorer-

- PPO algorithm to train an exploration policy
  - 4 convolution layers and a fully connected layer.
- Predictor network and target network
  - 3 convolution layers and fully connected layers.(the output embedding size is 512)

batch size : 2000
The number of epochs: 20.
Adam optimizer (learning rate of 3e− 4)

Inspector – 2) Key object detector

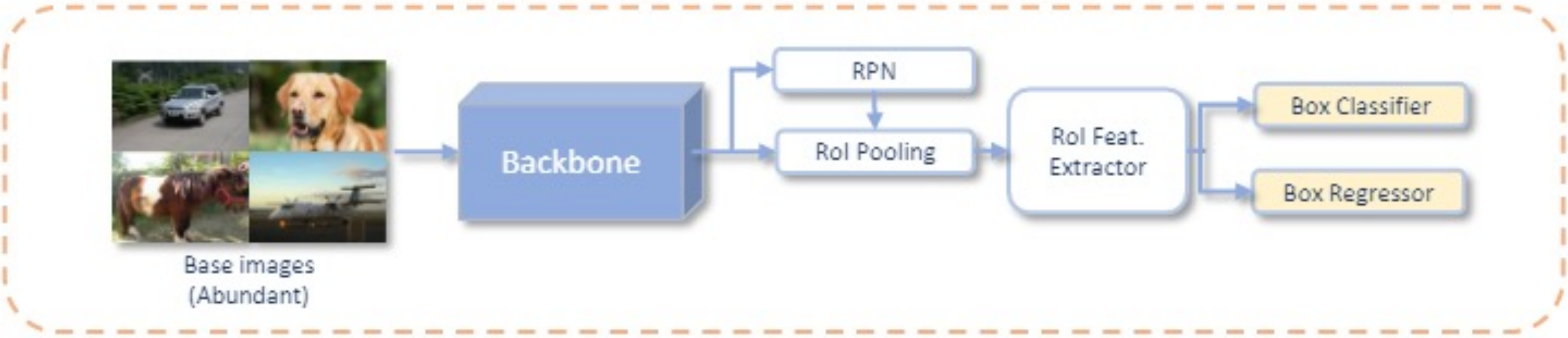
- Detect and investigate key objects like human testers.

- Two-stage fine-tuning method for few-shot object detection

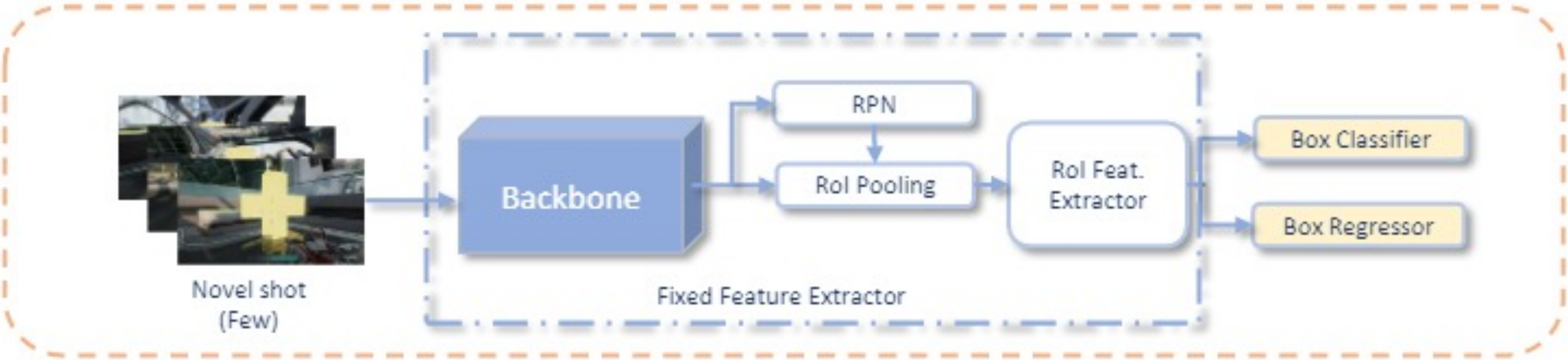
Inspector – 2) Key object detector

- Pre-train the entire object detector using **MSCOCO dataset**.
- Fine-tune the last layers using a small number of labeled screenshots
- base detection model, **Faster R-CNN**

Stage 1 : Base training



Stage 2 : Few-shot fine-tuning



Experiments -Key object detector-

- Faster R-CNN as base detector.
- Resnet-101 with FPN as backbone.
- SGD optimizer
  - batch size of 16,
  - momentum of 0.9,
  - weight decay of 0.0001

    the learning rate is 0.02(base training) and 0.001(few-shot fine-tuning).

Inspector

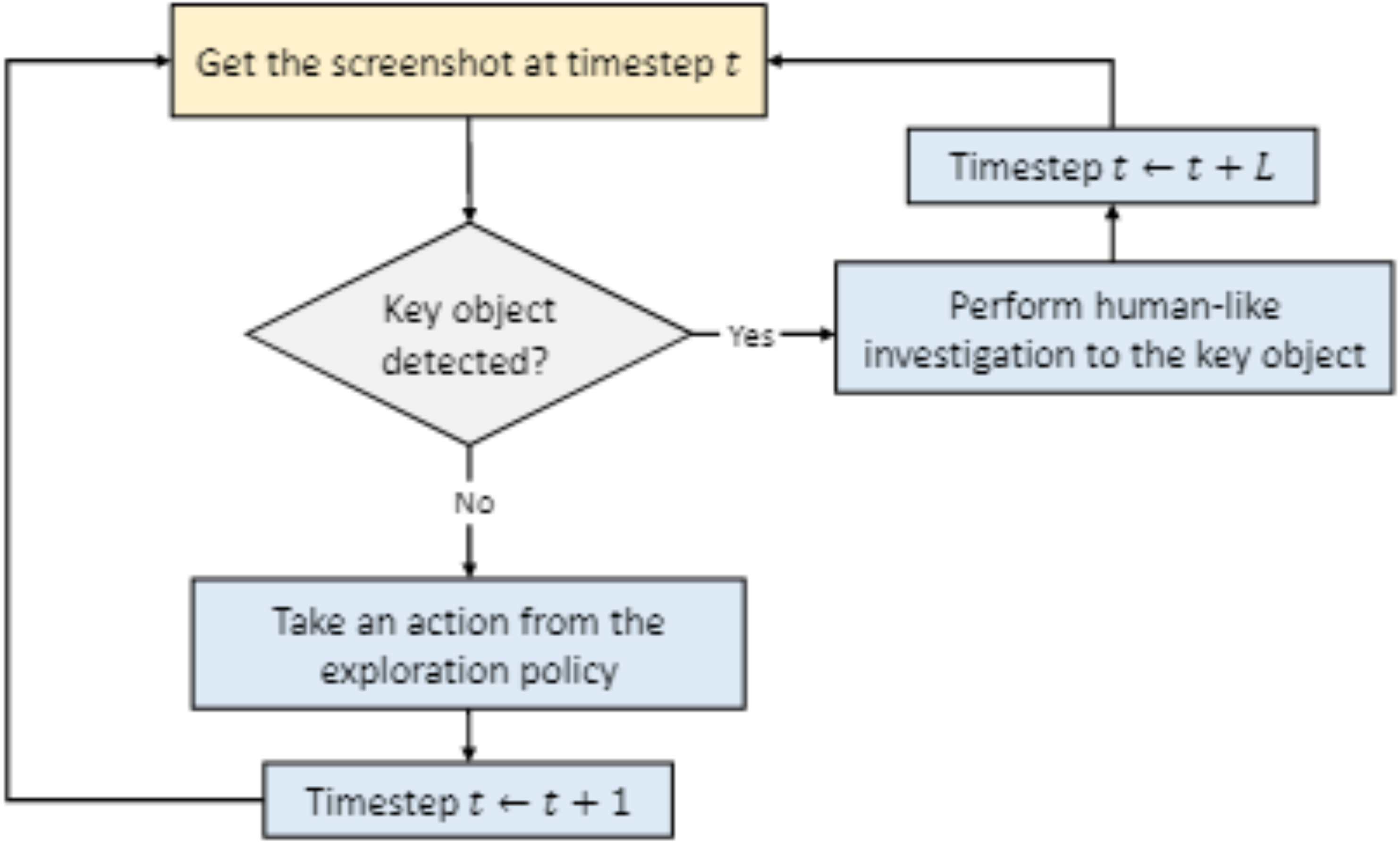
3) Human-like object investigator.

- To interact with key objects in a human like manner using **imitation learning**.
- Train a **Convolutional Neural Network** from human demonstrations by minimizing the behavior cloning loss

$$\mathcal{L}_{BC}(\pi_\theta) = -\sum_{(s,a)\in D} \log \pi_\theta(a|s),$$

Experiments
-human like object investigator-

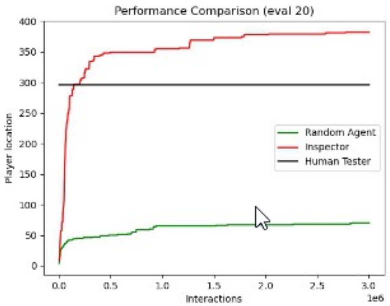
- the network structure
  - four convolution layers,
  - four fully connected layers,
  - a softmax layer.
- SGD optimizer with the learning rate of 0.001, and the batch size is 256.


Get the screenshot at timestep t
Timestep $t \leftarrow t + L$
Key object detected?
Yes
Perform human-like investigation to the key object
No
Take an action from the exploration policy
Timestep $t \leftarrow t + 1$

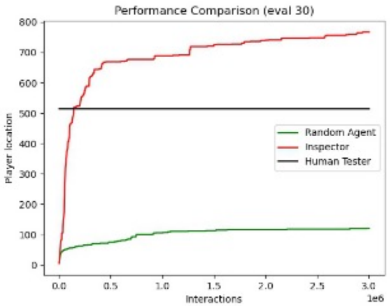
Experiments
-integrated system-

- the threshold of the bounding box size is 18000
- the threshold of the classification probability is 95%.

Result

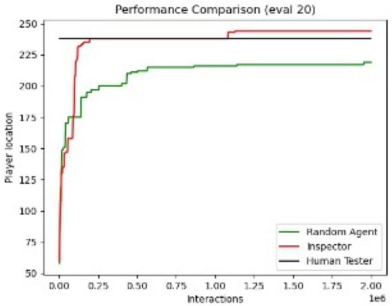


(a) $K = 20$

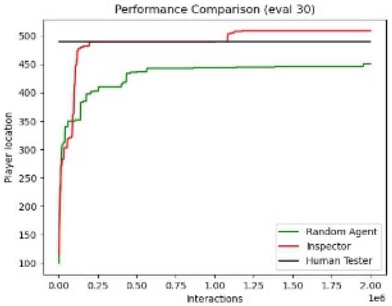


(b) $K = 30$

Fig. 5. The player location coverage results in Shooter Game. K represents the hyper-parameter for location discretization of the game map.



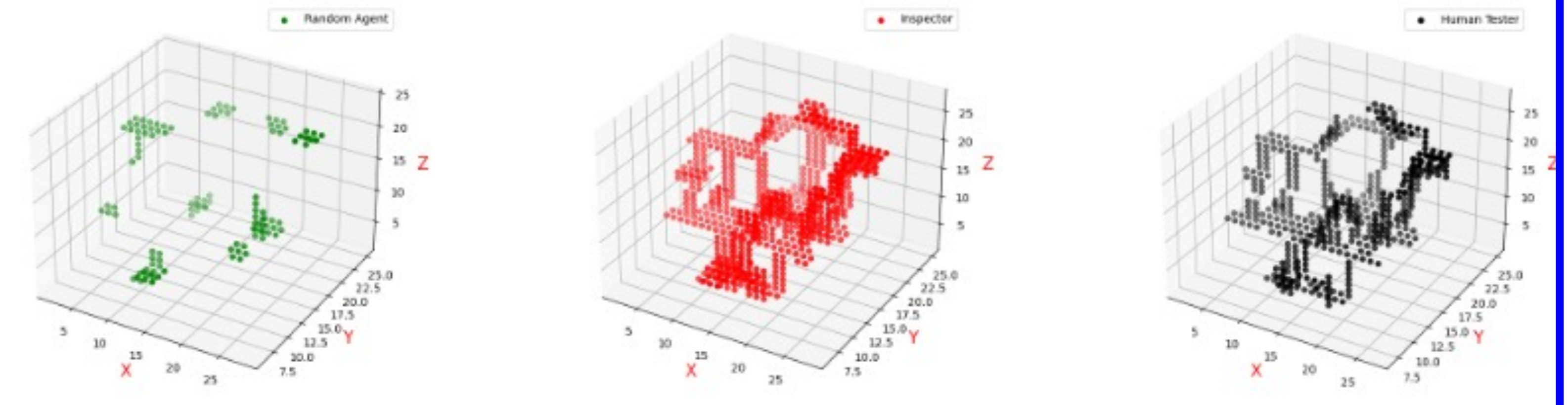
(a) $K = 20$



(b) $K = 30$

Fig. 7. The player location coverage results in Action RPG Game. K represents the hyper-parameter for location discretization of the game map.

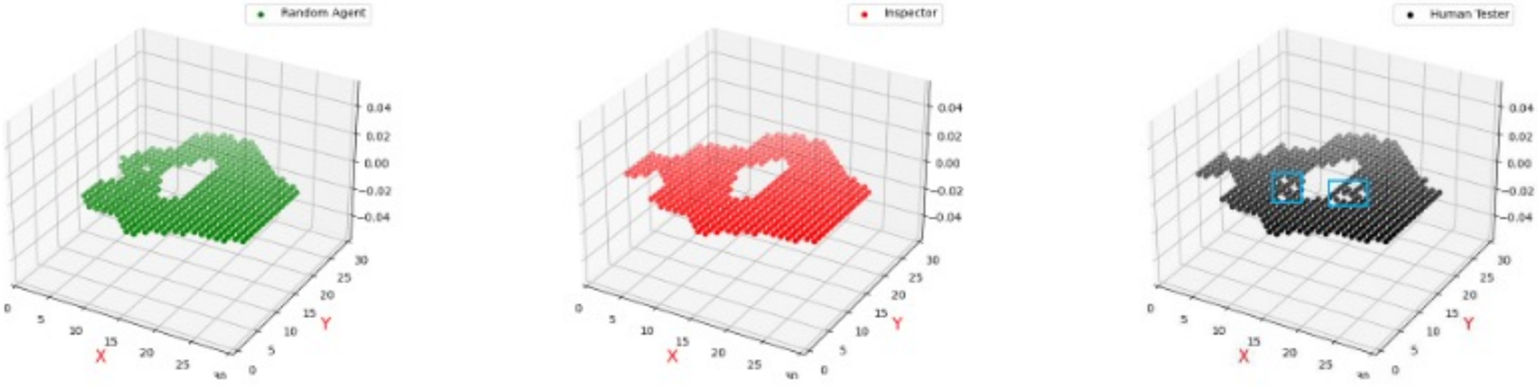
Result



(a) Random Agent (b) Inspector (c) Human Tester

Fig. 6. Visualization of the player location coverage results in Shooter Game.

Result



(a) Random Agent (b) Inspector (c) Human Tester

Fig. 8. Visualization of the player location coverage results in Action RPG Game

(a) 0/4 of a circle

(b) 1/4 of a circle

(c) 2/4 of a circle

(d) 3/4 of a circle

(e) 4/4 of a circle

Result

- Potential Bugs Discovered by Inspector

(a) An unimpressive corner in the Shooter Game map.

(b) Bug: the player stands without support under the feet.

Result

- Potential Bugs Discovered by Inspector

(a) Close to a rock in the Action RPG Game map.

(b) Bug: the player clips into the rock.

Conclusion

- Built a general pixel-based automated game testing agent named Inspector.
    - Larger application scope without the limitation of accessing game source code.
    - Ability to discover hidden and difficult bugs through human-like investigations with key objects.

Future work

- How about it in other game engines and across different devices.
- how to investigate more complex game.
- Further reduce human labeling costs .

Thanks!

Any questions?