

G-PY: A Game Playing AI To Simulate Real Time Close-Quarter Firefights Using 3D First- Person-Shooter Games

Adnan Ansari
Jaspreet Bhamra
Vishal Naidu



The Idea

The Idea

We believe that if an agent were to be created and trained such that it could successfully complete all missions in CS (an **ideal player**), then the experience that it would gain at the end would allow it to be used in defense applications



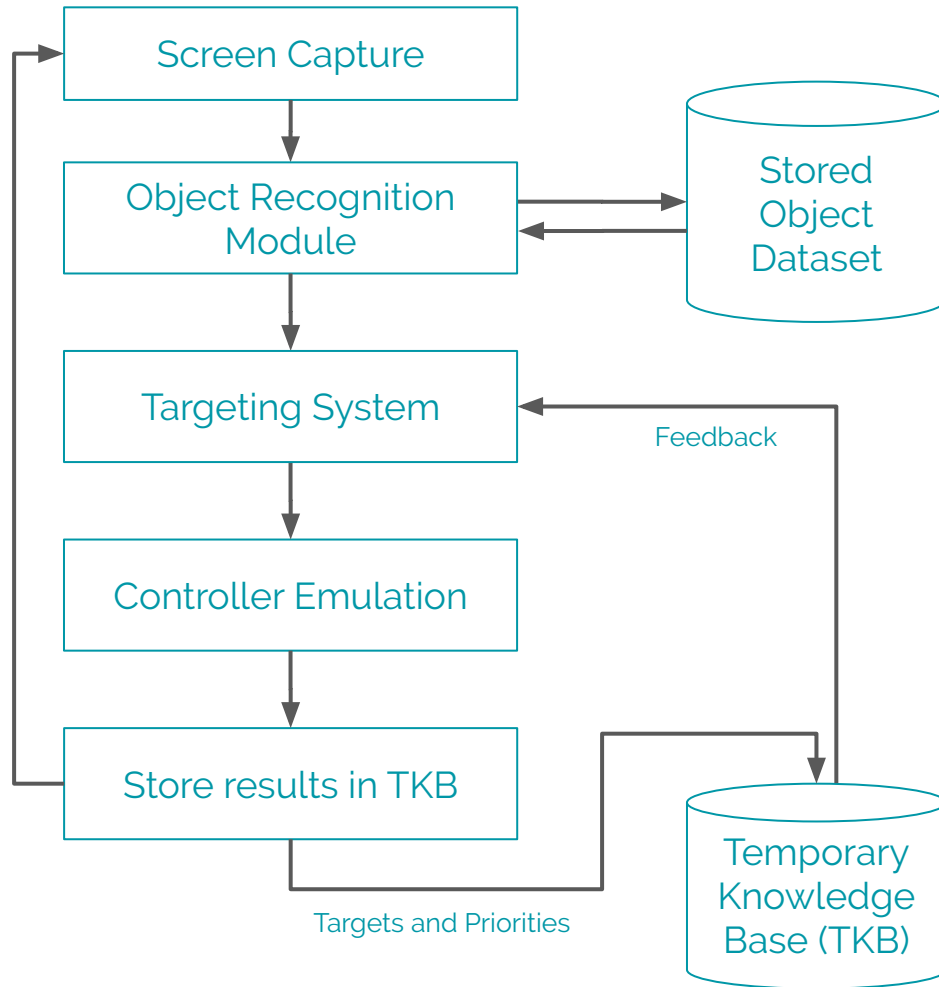
Assumptions

Assumptions

1. Enemies and friendlies have distinguishable attires
2. The agent can recognize objects from the frames
irrespective of the resolution of the captured frame
3. The agent can't see anything outside its field of view



System Architecture





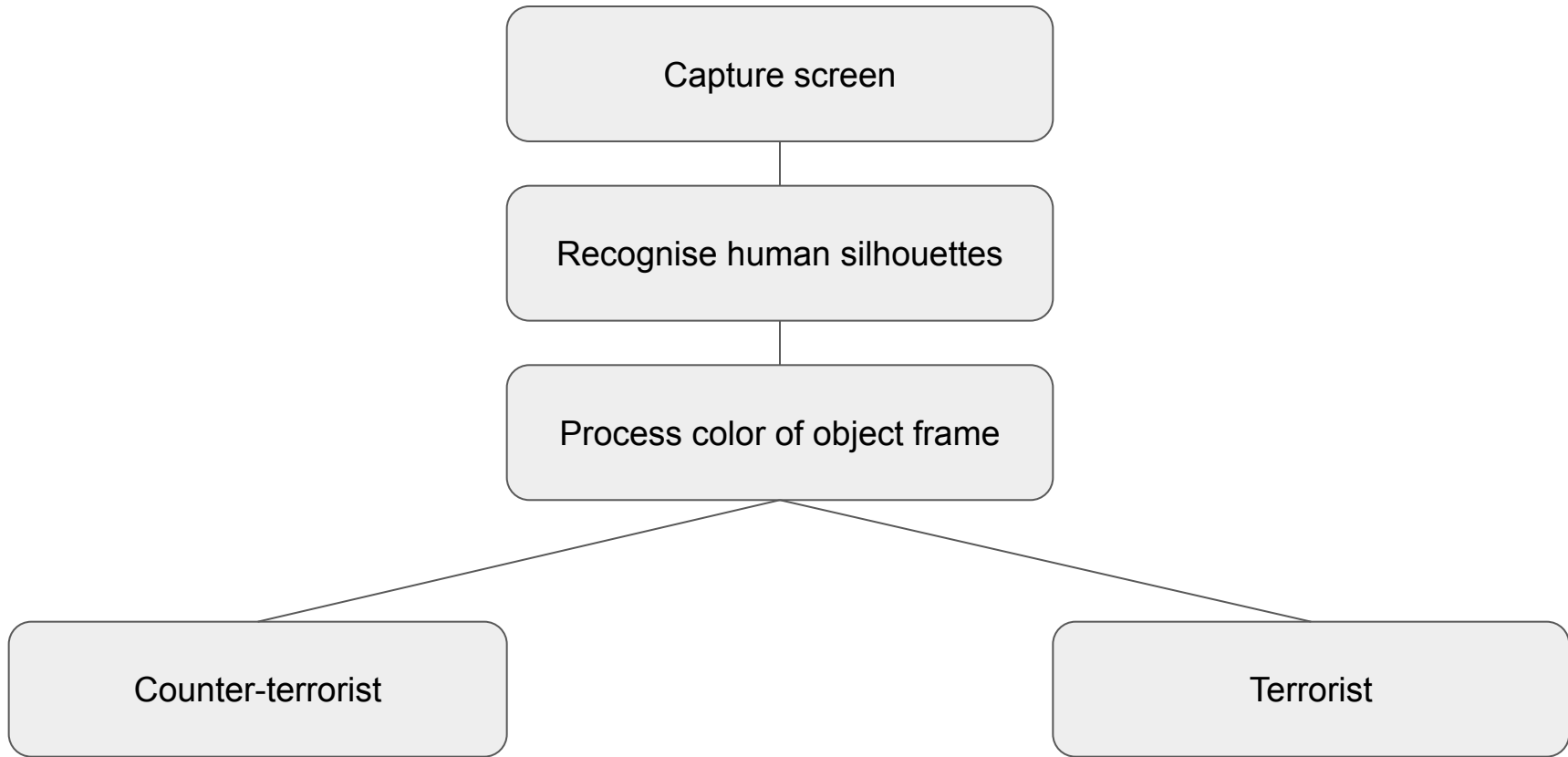
Screen Capture Module

Screen Capture Module

1. This module can be considered as the '**eyes**' of GPY
2. It captures pixels from a designated area on the screen
3. This is made possible using the python-windows native library **pywin32**.
4. An alternative to this can be **ImageGrab** from the library, Pillow.



Object Recognition Module





Positive samples (x1250)



Negative samples (x800)



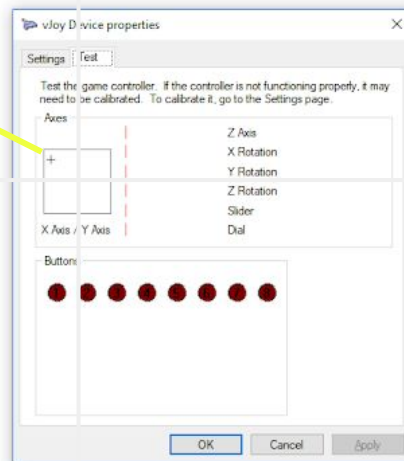
Classifier output



Controller Emulation



Chest-biased centering



Controller Monitor Tool
Axis

aligned with captured
screen centre

```
Windows PowerShell
controller.data.wAxisX = -100%
controller.data.wAxisY = -88%
----- 81
----- 82
----- 83
----- 84
----- 85
```



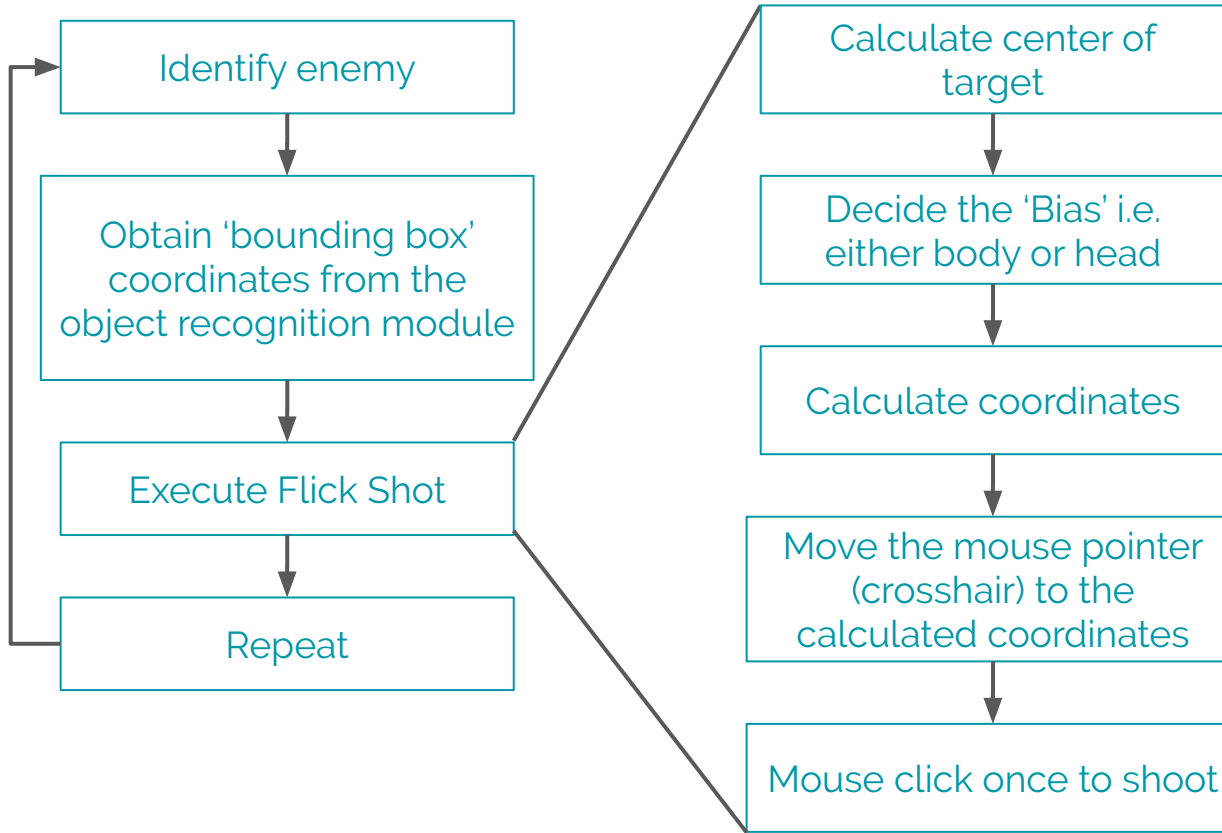
Flick Shot for Targeting

Flick Shot for Targeting

A Flick Shot is one of the ways of using semi-automatic rifles, not by tailing the target, but instead, by **flicking the sight** from the initial position to the target

The advantage of a flick shot is that it **compensates** for the time spent in **following the target** in the sight.

In order to compensate for a rapidly moving target, a **lead** has to be added to the destination position (**destination + lead_factor**)



Flick shot



Destination

Source





Memory based Targeting

Memory based Targeting

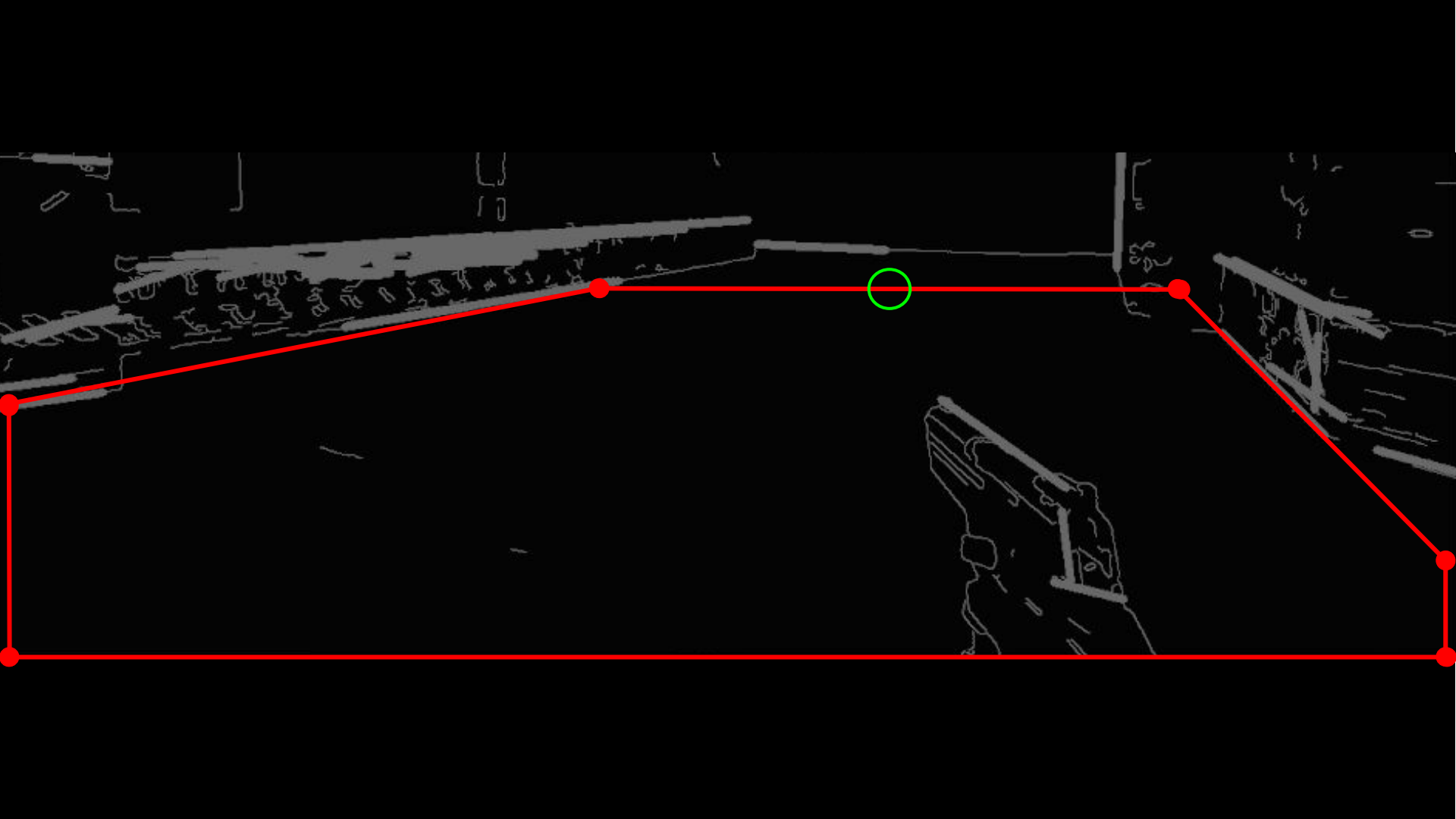
The flick shot would not be very accurate in cases when GPY is faced with **multiple enemies**.

In memory based targeting, the identified enemies will be assigned a **priority** based on some predefined rules.

This will allow the bot to eliminate those targets, that pose a **greater threat** to it, first.



Bot Navigation





Technology used

Technology Used

Screen Capture: The python-windows native library pywin32

Object Recognition: Haar Cascades from OpenCV

Controller Emulation: PyAutoGUI

Targeting Module: Image Processing using OpenCV



Social Relevance

Social Relevance

For defense & military: Prepare a BOT for combat without requiring an actual war environment

For game designers: Utilise the ideal behaviour to design and lay playing guidelines

For the general public: Utilise open-world games to learn safe driving by following all traffic rules, and then use it in self-driving cars



Thank You!