Human-Robot Interaction Interface in RoboCup SPL

Interface Overview:

- **Webnao:**
  - **Real-Time Location:** Continuously updates robot position and orientation.
  - **Control Panel:** commands issued immediately to robot.

Features:

- **Interactive Map:** Displays current location and bearing
- **Calibrations:** Provides kinematics and camera settings
- **Status Indicators:** Real-time feedback on battery life, connection status, and other critical parameters.

Input Devices:

- **Mouse and 'Keybaord'**

```python
#!/usr/bin/env python3
# this can be used either with a keyboard or with a gamepad
# to use a gamepad install inputs https://pypi.org/project/inputs/
# pip install inputs
from socket import *
import time
import sys
from Controller import *

DEFAULT_PORT = 2000

def printUsageMsg():
    print("Usage: python transmitter.py <nao_ip_addr> <port>")
    print("-k use keyboard input")

if __name__ == "__main__":
    if (len(sys.argv) < 2):
        printUsageMsg()
        sys.exit(1)
    useKeybaord = False
    if (sys.argv.count("-k") > 0):
        sys.argv.remove("-k")
        useKeybaord = True

    serverPort = DEFAULT_PORT

    if (len(sys.argv) > 2):
        serverPort = int(sys.argv[2])

    serverName = str(gethostbyname(sys.argv[1])) #accept both hostname and IPv4
    clientSocket = socket(AF_INET, SOCK_DGRAM) #create a socket, UDP use SOCK_DGRAM, TCP use SOCK_STREAM
    address = (serverName, serverPort)
    clientSocket.settimeout(1.0) #timeout to be one second

    # step size used for keyboard input
    stepSize = 5
    maxVelX = 300
    maxVelY = 300
    maxYaw = 1 #this may need tweaking: max angular velocity in rads/s
    turnStepSize = 0.1
    gameControllerDeadzone = 0.05 # 5% of stick

    speedController = KeyboardController(stepSize, turnStepSize, maxVelX, maxVelY, maxYaw)
    if (not useKeybaord):
        pads = inputs.devices.gamepads
        if (len(pads) == 0):
            print("There are no controllers connected... using keyboard")
        else:
            speedController = GamePadController(maxVelX, maxVelY, maxYaw, gameControllerDeadzone)

    speedController.displayUsageMsg()

    while (not speedController.exit()):

        speedController.update()
        msg = speedController.__str__()

        msgIsEmpty = not msg
        if not msgIsEmpty:
            #print("sending " + msg)
            clientSocket.sendto(bytes(msg, 'utf-8'), address)

    clientSocket.sendto(bytes(speedController.stopMsg(), 'utf-8'), address)

clientSocket.close()
```

Command Structure and Strategy in RoboCup SPL

Command Types:

- **Movement Commands:**
  - **Directional:** Move forward, backward
  - **Rotational:** Turn to specific angles

Action Commands:

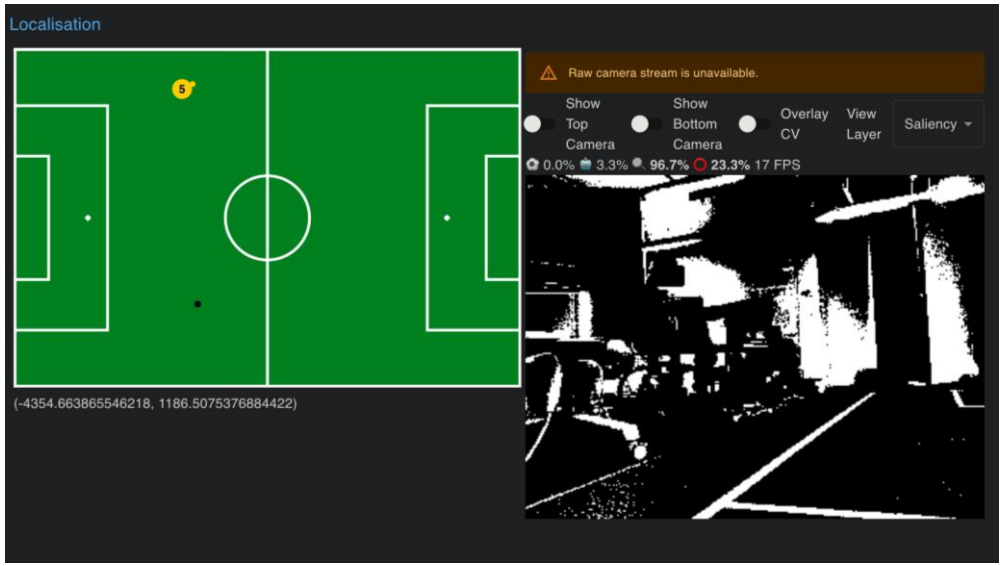
- **Kicking:** Execute kicks
- **Defensive Actions:** Block and position in defense
- **Custom Plays:** Execute specific plays

Autonomous Mode Activation:

- **Switching Modes:** Transition between manual and autonomous modes

Strategy for Coordination:

- **Integration:**
  - **Dynamic Roles:** Autonomous play during routine phases, human intervention during critical moments. Relies on WiFi TCP/IP protocol suite
  - **Situational Overrides:** Operator can override autonomous behavior to respond to unexpected scenarios.
  - **Real-Time Decision Making:** Combines operator with autonomous capabilities.



Webnao live localisation and realtime video stream over field WiFi (~20fps)