// State machine of “Capture The Flag”

class CaptureTheFlag {

sm{

MainThread {

enterMapParamters() -> WifiReceival;

}

WifiReceival {

entry / { intializeParameters(Map data);}

buttonPressed() [ centerButton == buttonPressed()] /

{startThreads();}->CourseTraversal;

}

CourseTraversal{

OdometerThread {

clock / {doOdometry();}-> OdometerThread;

}

||

OdometryCorrectionThread{

lineIsDetected()[isLocalizing == false]/{odo.setPos();}->

OdometryCorrectionThread;

}

||

OdometryDisplayThread{

clock/ {updateDisp(odo.getPos)} -> OdometryDisplayThread;

}

ControllerThread {

UltrasonicLocalization{

entry/{performUltrasonicLocalization();}

usLocalizationIsOver()->LightLocalization;

}

LightLocalization{

entry/{performUltrasonicLocalization();}

llLocalizationIsOver()/{navToZipline();}->

Navigating;

}

Navigating{

arrival() [atZipline == true]->ZiplineTraversal;

arrival()[completedZipline == true && findBlock == true &&

completedRiver == false]

-> BlockDetection;

arrival()[completedRiver == true]

-> End;

}

ZiplineTraversal{

entry/ {doZiplineTraversal();}

completeZipline()[completedZipline == true]/{navToBlock();} ->

Navigating;

}

BlockDetection{

blockFound()[isCorrectColor() == true]/{navToRiver();} -> Navigating;

}

}

}

End{}

}

}