

```

virtual void setup() {
    StateNode *wait=start=addNode(new WaitNode("Wait",this,bandit));
    StateNode *left=addNode(new PressNode("Left",this,LFrLegOffset+KneeOffset));
    StateNode *right=addNode(new PressNode("Right",this,RFrLegOffset+KneeOffset));
    StateNode *decide=addNode(new DecideNode("Decide",this,bandit,left,right));
    StateNode *recoverl=addNode(new OutputNode("\nBadPressLeft",this,std::cout,wait));
    StateNode *recoverr=addNode(new OutputNode("\nBadPressRight",this,std::cout,wait));
    left->addTransition(new SmoothCompareTrans<float>(wait,&state->pidduties[LFrLegOffset+RotatorOffset], ...
    right->addTransition(new SmoothCompareTrans<float>(wait,&state->pidduties[RFrLegOffset+RotatorOffset], ...
    wait->addTransition(new TimeOutTrans(decide,2000));
    left->addTransition(new TimeOutTrans(recoverl,1500));
    right->addTransition(new TimeOutTrans(recoverr,1500));
    //    recover->addTransition(new TimeOutTrans(decide,500));
    StateNode::setup();
}

```