鼠标拖动功能是MouseArea是drag.target属性。可以在控件内部添加MouseArea子控件来实现拖动效果

|  |  |
| --- | --- |
| //main.qml  import QtQuick 2.15  import QtQuick.Window 2.15  import QtQuick.Controls 2.15  import "./effect.js" as *Effect*  Window {  width: 640  height: 480  visible: true  title: *qsTr*("Drag")  Rectangle{  id: *rect*  x:*btn*.x  width: 400  height: 350  anchors.bottom: *btn*.top  color: "lightsteelblue"  }  Button{  id: *btn*  anchors.bottom: *parent*.bottom  x:50  anchors.bottomMargin: 20  width: 100  height:50  Rectangle{  color:"yellow"  anchors.fill: *parent*  Text {  id: *name*  text: *qsTr*("颜色")  anchors.centerIn: *parent*  }  }  onClicked: {  // rect.color = Effect.getColor()  var *rgb* = *Effect*.getRndColor()  *rect*.color = *Qt*.rgba(*rgb*.red,*rgb*.green,*rgb*.blue,1.0)//注意这个方法是所有参数的值必须在0-1之间，否则没有效果  }  MouseArea{  anchors.fill: *parent*  drag.target: *btn* //实现拖拽的效果  }  }  } | //effect.js  function *getColor*(){  let *colors*=["blue","deeppink","steelblue","lime","orange","yellow"];  var *index* = *Math*.round(*Math*.random()\*10)%6; //Math.random()产生0-1之间的随机数，需要\*10变为1-10之间  //console.log(index)  return *colors*[*index*];  }  function *getRangedRnd*(max){  return *Math*.floor(*Math*.random() \* *max*)  }  function *getRndColor*(){  let *r* = *getRangedRnd*(256)  let *g* = *getRangedRnd*(256)  let *b* = *getRangedRnd*(256)  *console*.log(*r*,*g*,*b*)  return {red: *r*/255,green: *g*/255,blue: *b*/255} //注意这个对象是所有参数的值必须在0-1之间，否则没有效果  } |

**注意如果有另外一个控件的坐标是工具被拖动的控件的坐标来计算的，它也会被拖动**