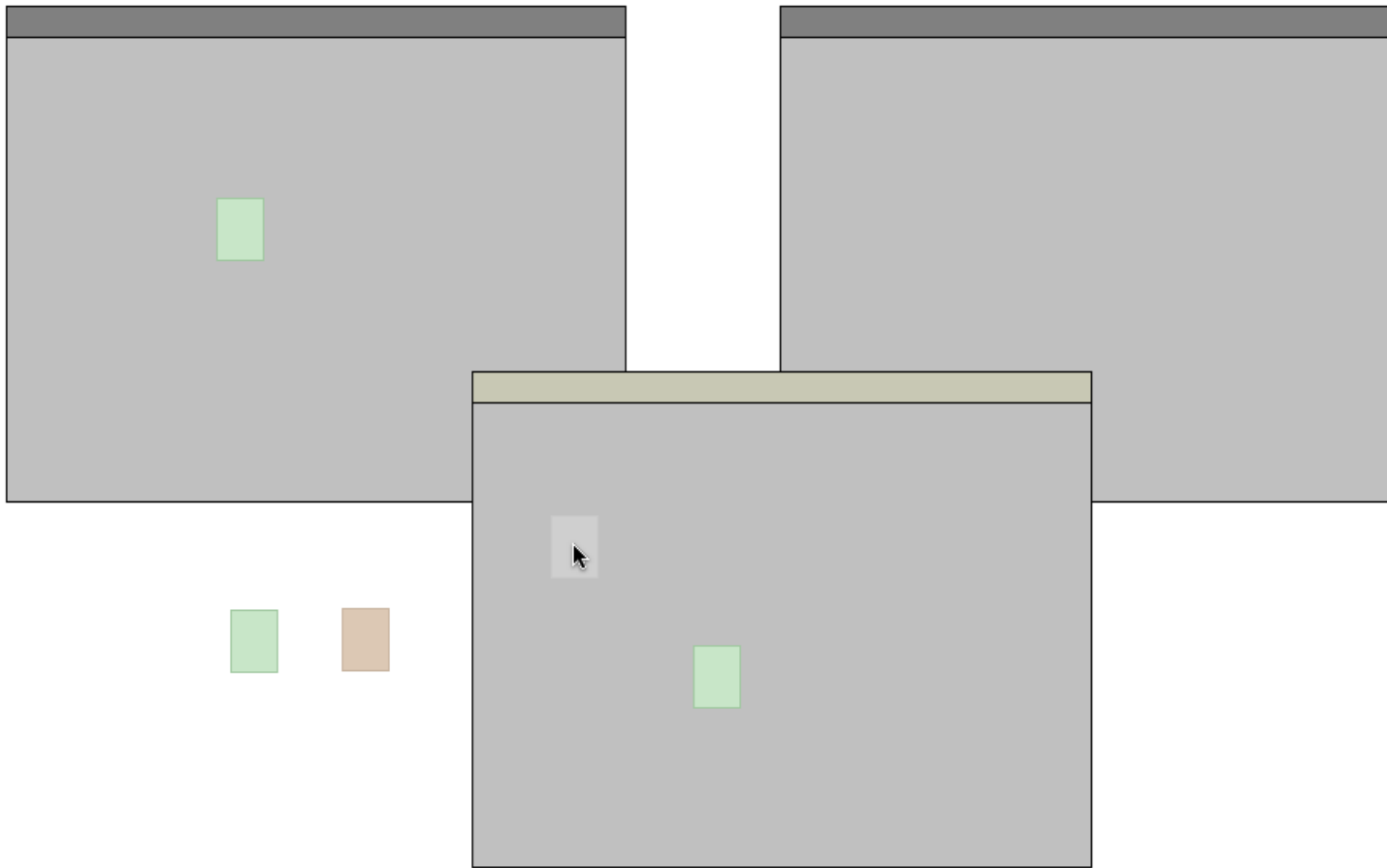


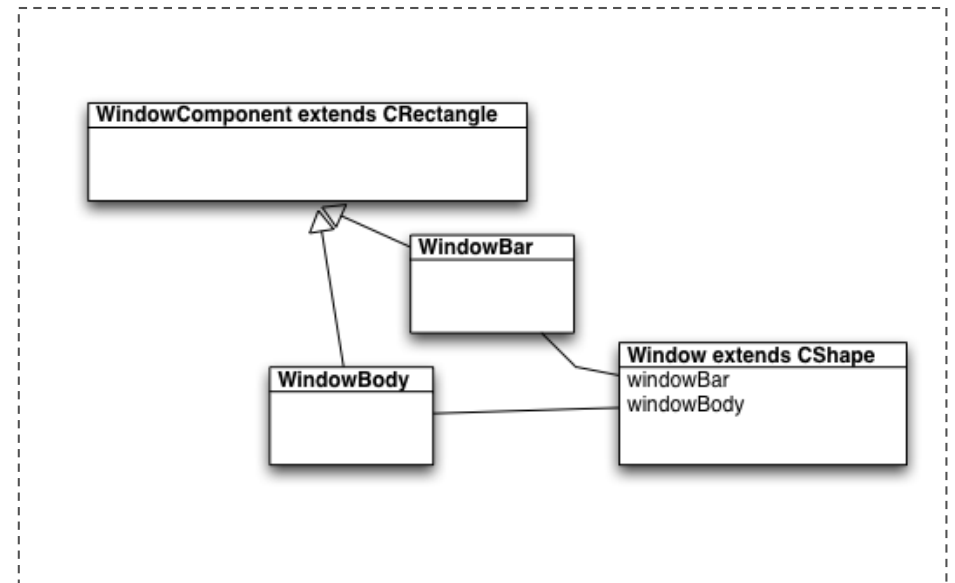
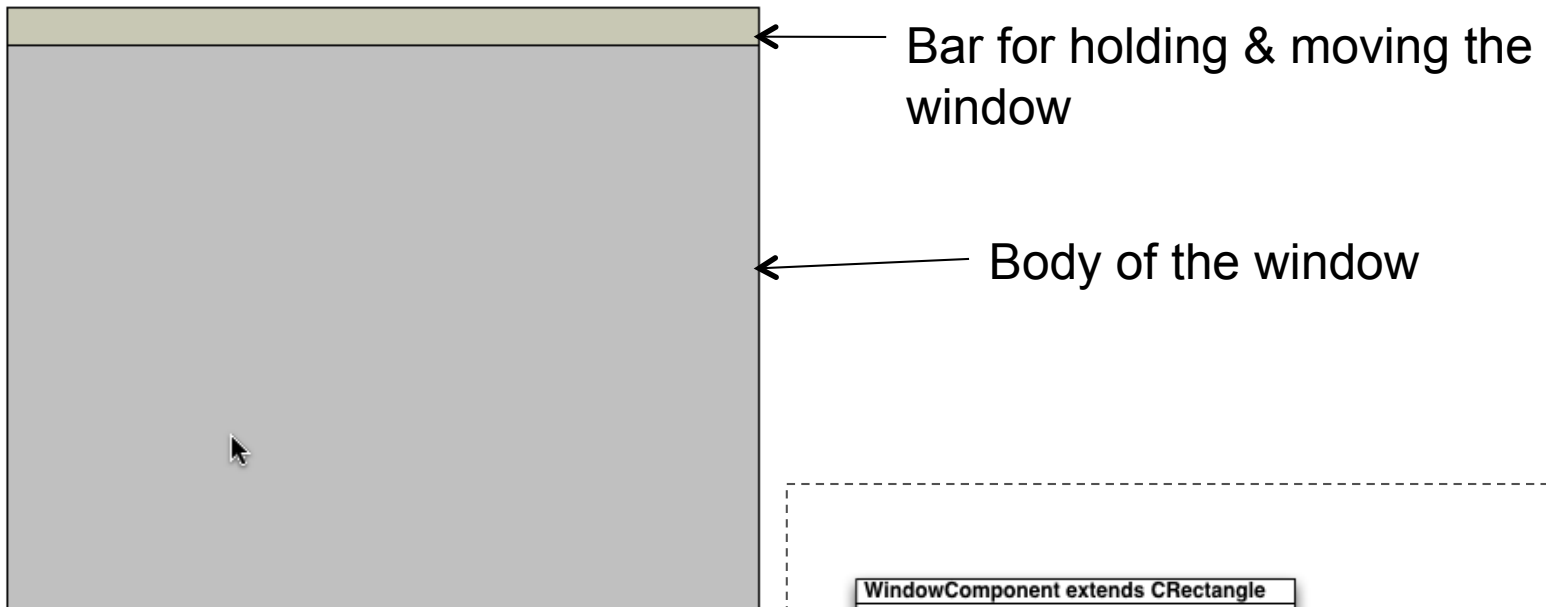
SwingStates – Example

Theophanis Tsandilas

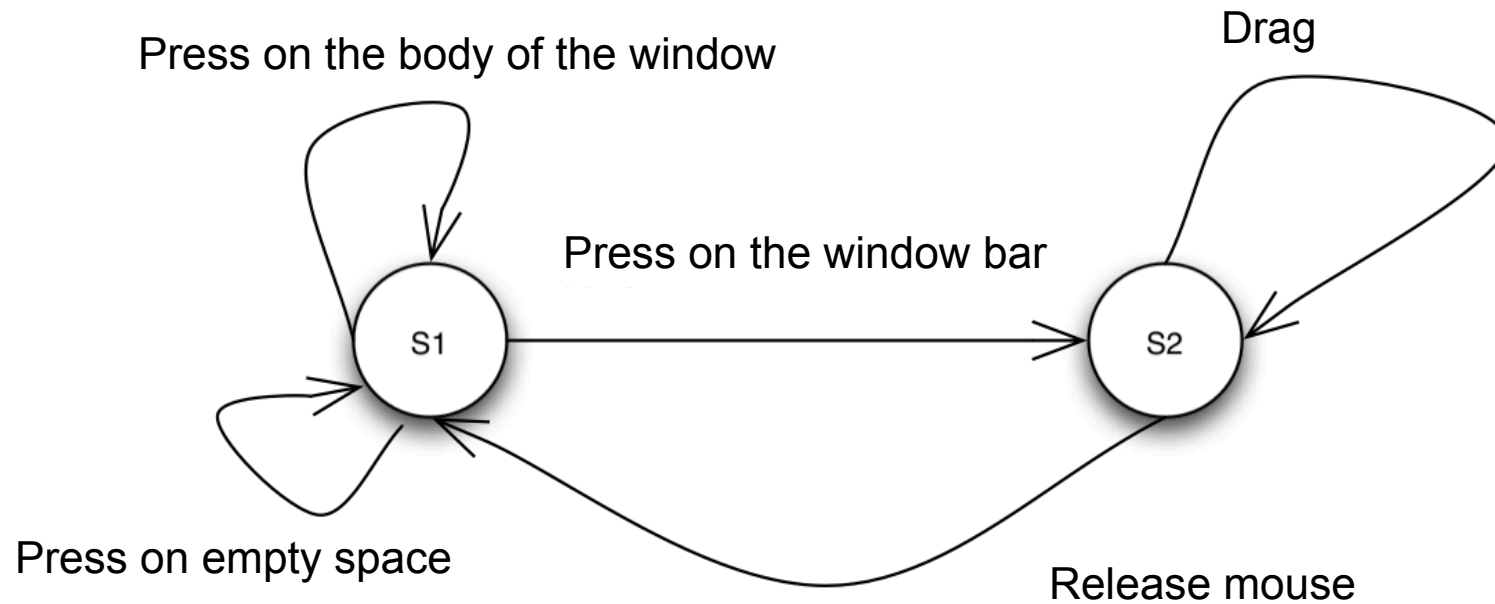
windows & drag-and-drop



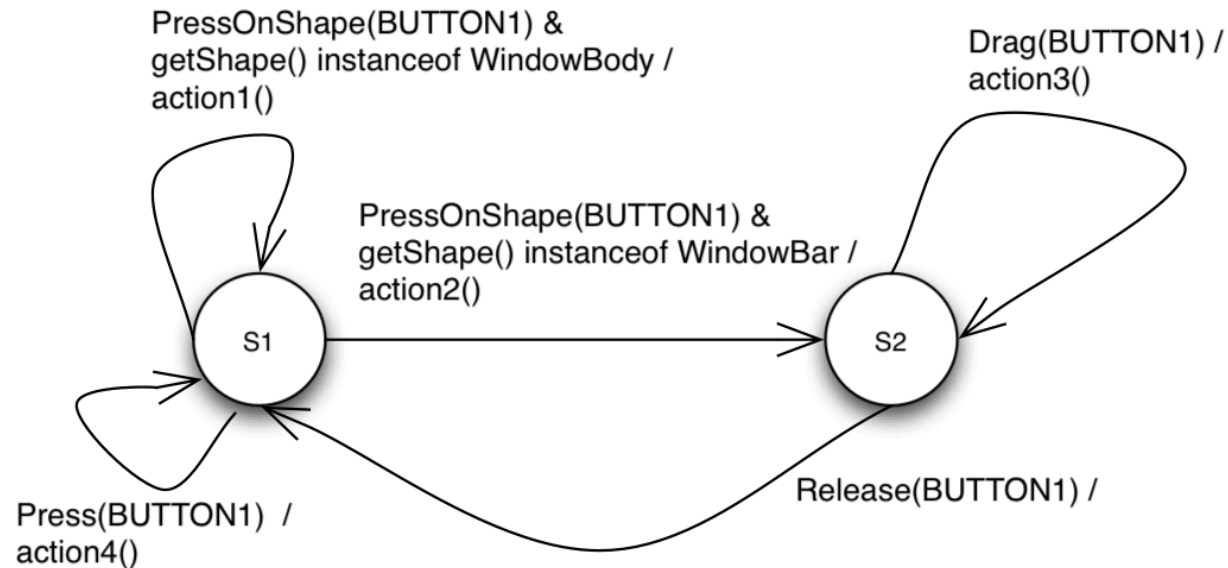
window



windows – state machine



windows – state machine



```
action1() {  
    WindowBody wbody = (WindowBody)getShape();  
    activateWindow(wbody.getWindow());  
}
```

```
action2() {  
    WindowBar wbar = (WindowBar)getShape();  
    activateWindow(wbar.getWindow());  
    p_ = getPoint();  
}
```

```
action3(){  
    Point2D p = getPoint();  
    activeWindow.translateBy(p.getX() - p_.getX(), p.getY() - p_.getY());  
    p_ = p;  
}
```

```
action4(){  
    if(activeWindow != null) activeWindow.setActive(false);  
    activeWindow = null;  
}
```

windows - SwingStates

```
CStateMachine m1 = new CStateMachine(){
    private Window activeWindow = null;
    private Point2D p_;

    public State s1 = new State(){

        // Press on the bar of the window
        Transition t11 = new PressOnShape(BUTTON1, "=> s2") {
            public boolean guard(){
                return (getShape() instanceof WindowBar);
            }

            public void action(){
                WindowBar wbar = (WindowBar) getShape();
                activateWindow(wbar.getWindow());

                p_ = getPoint();
            }
        };
        ...
    };
    ...
};
```

windows - SwingStates

```
CStateMachine m1 = new CStateMachine(){  
    private Window activeWindow = null;  
    private Point2D p_;
```

```
    public State s1 = new State(){
```

```
        // Press on the bar of the window
```

```
        Transition t11 = new PressOnShape(BUTTON1, "=> s2") {
```

```
            public boolean guard(){
```

```
                return (getShape() instanceof WindowBar);
```

```
            }
```

```
            public void action(){
```

```
                WindowBar wbar = (WindowBar) getShape();
```

```
                activateWindow(wbar.getWindow());
```

```
                p_ = getPoint();
```

```
            }
```

```
        };
```

```
        ...
```

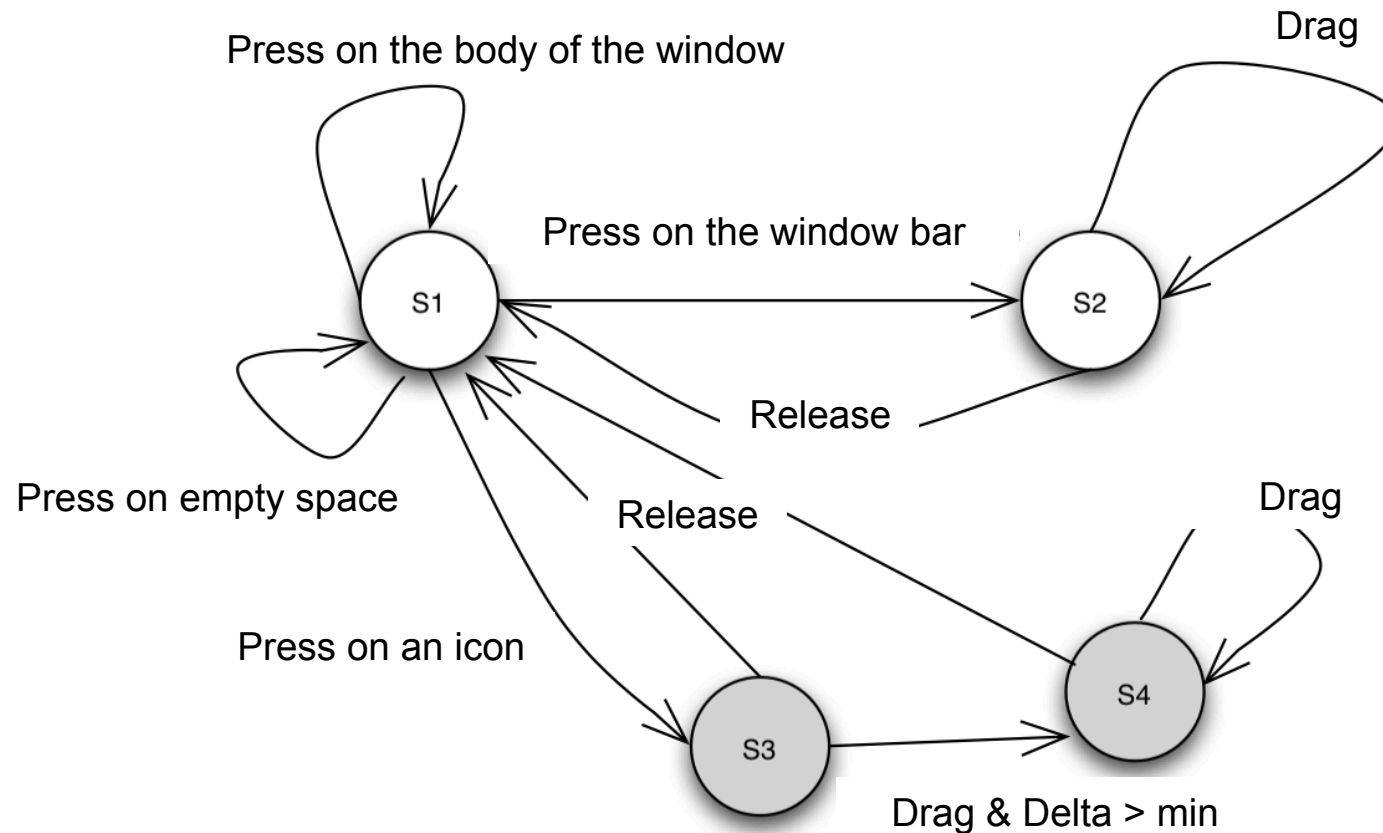
```
    };
```

```
    ...
```

```
};
```

guard condition for
transitions
(always evaluated)

complete interaction



Java code is available online