



State machines

Simple, Unconventional, Advanced

Andrei Zagorodni

2014-11-11

Content

- What is a state machine?
- Differently driven state machines
- UI-serving state machines
- OOP-based state machine
- Example: Let's develop a game

Good programming practice

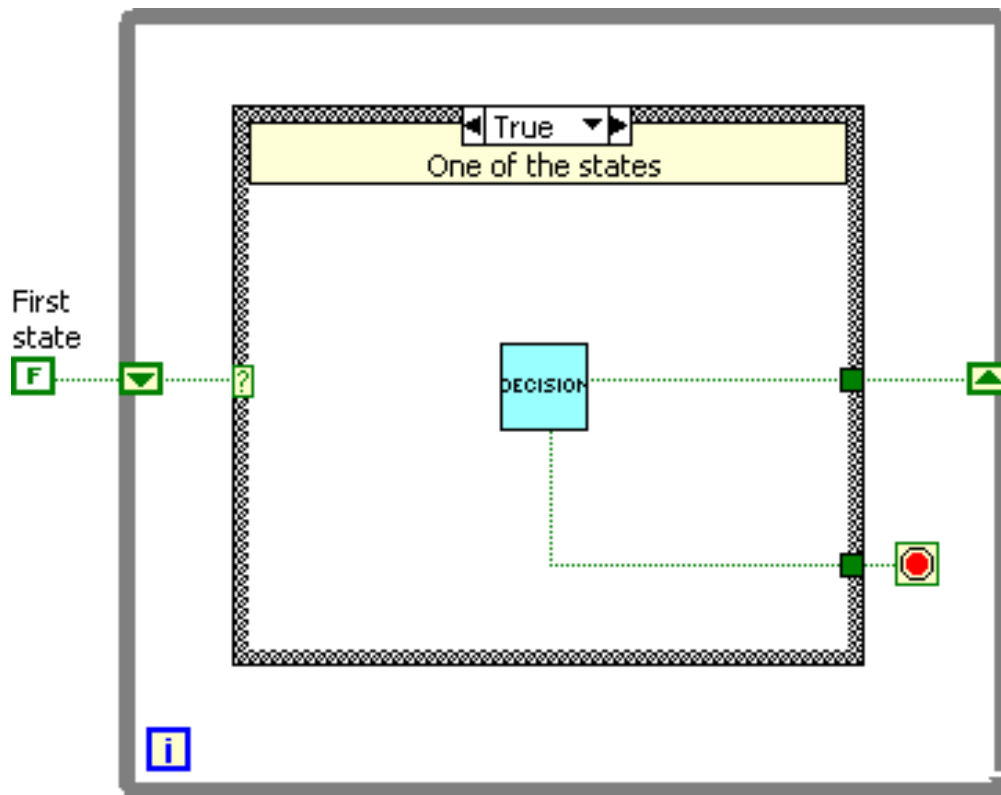
- Readability
- Maintainability
- Extendibility

But!

- Development time
- Quick and dirty solutions
- That customers love
- Until they ask for some changes

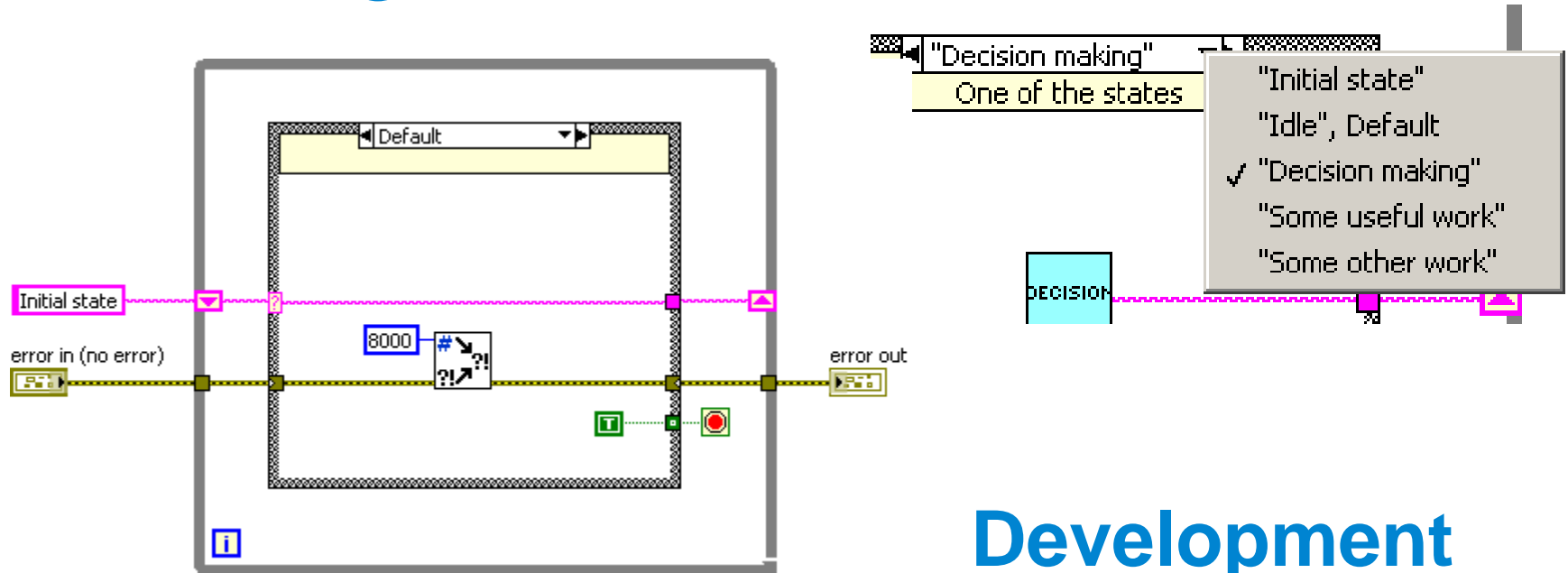
What is a state machine?

- A **While Loop** with a **Case Structure** driven with a **Shift Register**.
- At least on case of the Case Structure contains **decision-making code**.



- Must the loop be **While Loop**?
Almost
- Do we always need **Case Structure**?
Not really
- Should **Shift Register** be always used?
A Queue could be useful
- Is **decision making** an essential part of each state machine?
Strictly speaking yes, but...

String-driven state machines



Development

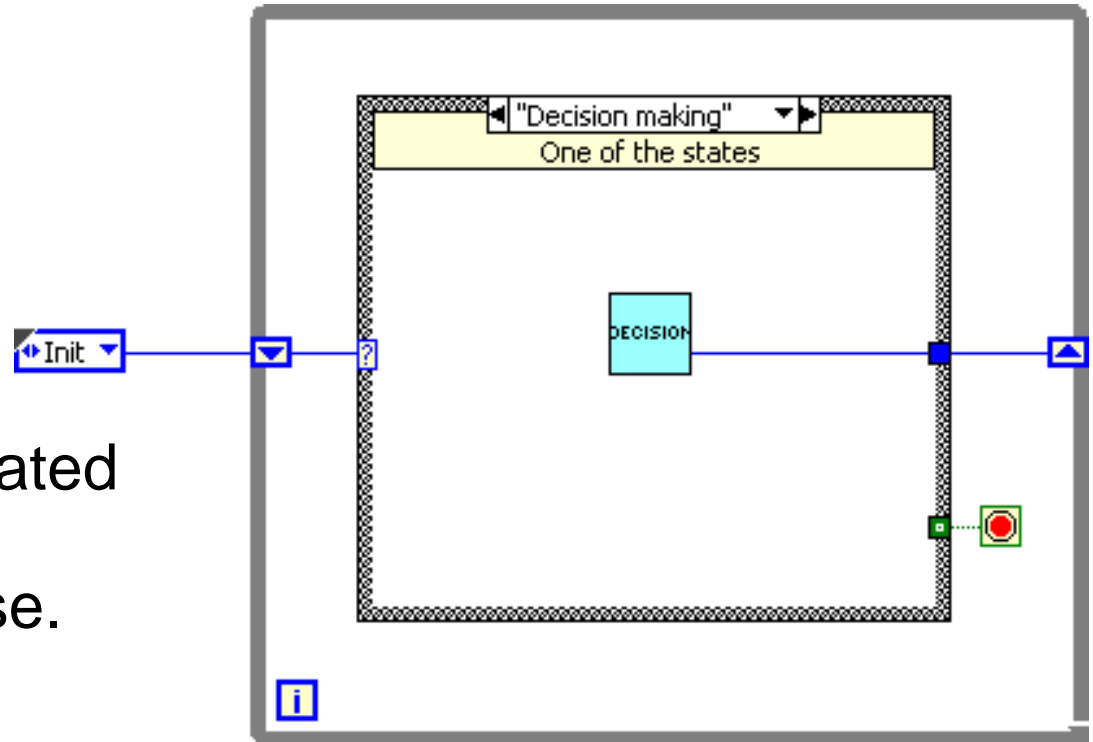
- Really quick
- Quite dirty

Do not use similar strings!

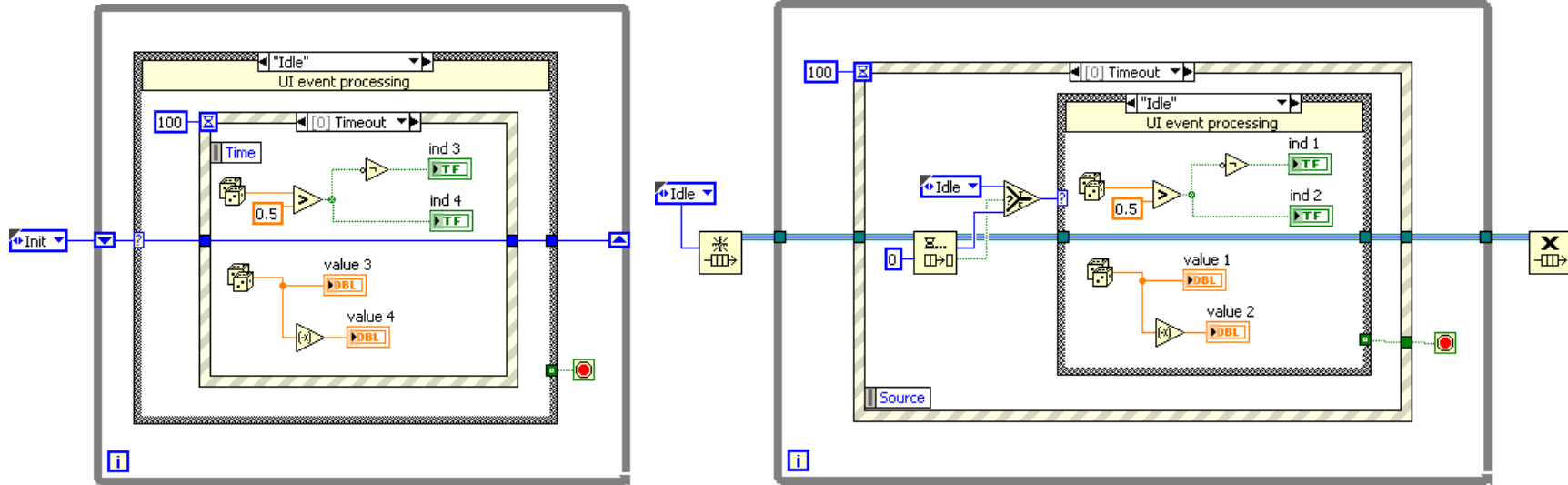
~~Init Sate 1 start~~
~~Inif Sate 7 stop~~

Enumerated-driven state machines

- Best possible solution
- Always use **Type defined** enumerated
- Never use **Default** case.

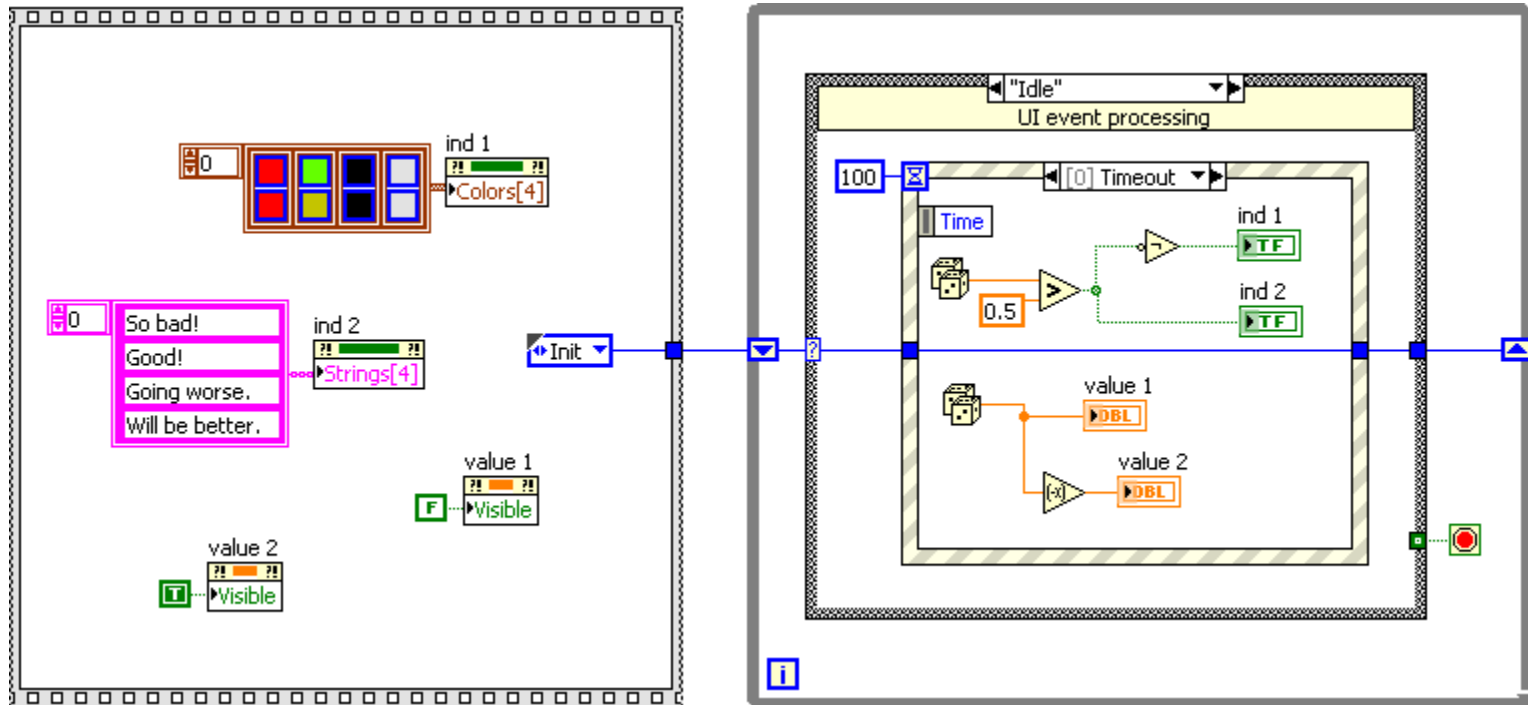


UI-serving state machines



- State machines unified with UI.
- Choice depends on priority.
- What about initialization steps?
- Specific problem of UI
Large block diagrams

Initialising UI state machines



- Avoid creating a block diagram that occupies more than one screen.

UI-serving state machines

VI-s with complicated UI always create style problems

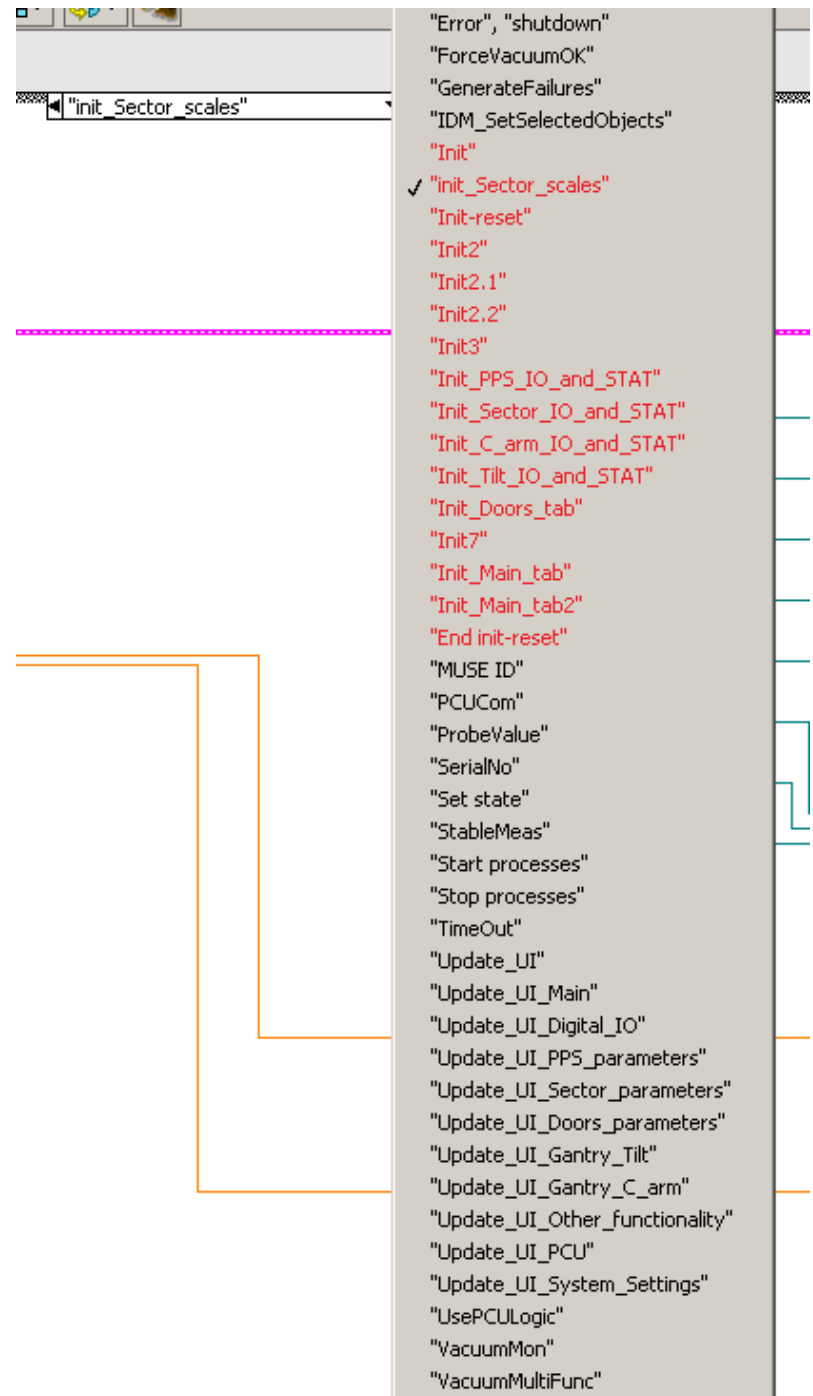
- Always conflict with the rule:

Block diagram must not exceed screen size

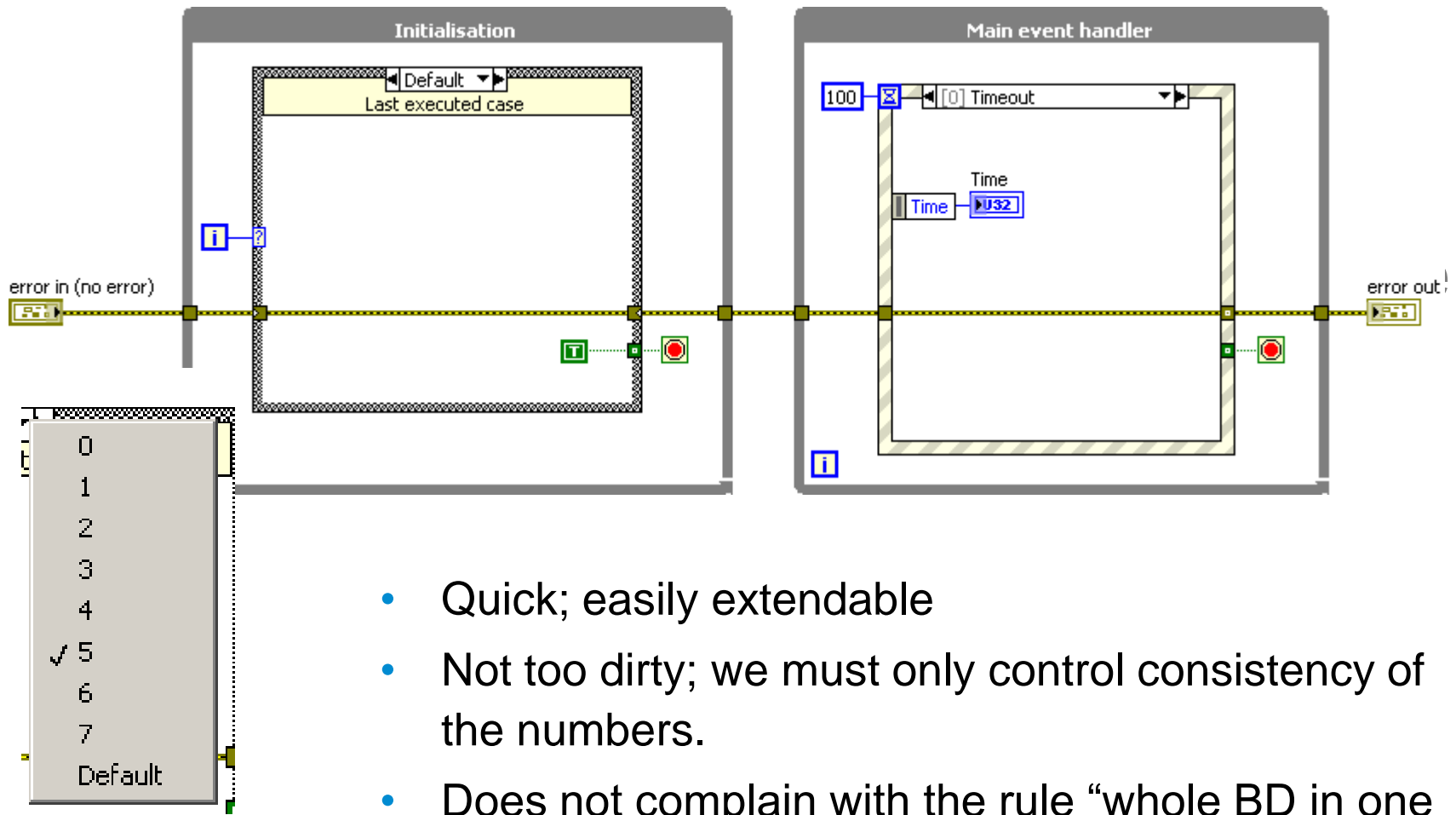
- Large number of states

Desirable solution

- Minimum effect of bad style
- Easy to implement

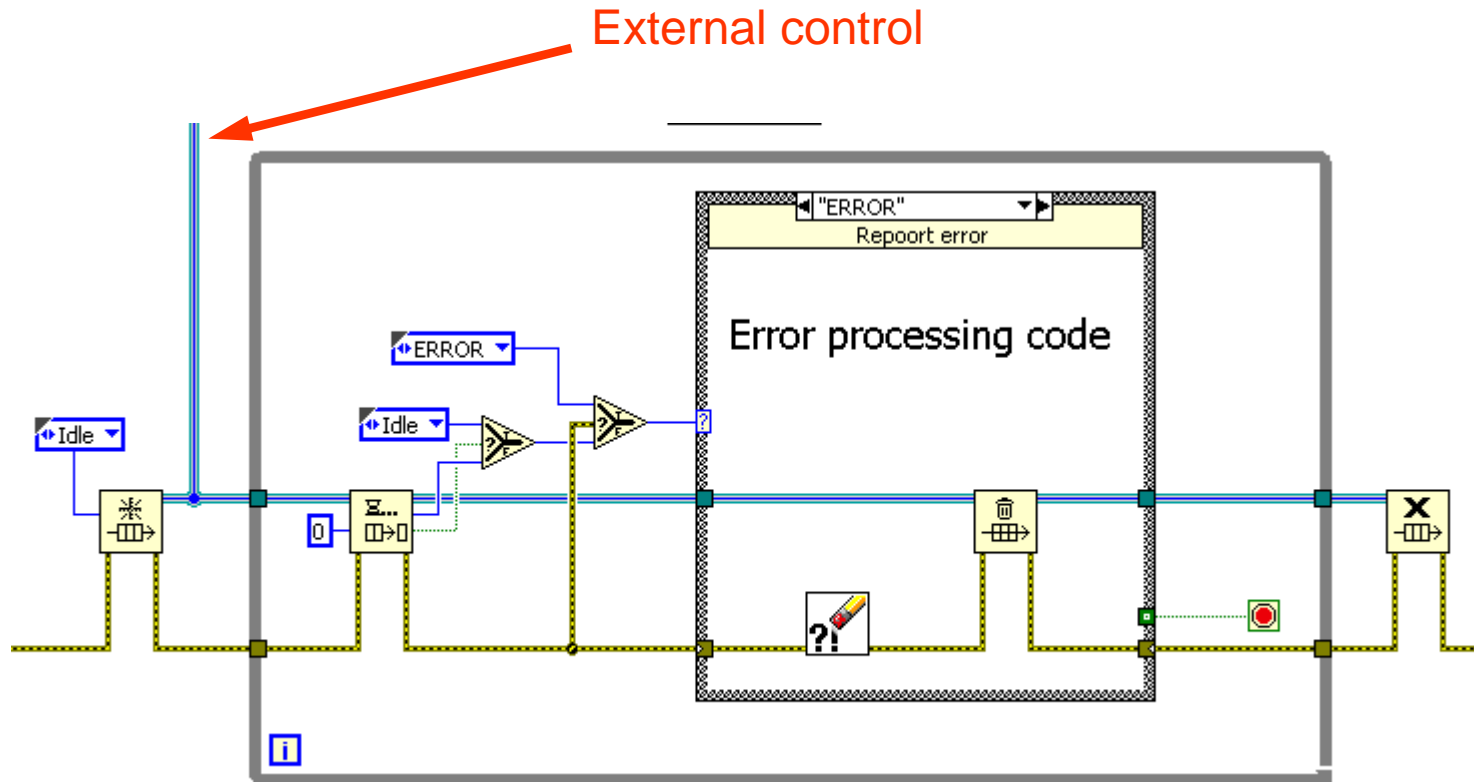


Number-driven state machines

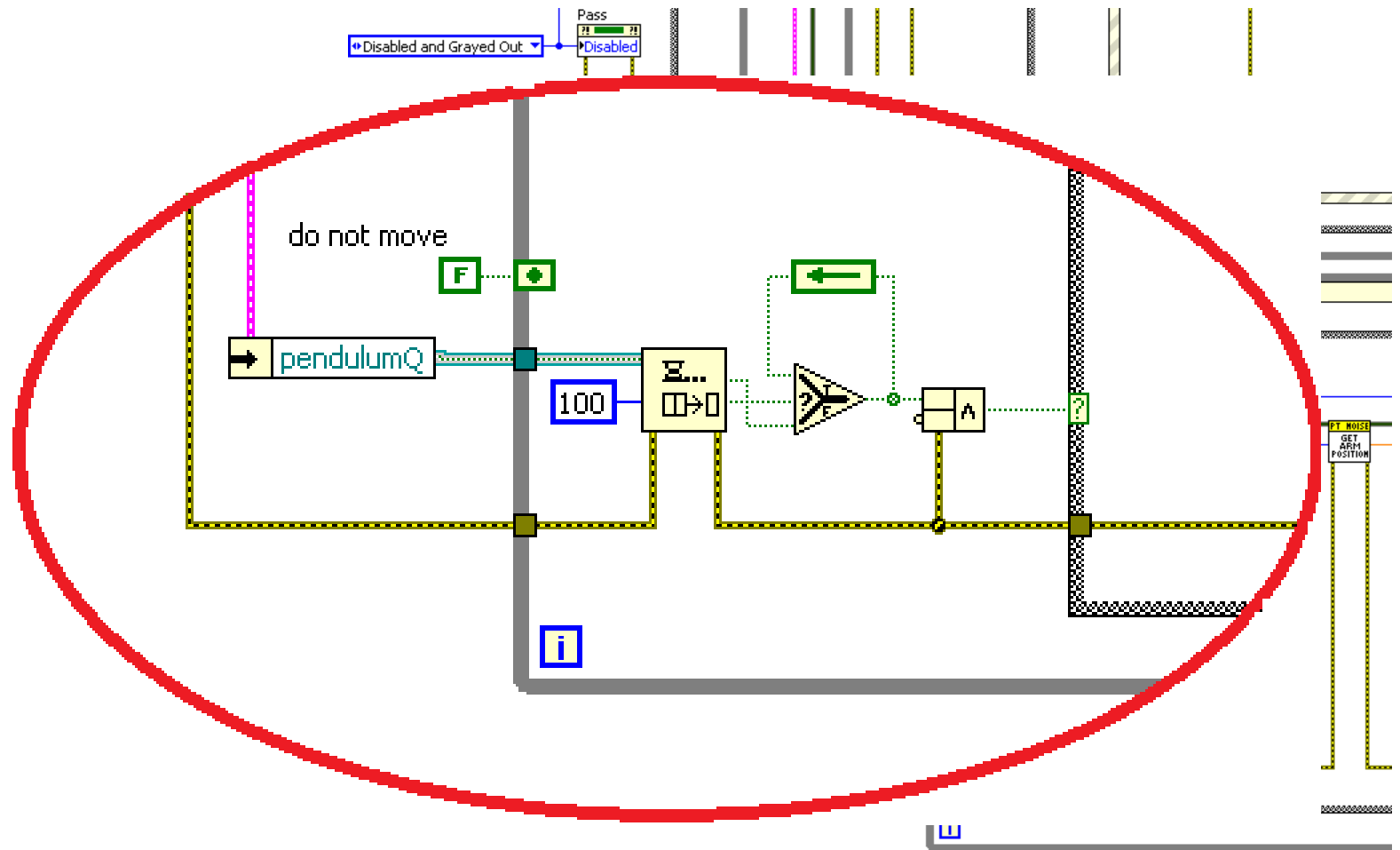


- Quick; easily extendable
- Not too dirty; we must only control consistency of the numbers.
- Does not complain with the rule “whole BD in one screen”. But...

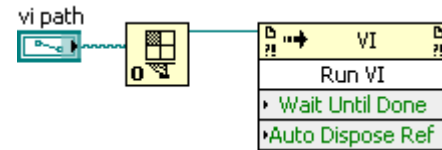
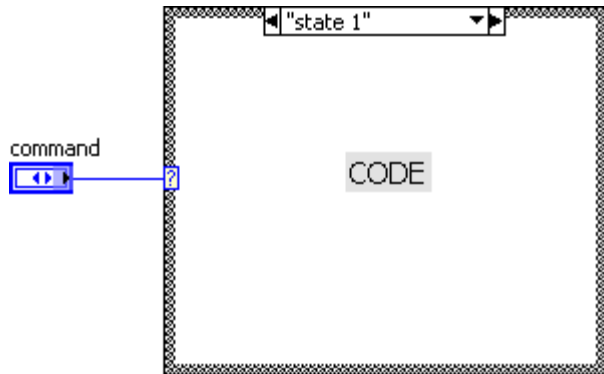
Queue-controlled state machine



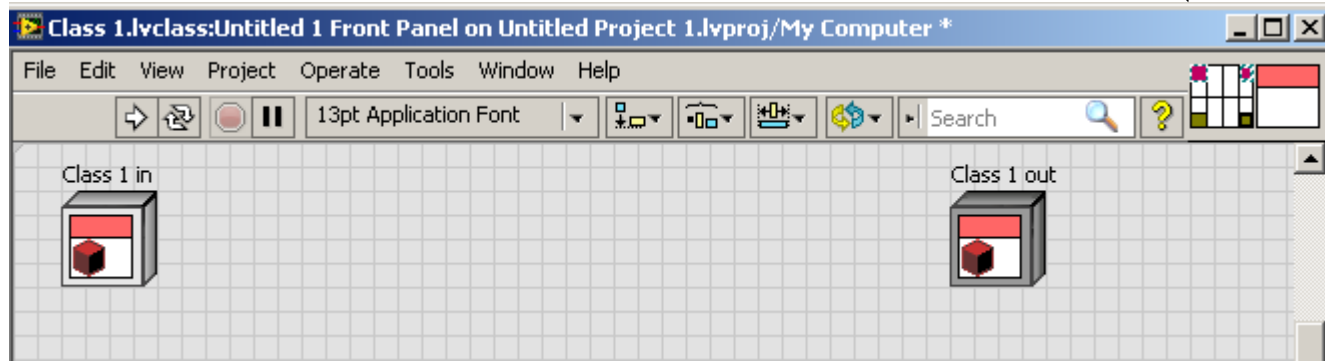
Boolean-driven state machine



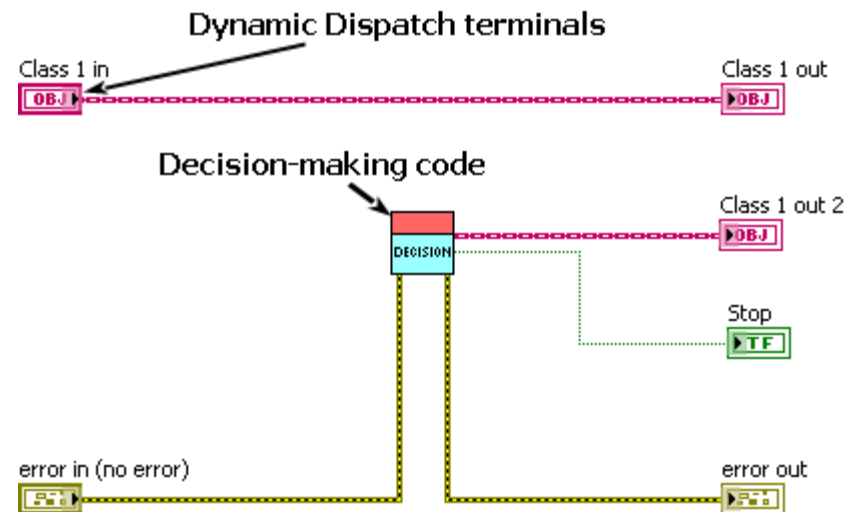
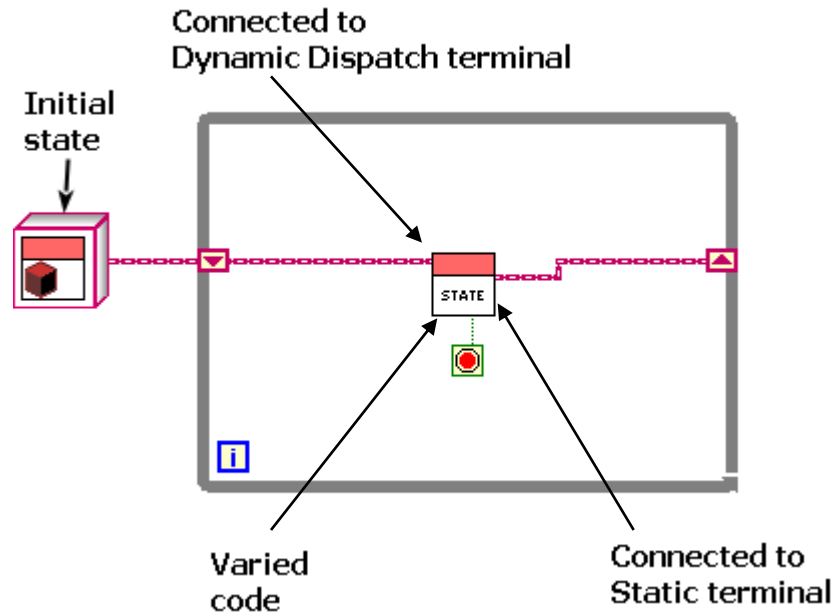
Code selectors



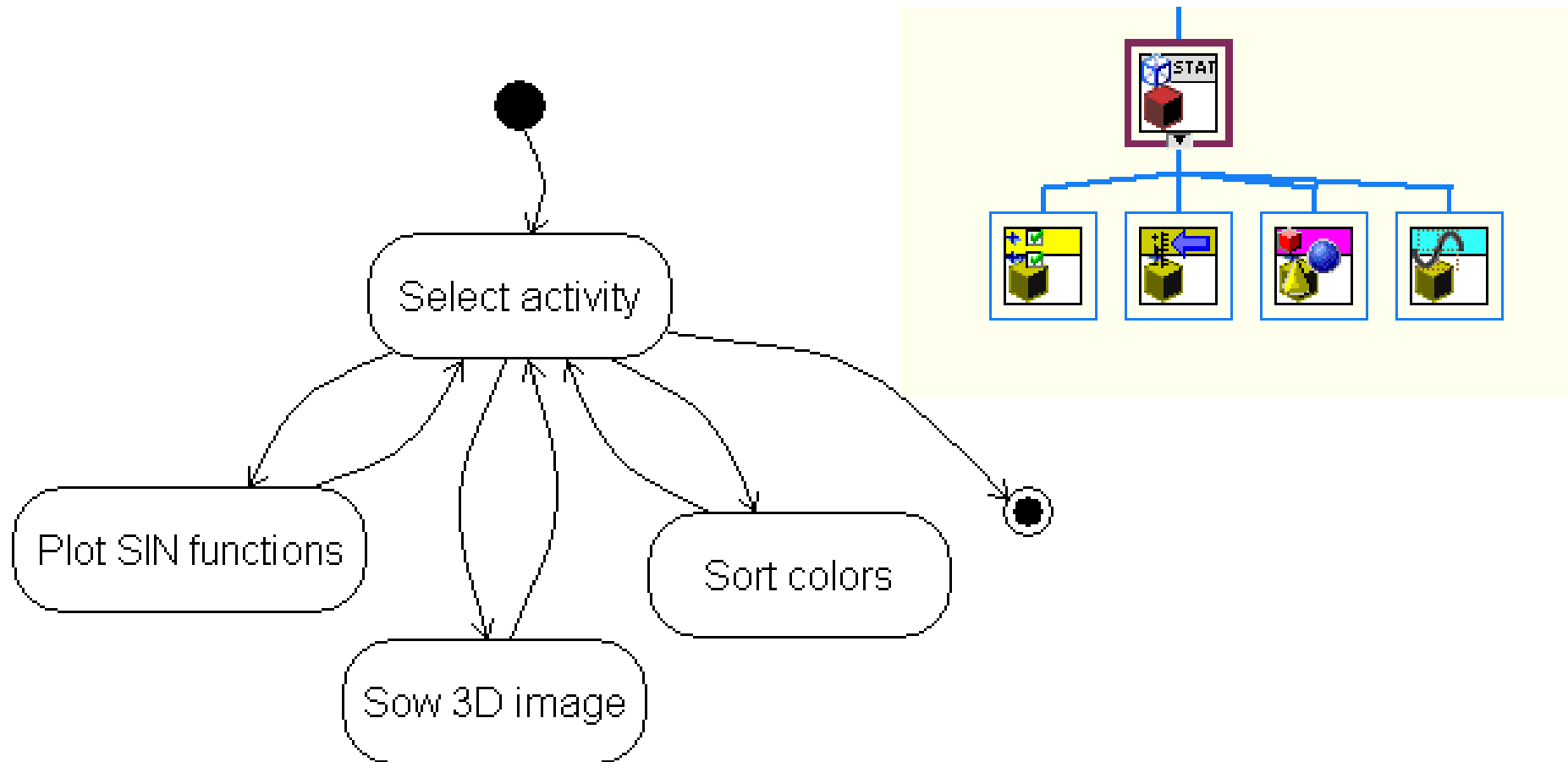
Dynamic Dispatch Terminals



OOP-based State Machine



Example of OOP-based State Machine



Let's Play Games





Thank you!