

Finite State Machines

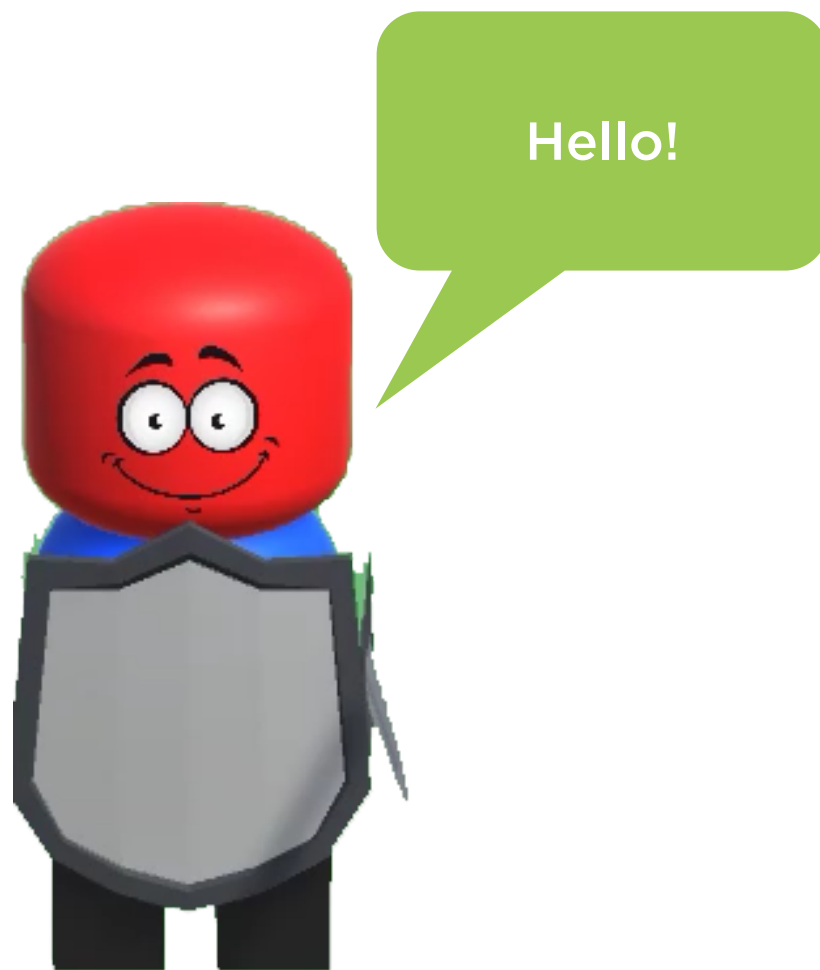


Marc Gilbert

FRIVOLOUSTWIST, LLC

@frivoloustwist www.frivoloustwist.com





Behavior

Actions

Triggers

Conditions



Focus on state!



States of Matter



Solid

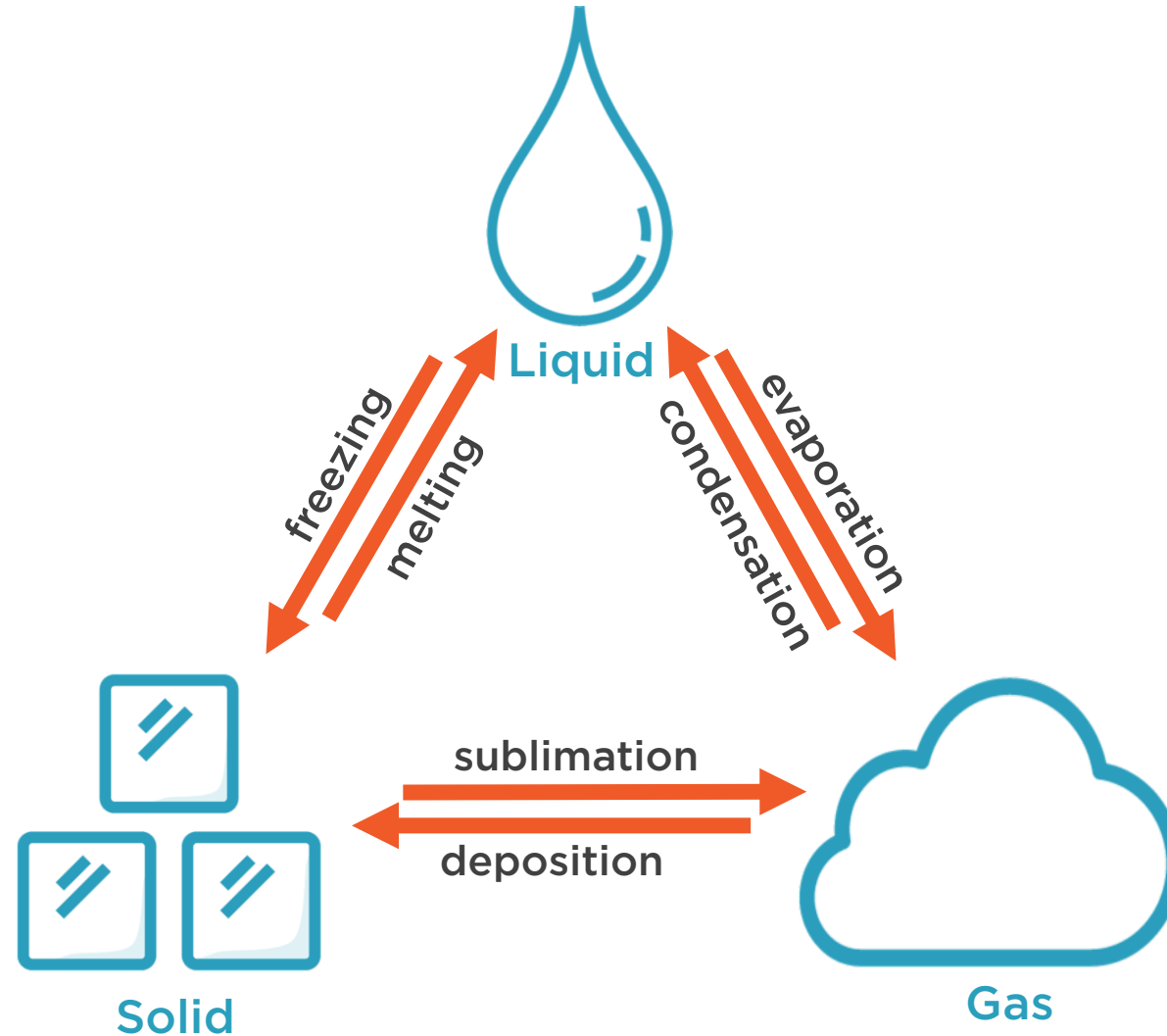


Liquid

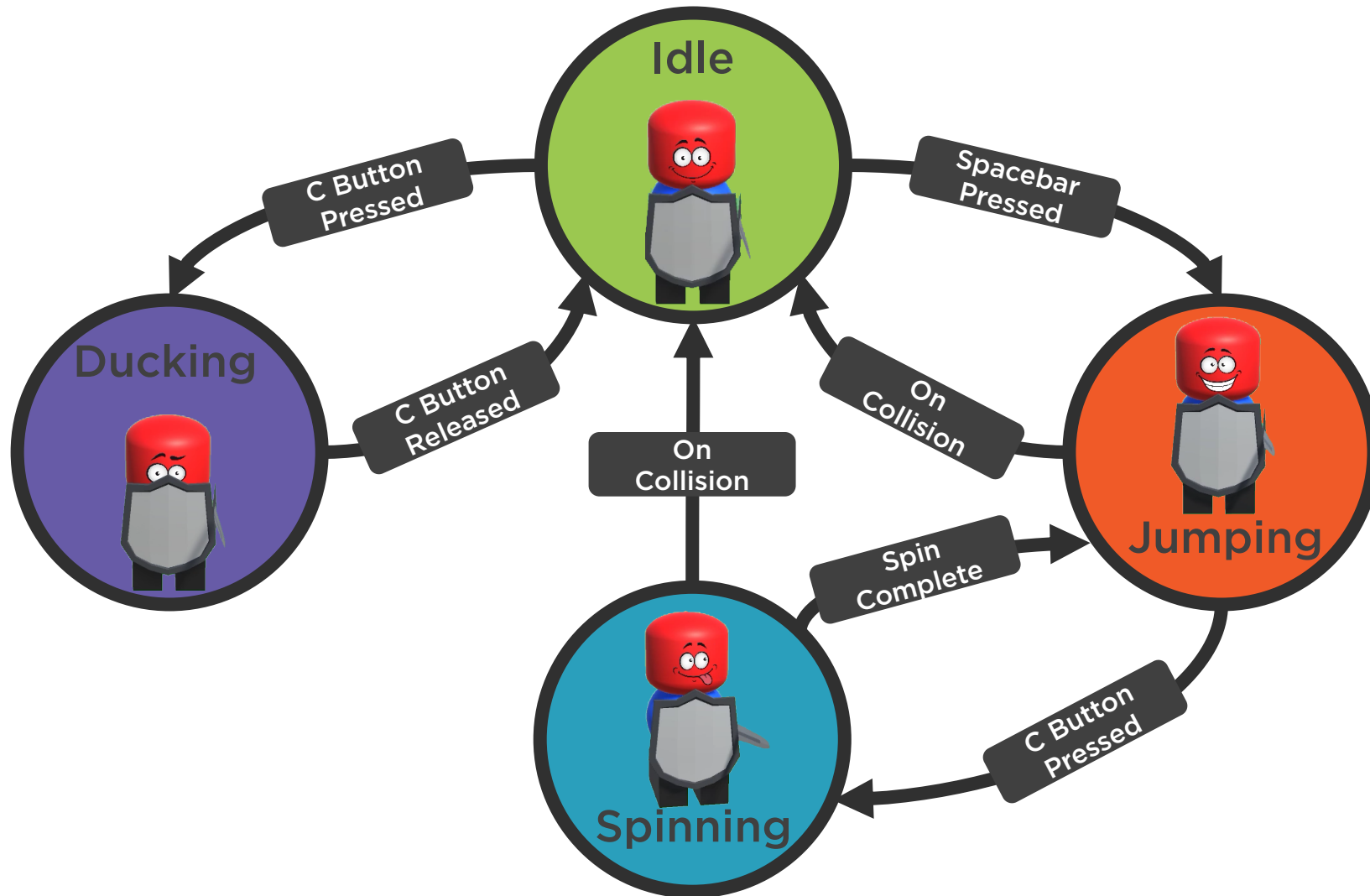


Gas

Matter as a Finite State Machine



A Finite State Machine for Arthur



Benefit of a Finite State Machine



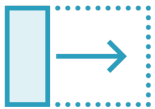
Easier to read



Less difficult to maintain



Easier to debug



More extensible



Finite State Machine

A Finite State Machine is an **abstract** machine that can be in exactly one of a finite number of states at any given time.



What is a Finite State Machine?



A list of possible **states**

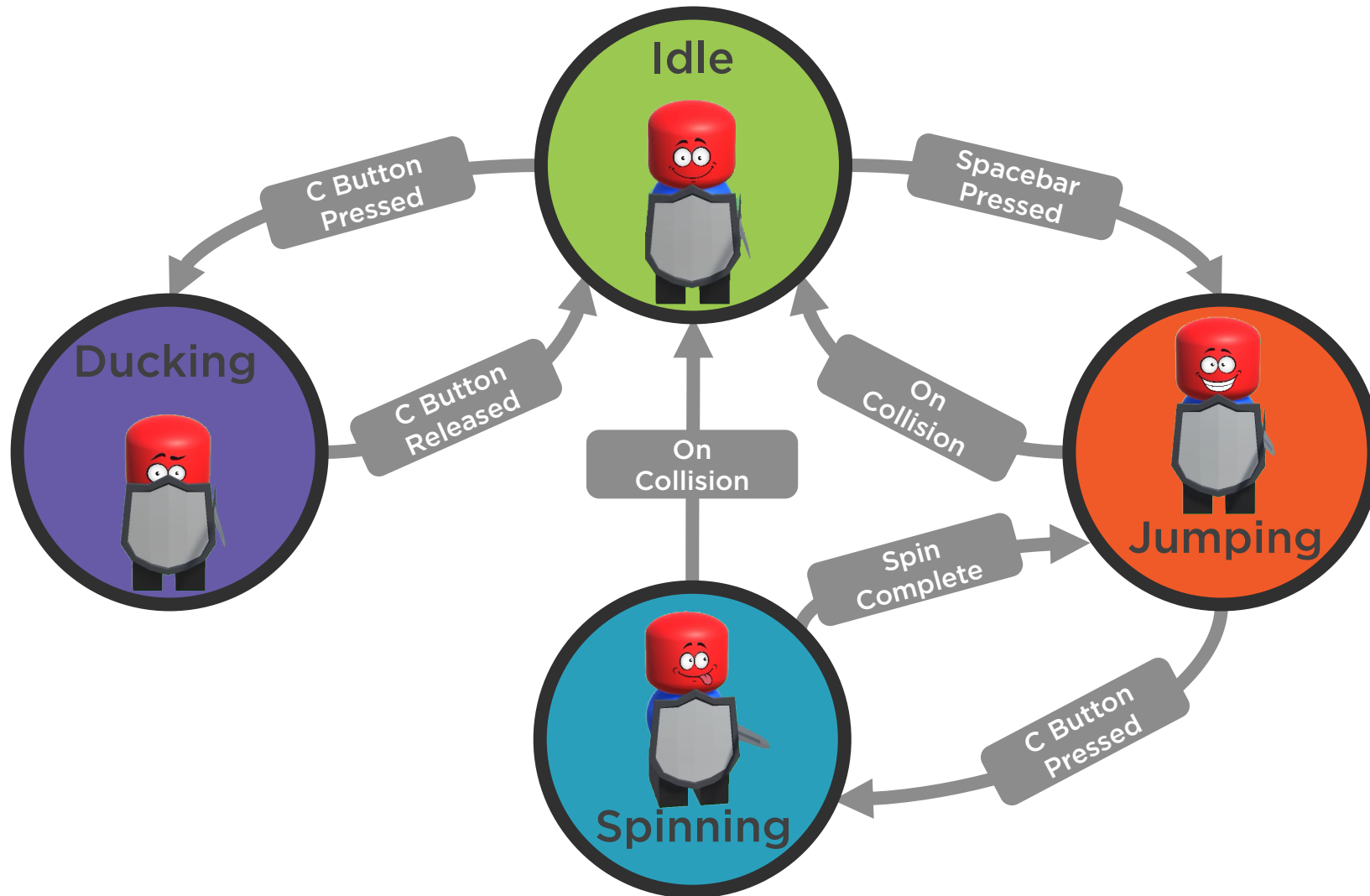


The conditions for **transitioning** between those states

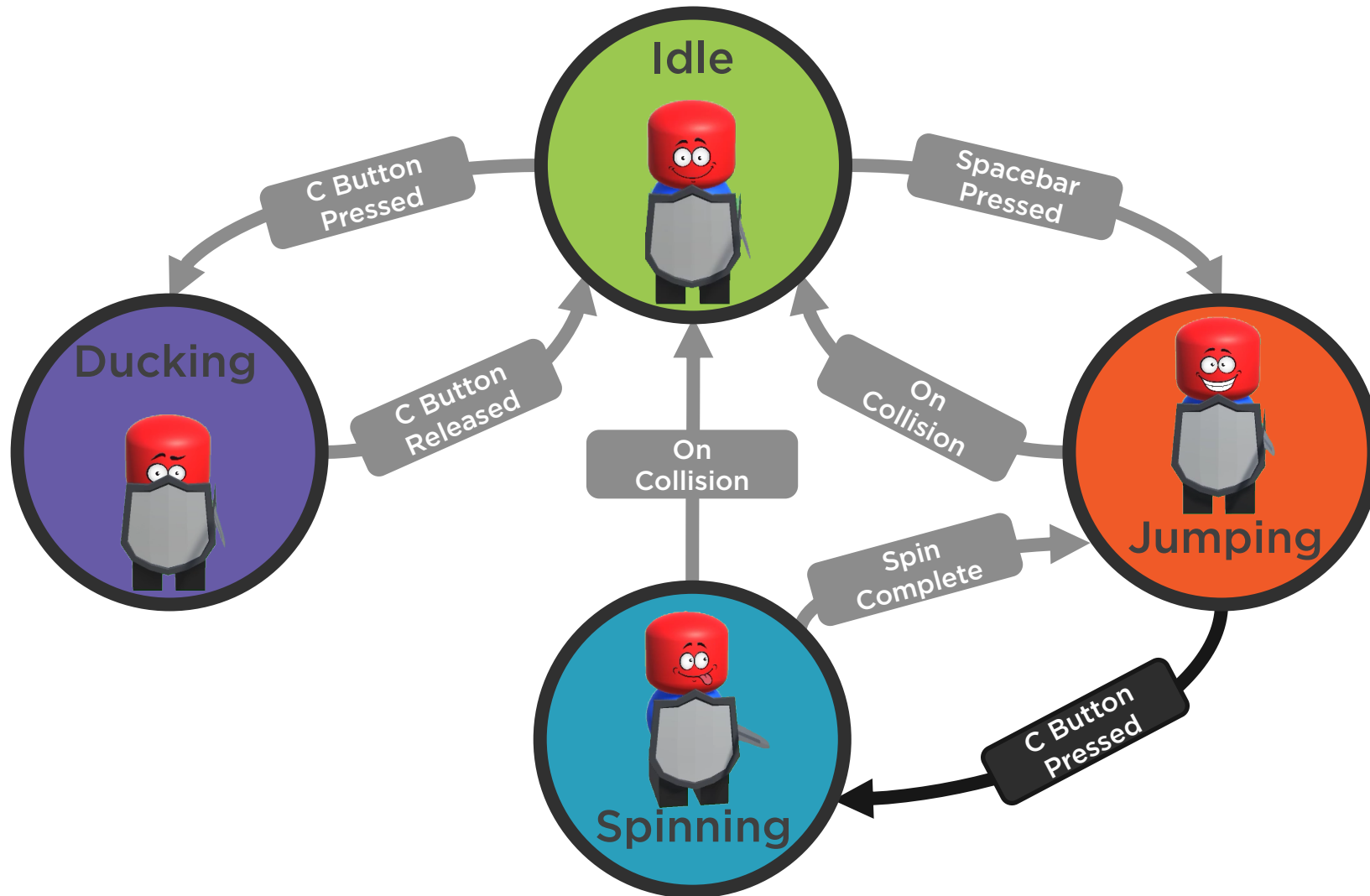


The state its in when initialized, or its **initial state**

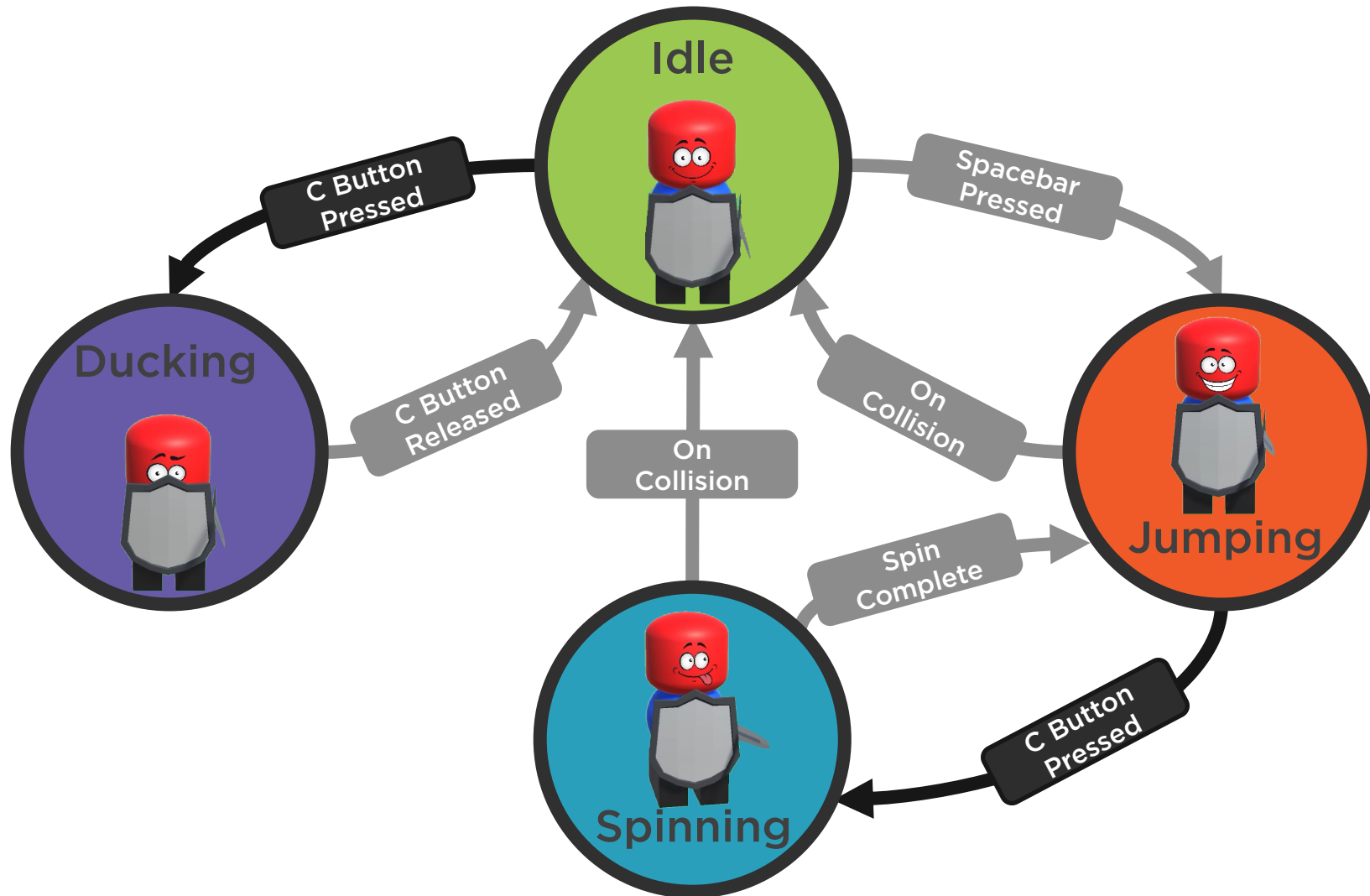
A Finite State Machine for Arthur



A Finite State Machine for Arthur



A Finite State Machine for Arthur



Each state is completely
self-contained



Building the Machine



Elements of a Finite State Machine

Context

Maintains an instance of a concrete state as the current state

Abstract State

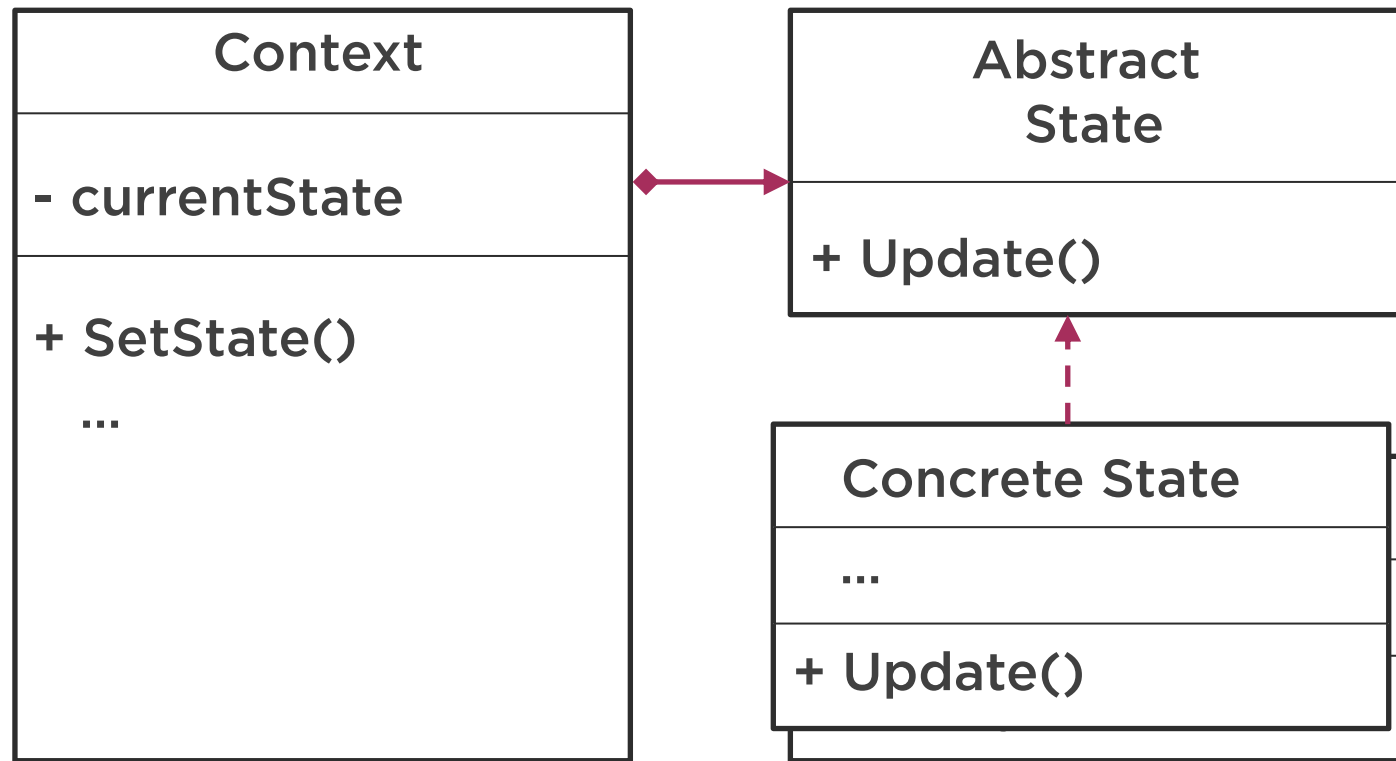
Defines an interface which encapsulates behaviors common to all concrete states

Concrete State

Implements behaviors specific to a particular state of context



Anatomy of a Finite State Machine



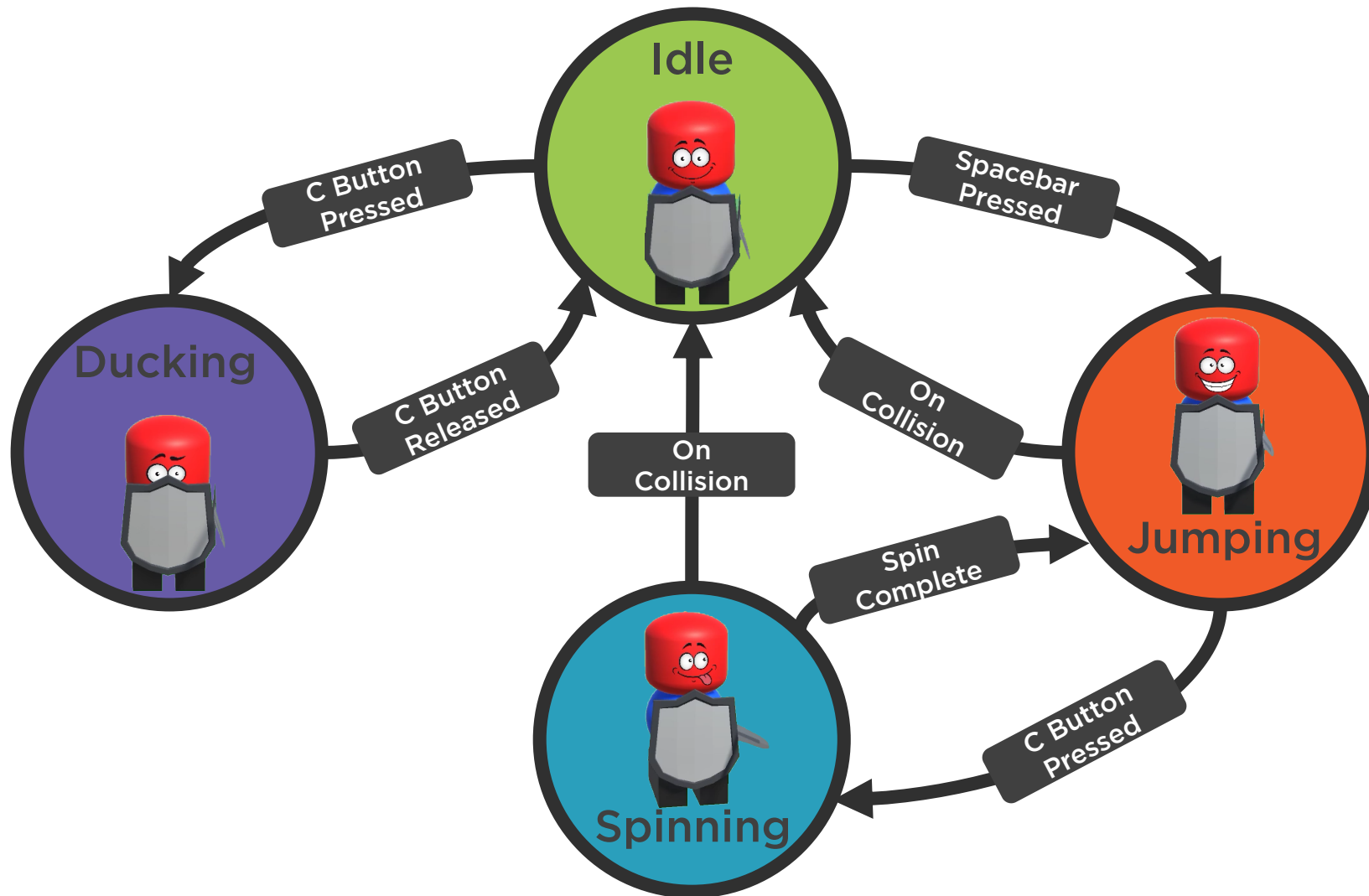
Coming Up



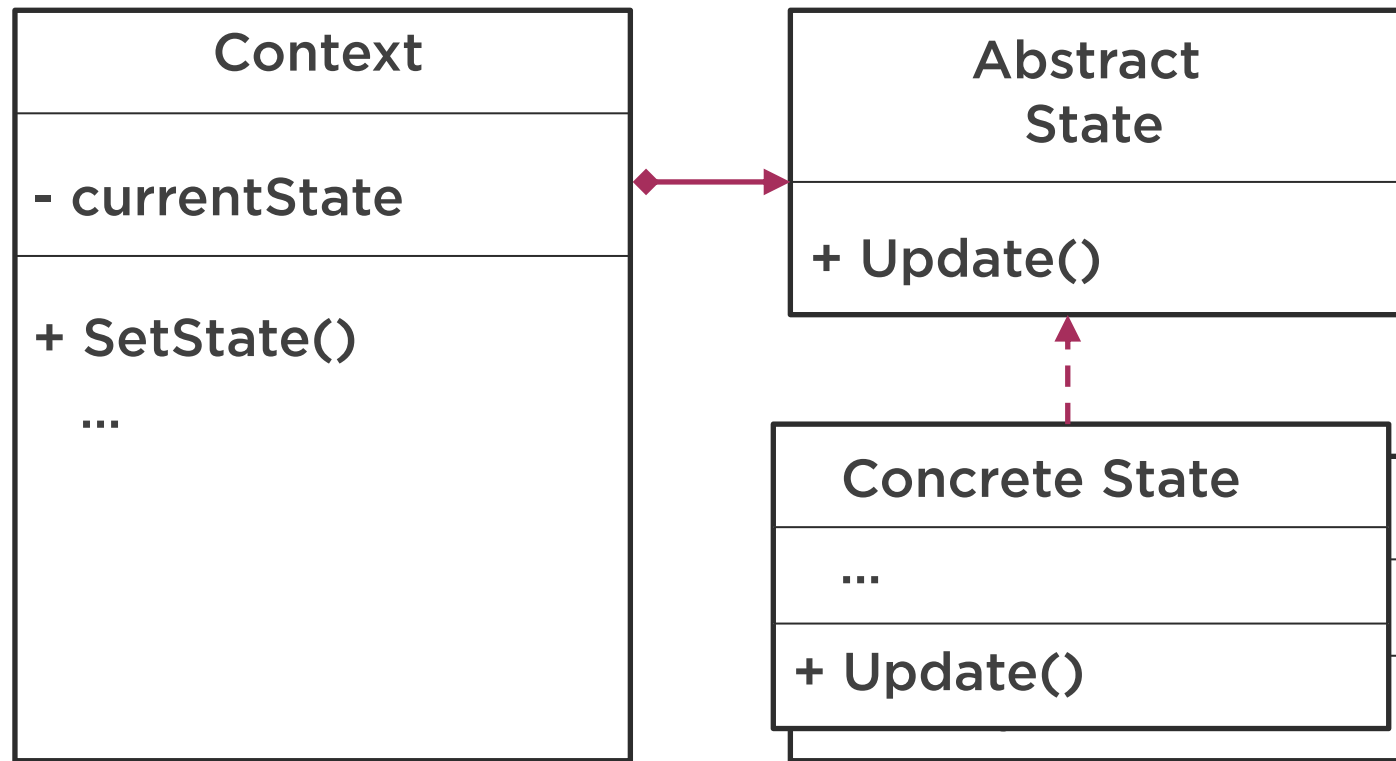
Concrete States



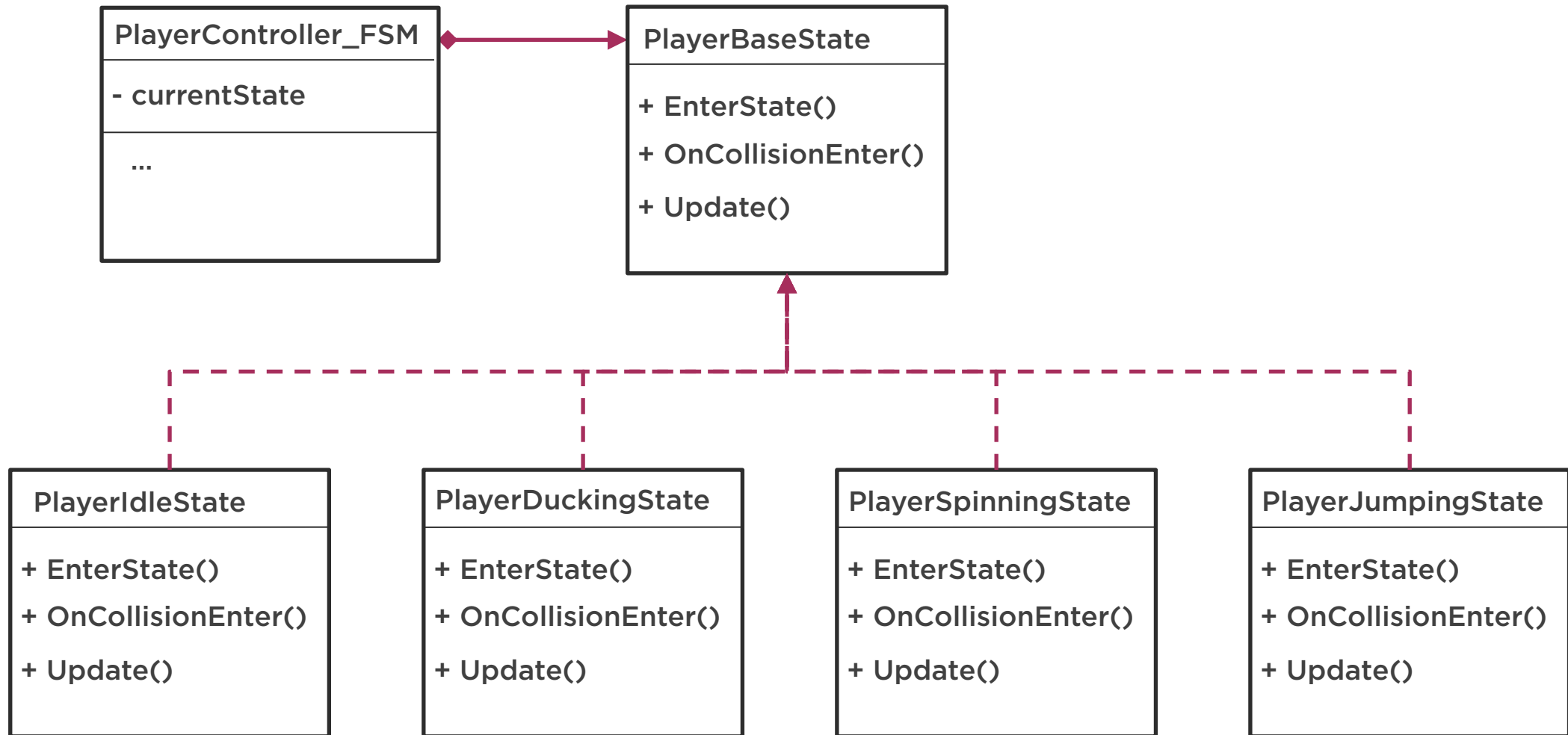
A Finite State Machine for Arthur



Anatomy of a Finite State Machine



Arthur's Finite State Machine



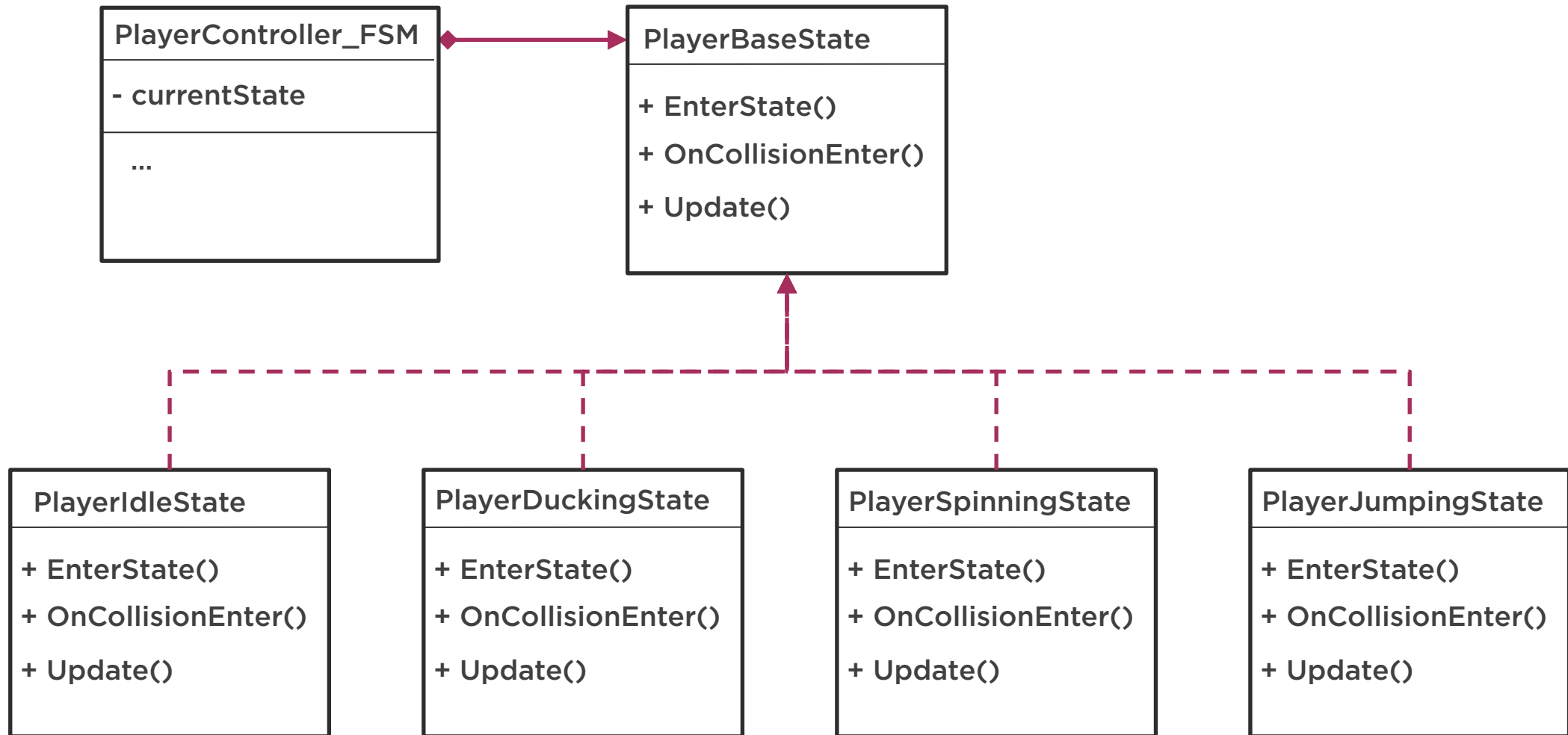
Coming Up



Context and State



Arthur's Finite State Machine



Elements of a Finite State Machine

Context

Maintains an instance of a concrete state as the current state

Abstract State

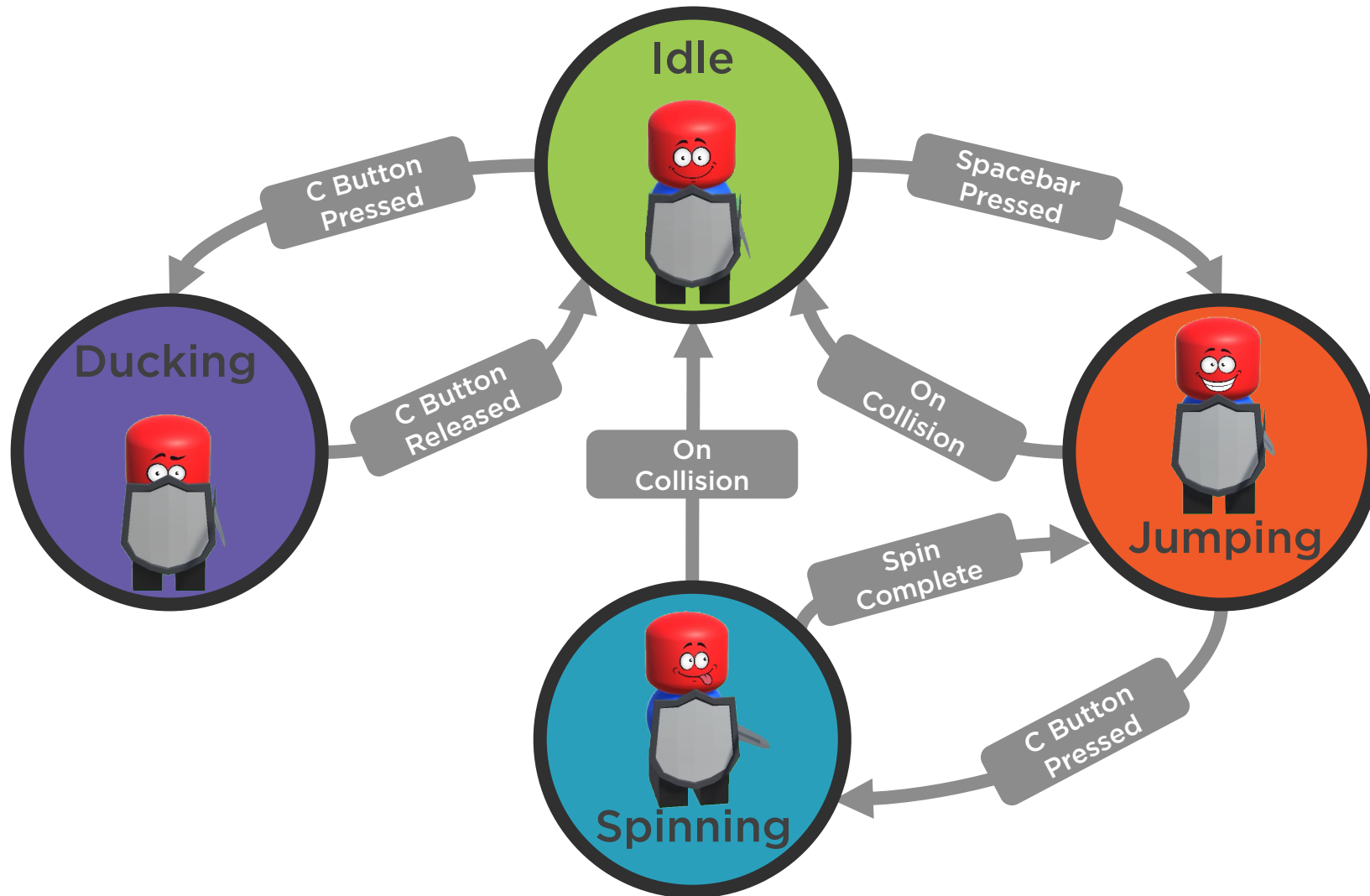
Defines an interface which encapsulates behaviors common to all concrete states

Concrete State

Implements behaviors specific to a particular state of context



A Finite State Machine for Arthur



Coming Up



Setting the Scene



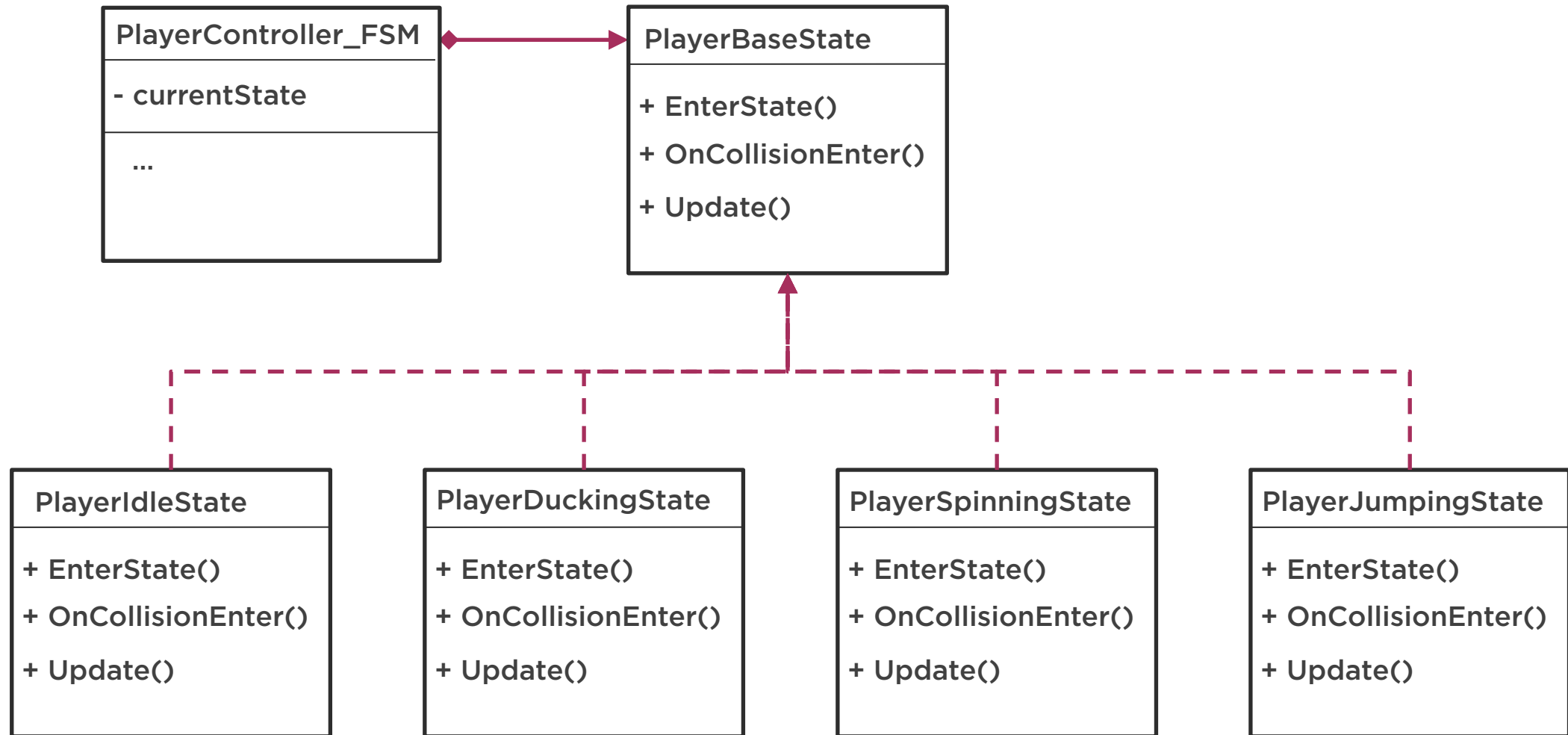
Coming Up



Beginning the Implementation



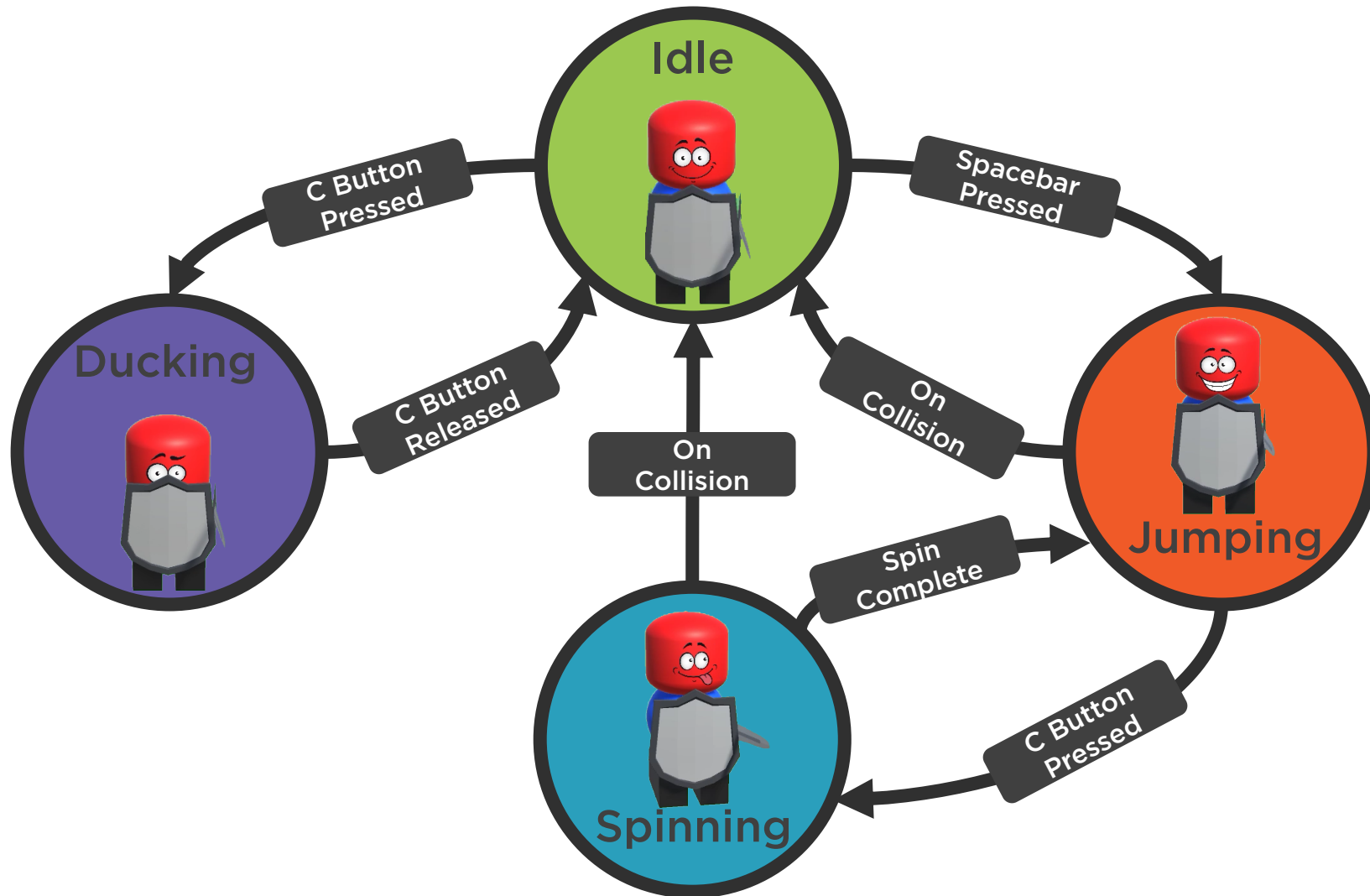
Arthur's Finite State Machine



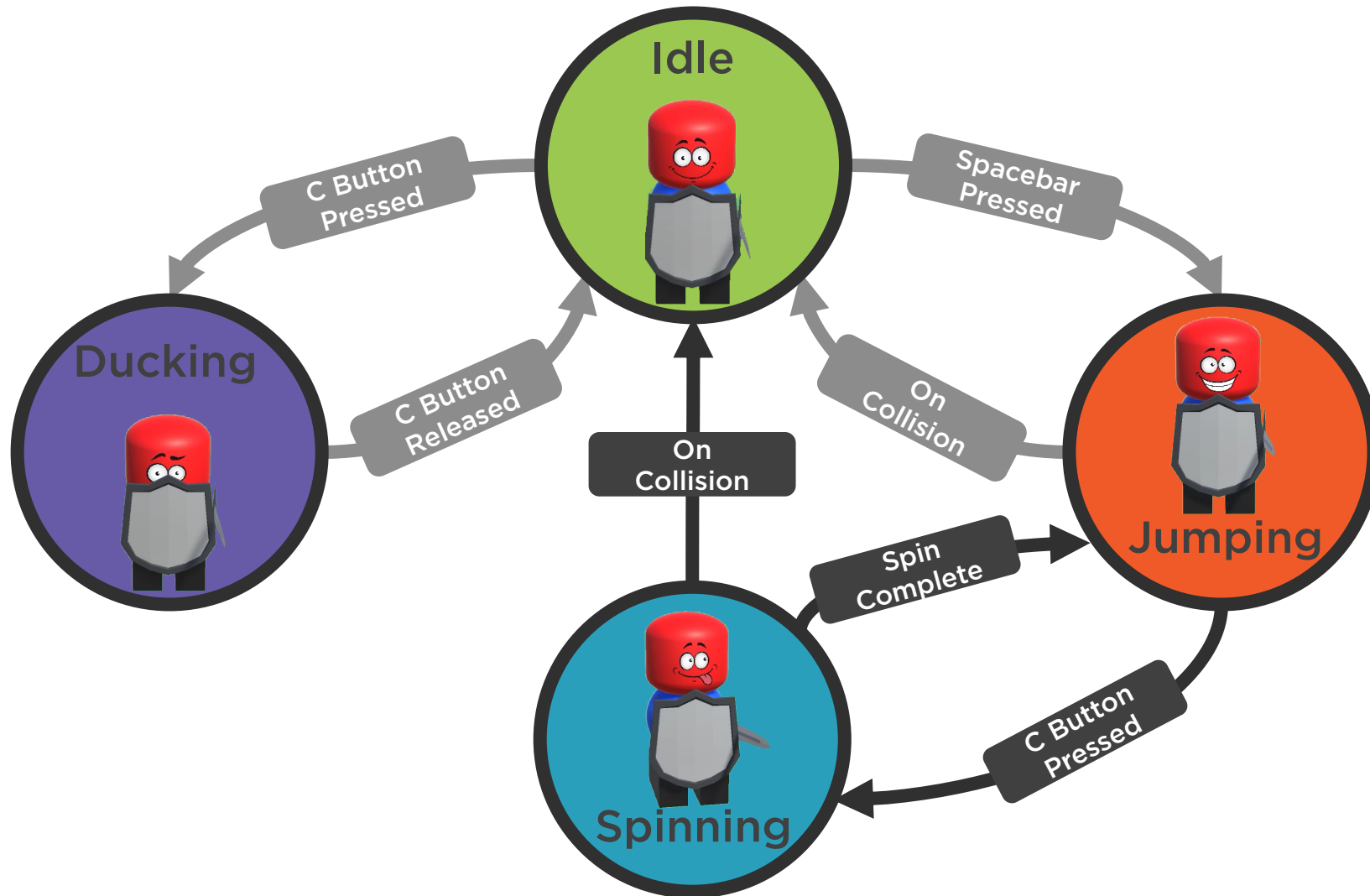
Continuing the Implementation



A Finite State Machine for Arthur



A Finite State Machine for Arthur



Beginning the Implementation



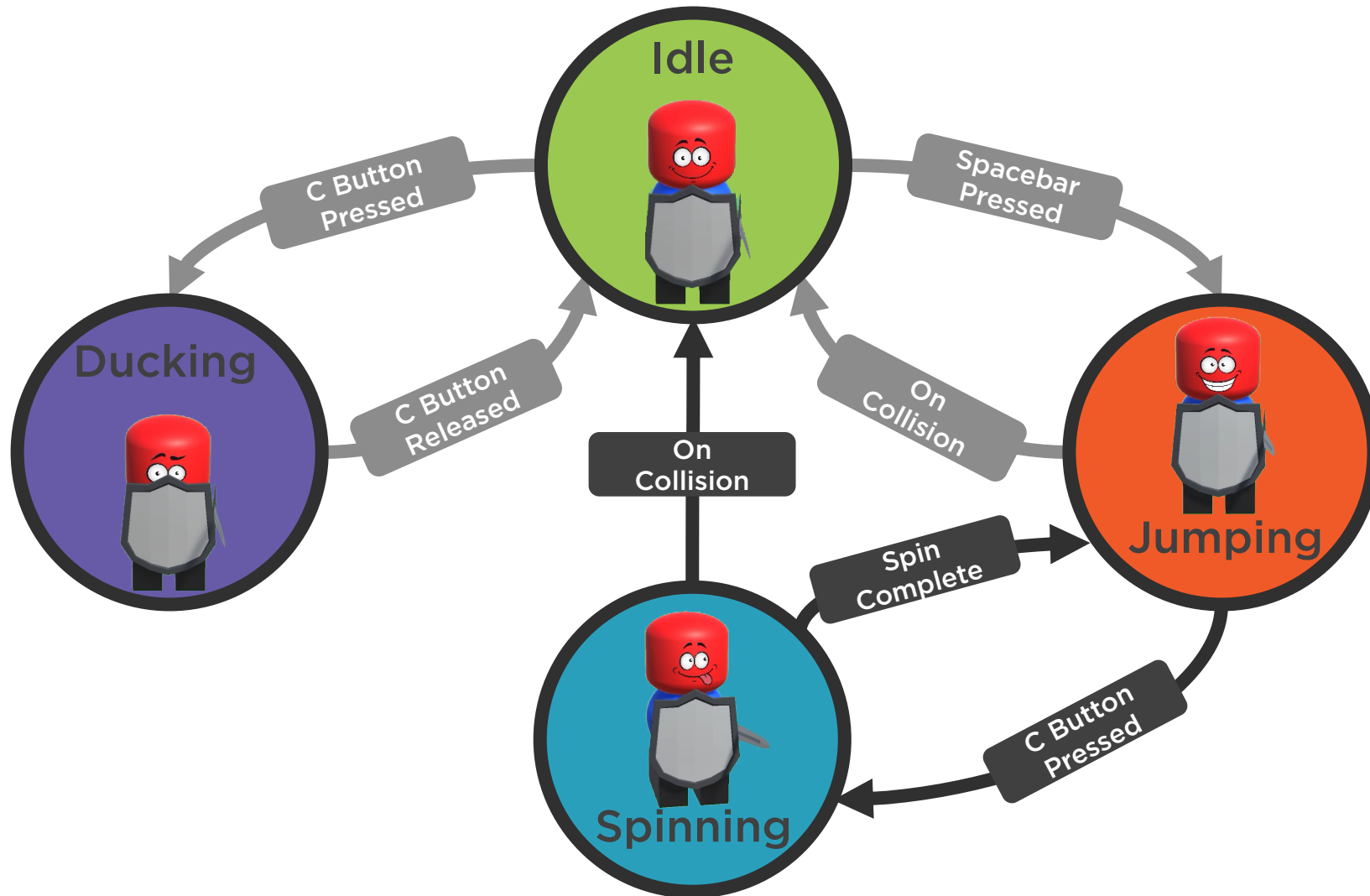
Coming Up



Module Conclusion



A Finite State Machine for Arthur



Swap Weapon Behavior

Actions

Arthur changes weapon

Triggers

X button is pressed

Conditions

Arthur is not jumping
Arthur is not ducking
Arthur is not spinning

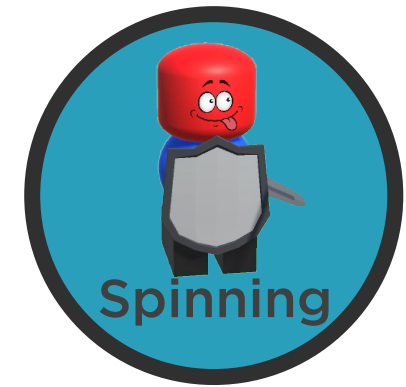
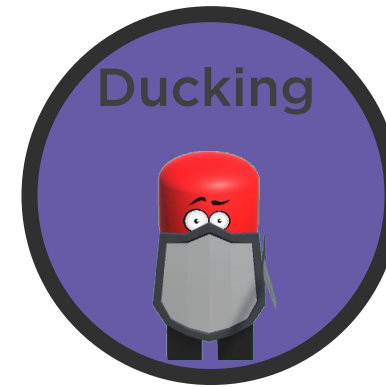
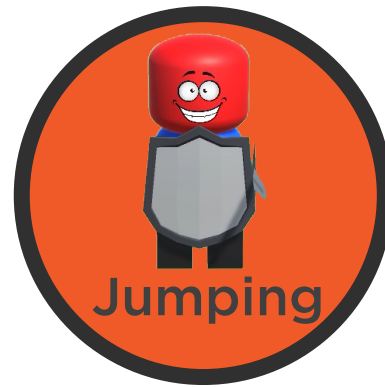


Finite State Machine

A Finite State Machine is an **abstract** machine that can be in exactly one of a finite number of states at any given time.



Arthur's States



Benefit of a Finite State Machine



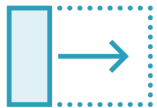
More modular



Easier to read and maintain



Less difficult to debug



More extensible



Disadvantages of a Finite State Machine



Takes time to set up



More moving parts



Potentially less performant



Finite State Machines are a
great addition to your
developer's toolbox.



Thank You!

