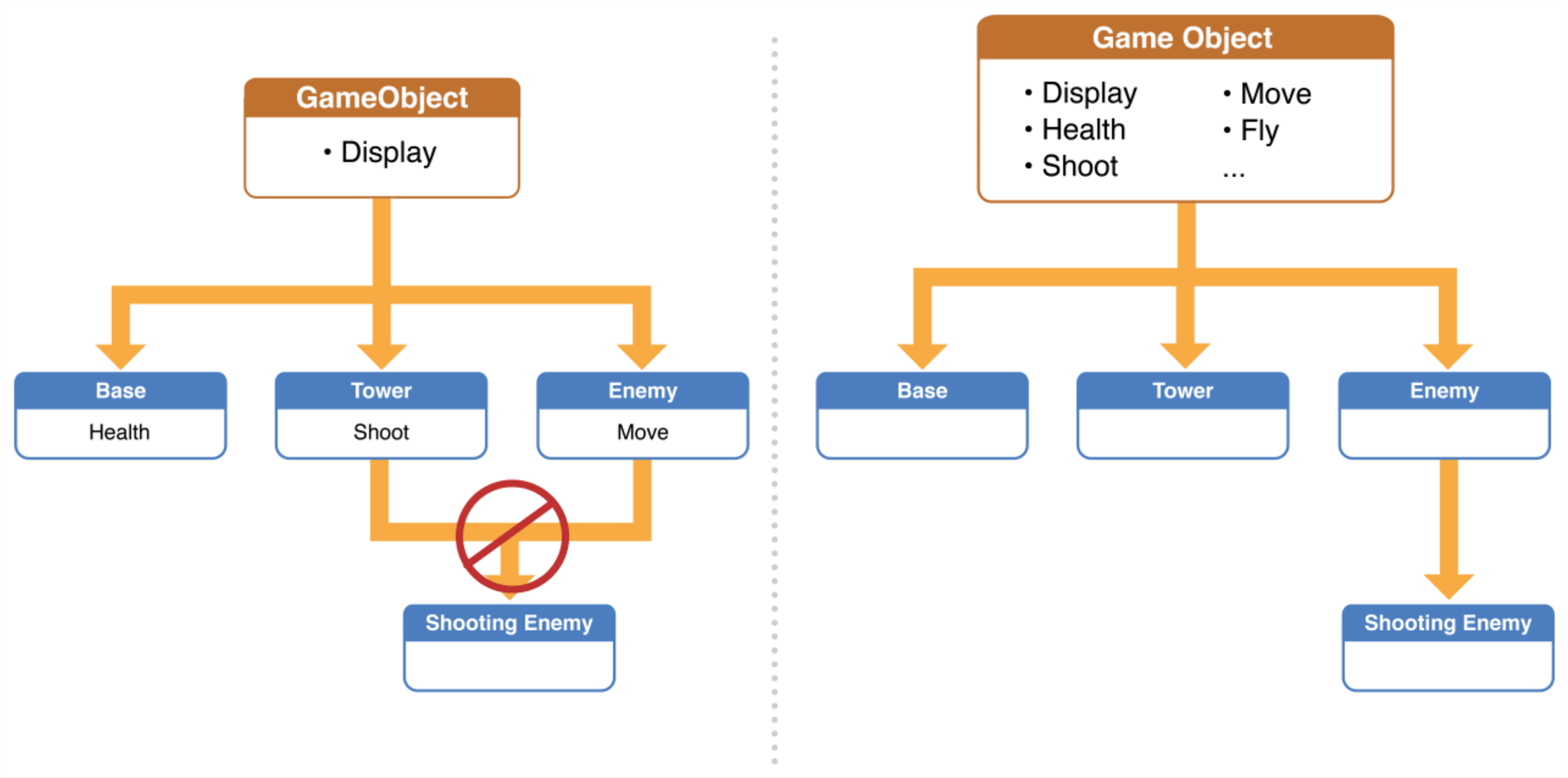
GameplayKit: beyond games

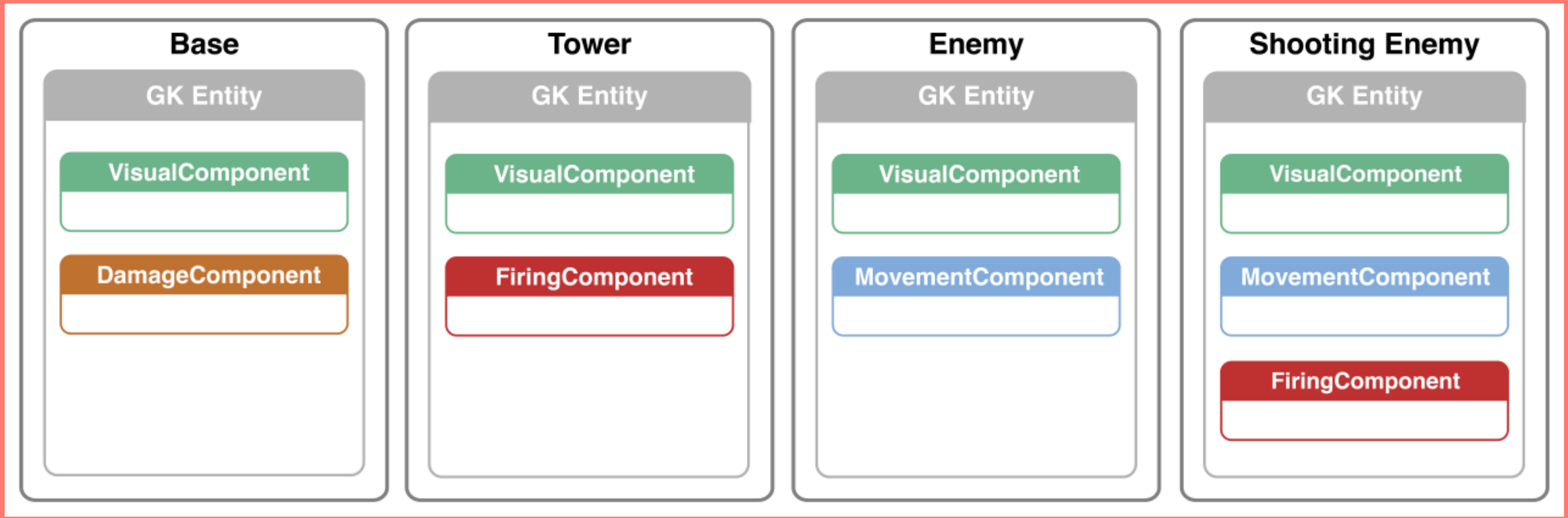
General goodies

- Components & Entities
- Random Numbers
- State Machine

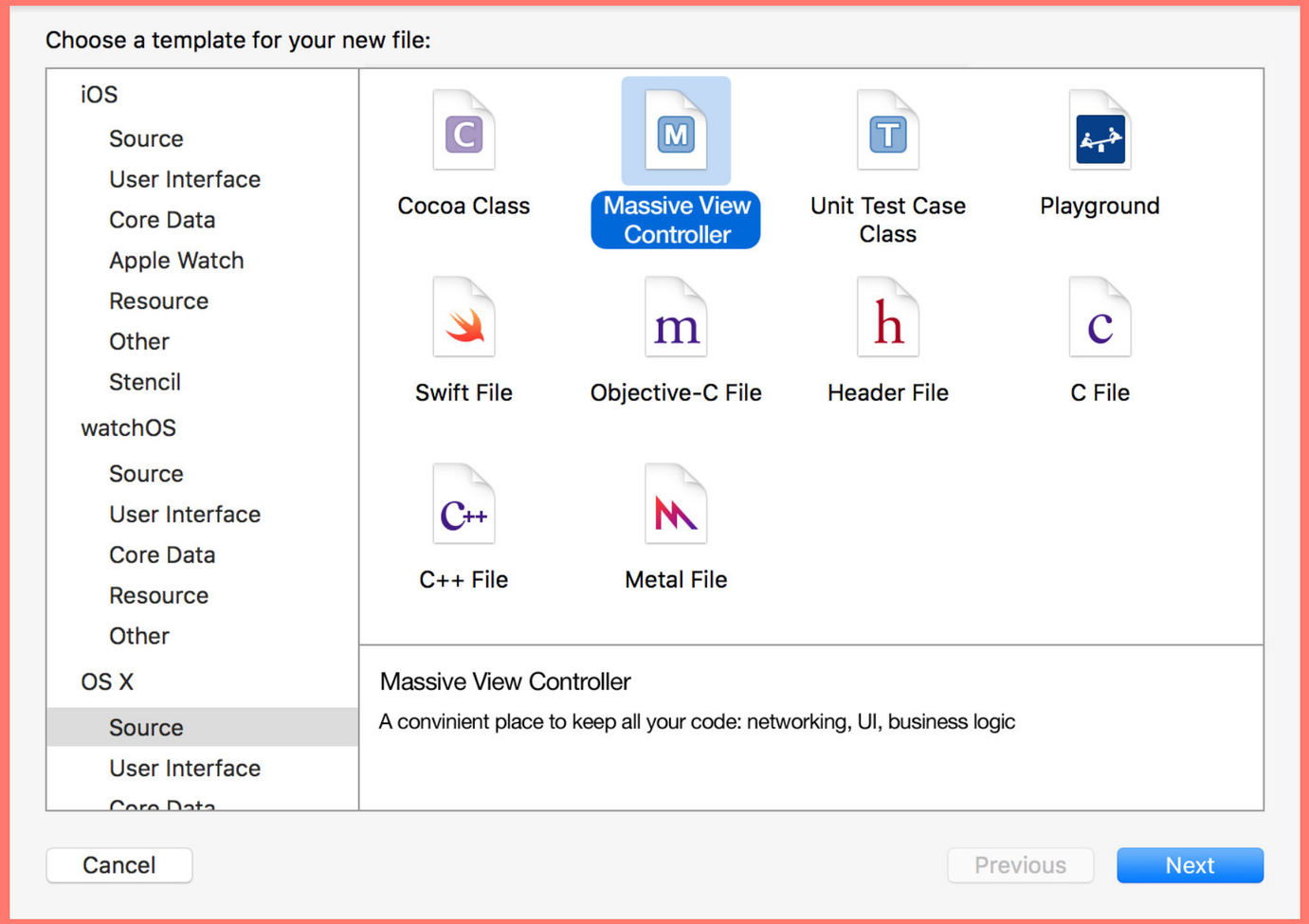
Components & Entities: Inheritance



Components & Entities: Composition



Components & Entities



Random numbers

Distributions:

Standard, Shuffled, Gaussian

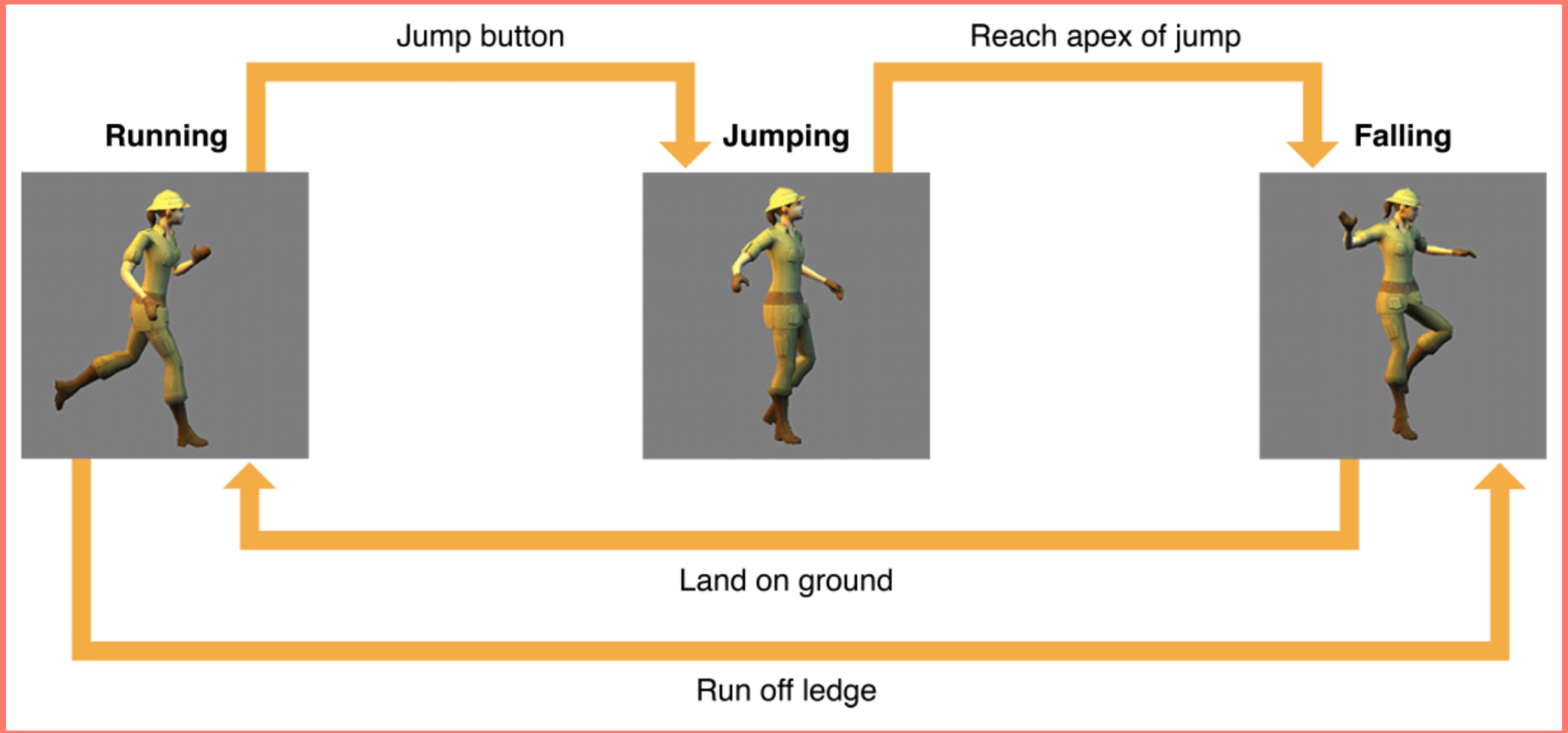
Sources:

Mersenne Twister, Arc4, Linear Congruential

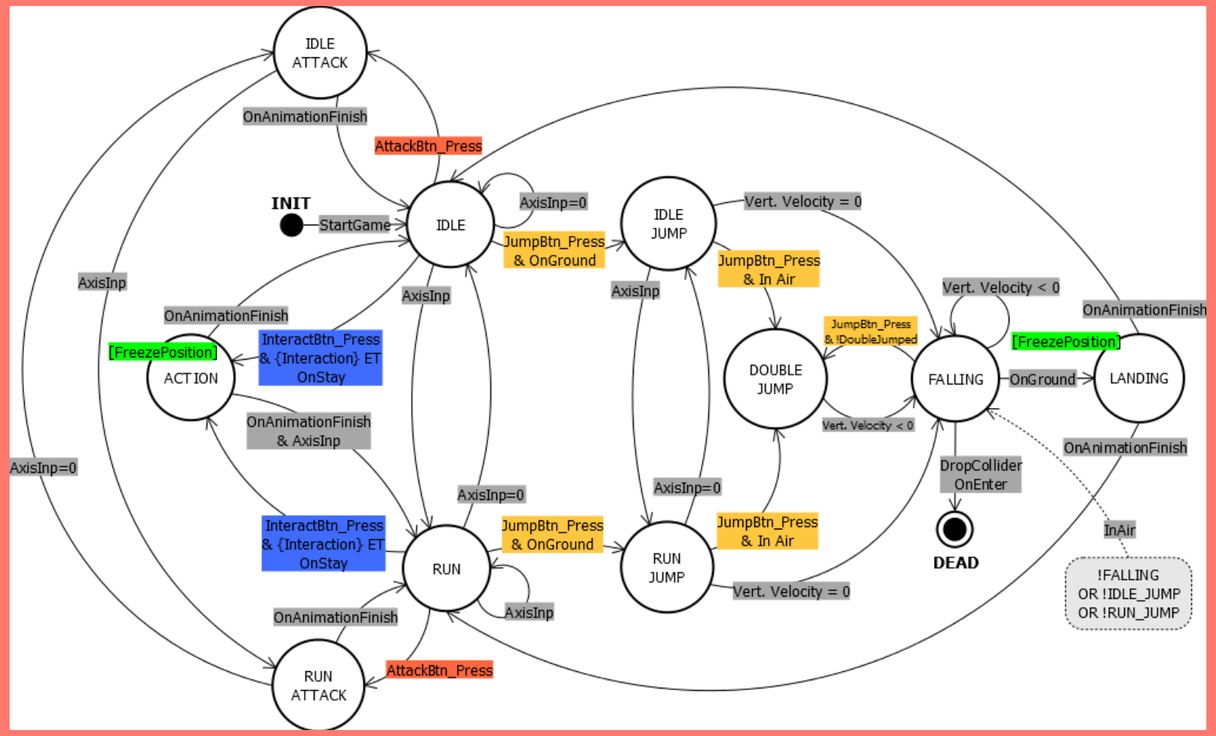
State Machine

- Disclaimer: The "S" word.
- "GameplayKit: State Machine for non-game Apps" invasivecode.com/weblog/gameplaykit-state-machine by @vicentevicens

State Machine: 101

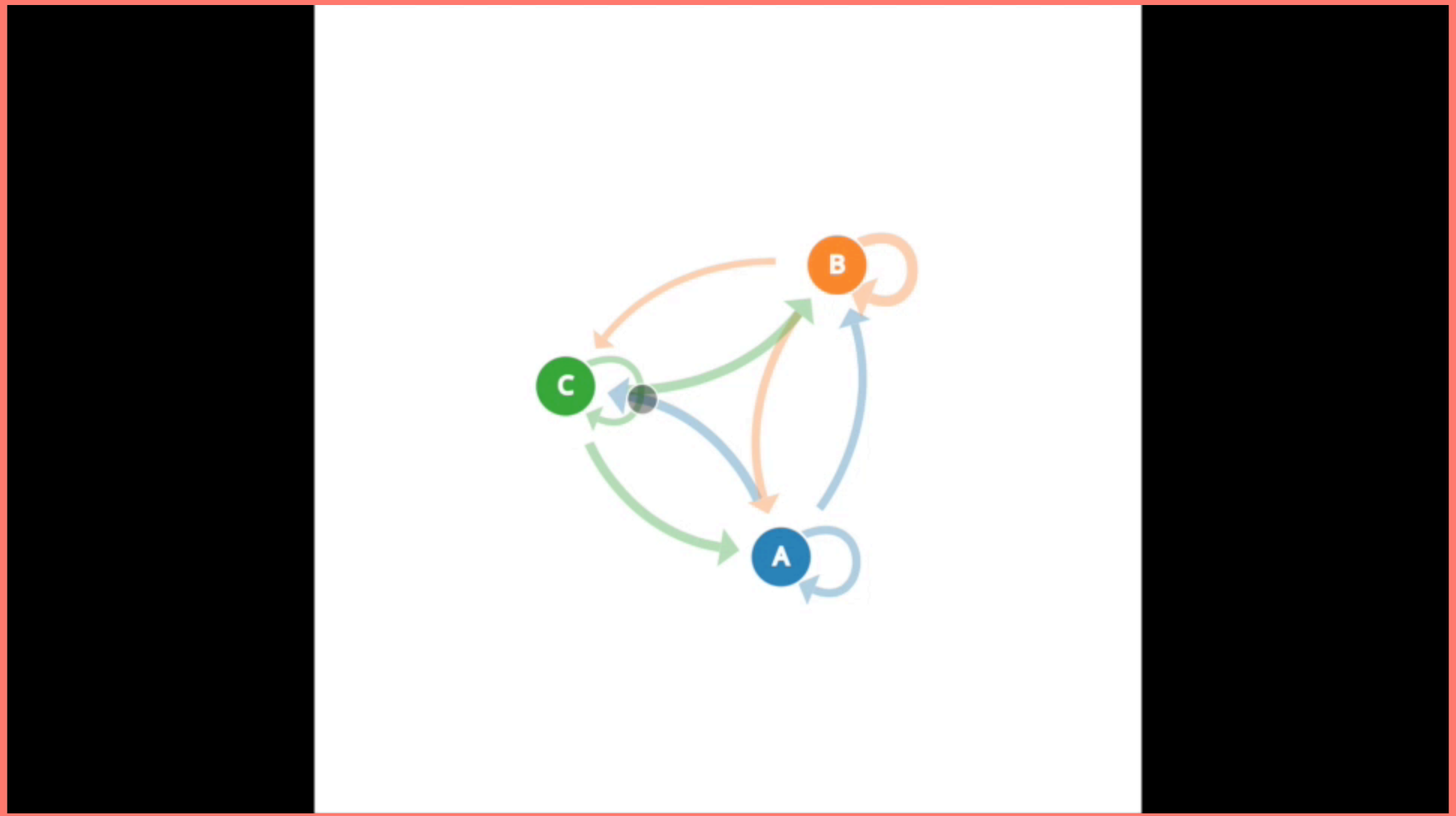


State Machine: 102[1]



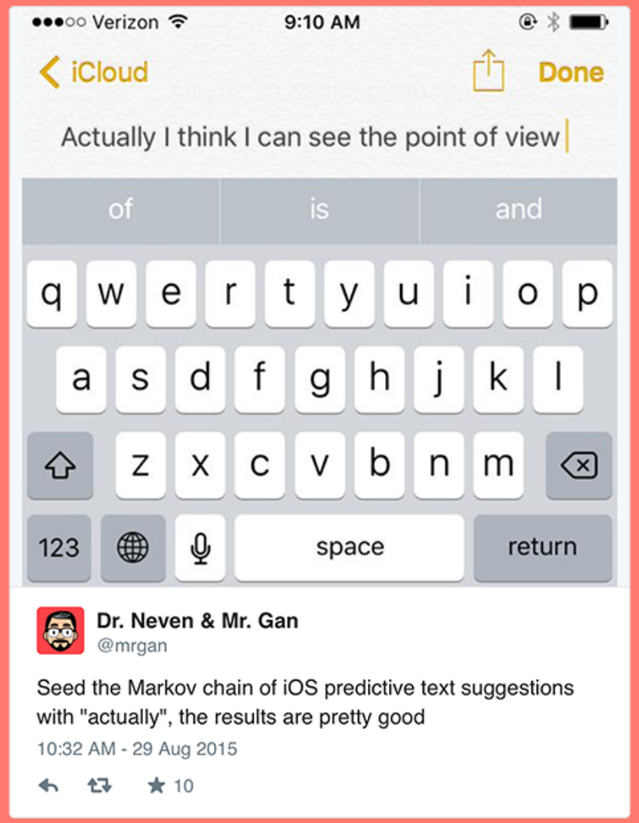
[1] How are video game AIs programmed? Is it a just a long series of "If Then" statements? reddit.com/r/explainlikeimfive/comments/2r6g74/eli5*how*are*video*game*ais*programmed*is*it_a/?limit=500

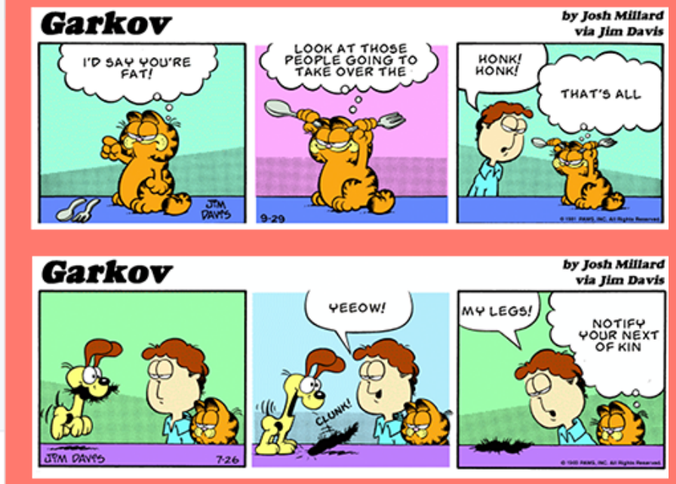
Markov Chains



Markov Chains

Modeling weather conditions, simulating stock exchange





Demo

SwiftDoc.org: Swift 3.0

```
class MarkovChainMachine: GKStateMachine {
    let outcomesMap: [[GKState]: [Double: GKState]]
    var buffer: [GKState]

    func enterNextState() {
        let next = nextState()
        enterState(next)
        buffer.removeFirst()
        buffer.append(next)
    }

    func nextState() -> GKState {
        let random = ...
        return nextState(buffer, random)
    }
}
```

MinMax

- Widely used in turn-by-turn games
- Applicable for 1+ player
- Increasing depth of prediction increases the computation time exponentially
- Alpha-beta pruning and other algorithms to speed up calculation

Demo

Sol: a Smart(er) Weather App


Carrier
10:55 PM
Light Snow
Chicago, IL
17°
H 43 L 5
Mon
Tue
Wed


Carrier
10:54 PM
Overcast
Austin, TX
36°
H 64 L 32
Mon
Tue
Wed


Carrier
10:57 PM
Mostly Cloudy
Houston, TX
58°
H 68 L 41
Mon
Tue
Wed

MinMax

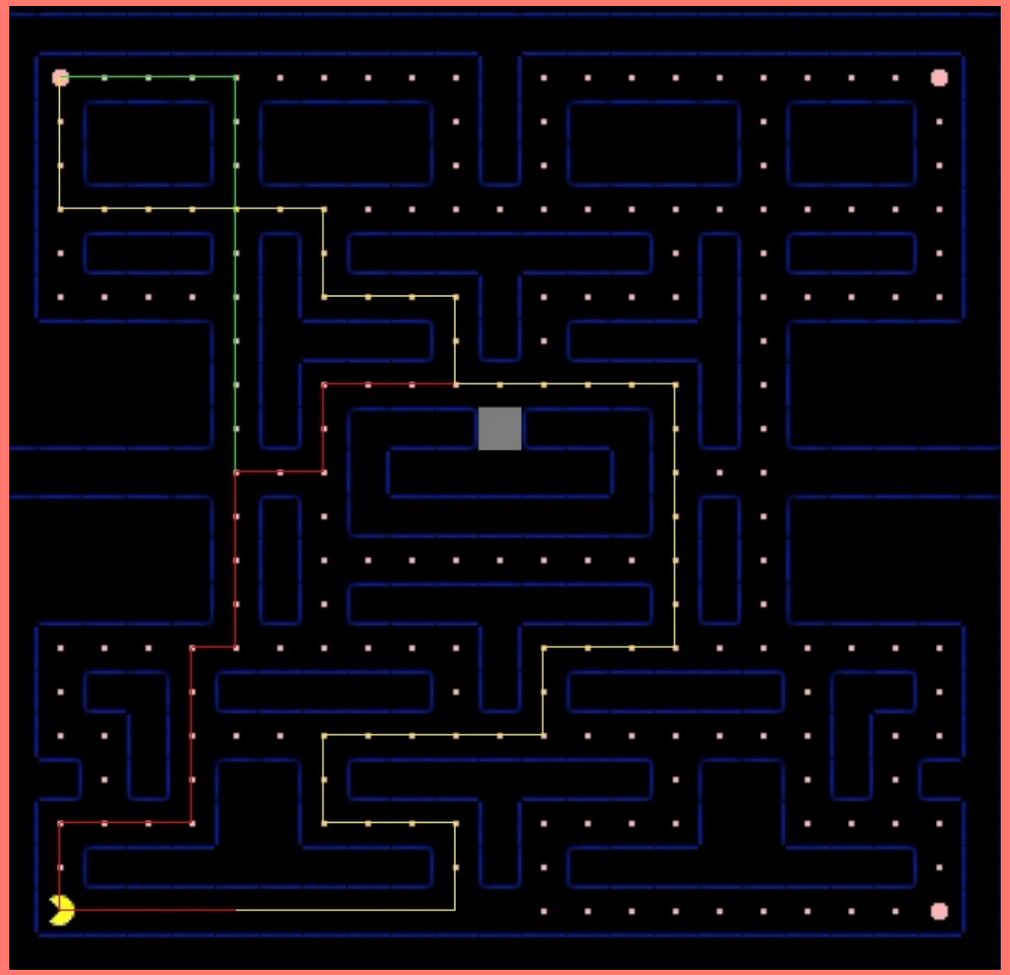
```
suggestions.register(AddCity.self) { history in
    switch history.filter({ $0 is AddCity }).count {
    case 0:
        return best
    case 1..<3:
        return good + 1
    default:
        return nil
    }
}
```

MinMax

```
var history: [GKState]

// GKGameModel
func scoreForPlayer(player: GKGameModelPlayer) -> Int {
    var maxScore = Int.min
    for predicate in predicates {
        if let result = predicate.score(history: history) {
            if maxScore < result {
                maxScore = result
            }
        }
    }
    return maxScore
}
```

Pathfinding



Pathfinding

Demo

In App Navigation

Pathfinding

```
func setupGraph() {
    root.addConnectionsToNodes([privacy, facebook], bidirectional: true)
    facebook.addConnectionsToNodes([facebookSettings, facebookAccount], bidirectional: true)
    facebookSettings.addConnectionsToNodes([facebookLocation], bidirectional: true)
    privacy.addConnectionsToNodes([bluetooth, location], bidirectional: true)
    location.addConnectionsToNodes([facebookLocation], bidirectional: true)

    graph.addNodes([
        root, privacy, facebook,
        bluetooth, location,
        facebookSettings, facebookLocation, facebookAccount
    ])

    favorite = facebookLocation
}
```

Pathfinding

```
func goToFavoriteNode() {
    let current = currentViewController.node
    let path = root.findPathFromNode(current, toNode: favorite)
    navigate(path)
}
```

DeepLearningKit

GameplayKit: beyond games

- GameplayKit reimplemented github.com/mohiji/JLFGameplayKit
- This presentation github.com/zats/Presentations
- @zats

Thank you!