

The Wild West

Complexity slows you down, so treat simplicity as a precious resource.

Using the wrong pattern can be worse than using no pattern at all, so this section is normally here to caution against over-enthusiasm. The goal of design patterns isn't to cram as many into your codebase as you can.

These parts of GameplayKit help you organize your code in a different way that allows for more clean organization and code reuse, especially as your games become larger and more complex.

Another benefit of this approach is that the predicates are serializable, so they can be stored in external files or sent over the network. With just a few more lines of code, we could move the predicate and enum to a .plist file and load them instead of hard-coding them. Then, a game designer on your team could make or tweak levels without editing code. The main coding work would be to add more state and outcomes.

<https://www.gangboard.com/blog/artificial-intelligence-tutorial#Overview>

=====

Engine v7.1: Everything is Event

Active: equip

Triggered: discard previous weapon on eQuip

SFX: associated to update instead of move

Decodable Effect, PlayReq, Event => Decoding heterogenous collection

(?) Cards & Hits contains abilities with options

(?) Ability with options: silentAbility:X, silentCard:X, playCardAs:X,Y

(?) Fix InputHandler active when engine busy

(?) AddHit once (with parameter: targets, times) => We want dynamite sfx trigger once

(?) localised string with R

AI: Goal based agent: evaluate move by evaluating outcome state

=====

Engine v6 => Engine v7

One class for each ability => Json scripted abilities (ability = playReqs x effects) + scalable

Get inPlay players by filtering health >0 => players as Dictionary + playOrder as [String]

Database multiple observable (state, move, event, validMoves) => Database (state, event)

EngineLoop as temporary class => EngineLoop as Completable + Testable

Challenge object boilerplate => Resolving hits one by one

MoveMatcher boilerplate (autoplay, effect) => Triggered ability once per event

Database massive switch case to update method => EventMatcher + Testable

Hardcoded animation duration per event => EventMatcher define characteristic of each Event

Implement rules based on card name => Implement rules based on abilities attached to card

Implement rules using massive State extension => Implement rules using State implementation

Card suit and value conditions => Regex

Some default abilities may be disabled by "Silent"

=====

Backlog

Firebase: Sign in with Apple

Firebase: Cleanup game when all users quit

UI: PlayerSelector: Name - ability short description

UI: RoleSelector: Name - short description, imageSlideshow

UI: non copyright assets <https://www.svgrepo.com/collection/wild-west-collection-2/>

AI: AgressiveAI: attack weakest player

AI: deputy should avoid killing sheriff with global attack (indians, gatling, dynamite)

AI: SafeAI: when discarding excess cards, prefer keep missed or bang

UI: Settings from Menu, Game

UI: general store view
UI: animate gain health
UI: animate loose health
UI: select target player on overlay view
UI: select target card on overlay view
UI: configure game speed
UI: configure sound
UI: undo move
SFX: Eliminated specific voice

=====

Roadmap

v1: Structure the game + Single player VS AI + Basic Rules

Game state: exhaustive states
Game Engine: action = rules applied to state : exhaustive behaviours
View List of possible actions
Execute action through engine
View history
View instruction
Single player VS CPU
Implement rules
Player special abilities
Distribute via Firebase distribution
Get feedback from friends
Distribute <https://github.com/stephtelolahy/WildWest>

v2: Multiplayer + Store data online + Manage users

Gameplay v2: animation/update chaining
Multiplayer: Firebase Realtime database

v3: Extensions + Rules scripting

Use unprotected assets
Add Dodge City extensions: figures, brown cards, blue cards https://tesera.ru/images/items/125593/bang_dodge_city-rules.pdf

Add High Noon scenario

Gameplay v3: zoom card, drag and drop

=====

References

- Online web game <http://chriscarr.name:8080/westerncardgame/index.html>
- Cards list https://bang.dvgiochi.com/cardslist.php?id=7#q_result
- Card lists <http://www.dvgiochi.net/bang/BANG!%20Card%20List.pdf>
- Official blog <http://bangcardgame.blogspot.com/>
- Bachelor thesis <https://is.cuni.cz/webapps/zzp/download/130199523>
- iOS Emoji <http://www.grumdrig.com/emoji-list/>
- Sound effects https://freesound.org/people/cabled_mess/sounds/350923/
- Game engine strategies <http://web.eecs.umich.edu/~soar/Classes/494/talks/Schumaker.pdf>

#Jeu de base

#####

16 Cartes personnages

Bart Cassidy

Black Jack

Calamity Janet

El Gringo

Jesse Jones

Jourdonnais

Kit Carlson

Lucky Duke

Paul Regret

Pedro Ramirez

Rose Doolan

Sam le Vautour

Sid Ketchum

Slab le Flingueur

Suzy Lafayette

Willy the Kid

7 ROLES

Shérif x1

Hors-la-Loi x3

Adjoint x2

Renégat x1

63 cartes aux bords marrons

25x Bang

12x Raté

06x Bière

04x Braquages

04x Coup de Foudre

03x Duel

02x Indien

02x Magasin

02x Convoi

01x Gatling

01x Saloon

01x Diligence

17 - cartes aux bords bleus

2x Volcanic

3x Schofield

1x Remington

1x Carabine

1x Winchester

2x Mustang

1x Lunette

3x Prison

2x Planque

1x Dynamite

Extension DODGE CITY

#####

15 Cartes personnages

Apache Kid
Belle Star
Bill Sans Visage
Chuck Wengam
Doc Holiday
Elena Fuente
Greg le Fossoyeur
Herb Hunter
José Delgado
Molly Stark
Pat Brennan
Pixie pete
Sean Mallory
Tequila Joe
Véra Custer

8 ROLES

Shérif x1
Hors-la-Loi x3
Adjoint x2
Renégat x2

19 cartes aux bords marrons:

4x Bang
2x Esquive
2x Bière
1x Indiens
1x Bagarre
1x Magasin
1x Tequila
1x Coup de Foudre
1x Springfield
1x Coup de Poing

- 1x Pianiste
- 1x Whisky
- 1x Braquage
- 1x Raté

14 cartes aux bords vertes

- 1 Bible (Effet Raté) + Pioche 1 carte
- 1 Plaque d'acier (Effet Raté)
- 1 Stetson (Effet Raté)
- ☒ 1 Sombrero (Effet Raté)
- 1 Pony express (Effet Diligence pioche 3 cartes)
- 1 Canon (Effet Bang sur tous les joueurs)
- 1 Derringer (Effet Bang => à porté de tir égal à 1) + 1 carte de la pioche
- 1 Couteau (Effet Bang => à porté de tir égal à 1)
- 1 Gourde (Effet Bière)
- ☒ 1 French Cancan (Défausser 1 carte d'un joueur n'importe lequel)
- ☒ 1 Conestoga (Prendre une carte => n'importe quel joueur)
- ☒ 1 Pepperbox (Effet Bang => à porté de tir)
- ☒ 1 Fusil a bison (Effet Bang => n'importe quel joueur)

☒ 07 cartes aux bords bleus

- ☐ 1 Remington
- ☒ 1 Discretion
- 1 Jumelles
- ☐ 1 Carabine
- ☐ 1 Dynamite
- ☐ 1 Mustang
- ☐ 1 Planque

DONE

#####

☐ Engine v4

☐ Force dismiss current dialog if presenting reaction move or game over

Choose figure simply
UI: autosave setup in userDefaults
Player copy properties from figure
AutoPlay matcher return single move
Fix discard beer on dynamiteExploded
SFX move = drawsFomPlayerDamagesHim
SFX move = discardExcessCards
Player.lastDamage: DamageEvent?
Landscape layout
Rule: AllPlayers contains eliminated (health=0)
UI: Title white over image and health bottom
UI: role icon
SFX on discard excess cards
Remove StartGame -> GameSetup
Remove figures without ability
Describe: WildWest, the unofficial clone of Bang card Game
Add rules button
Add email link
UI: show instruction at bottom of screen
AI: fix renegade is not attacking while duel sheriff
Challenge: is a prior state waiting for reaction
AI: estimate roles with stats

Engine v5

Separate state from valid moves and executed moves
Calamity Janel ability as autoplay
Bug AI sheriff is not starting game
Bug delayed move on tap
UI: Assisted challenge -> randomAI
UI: cards specific values and suits
UI: animate card moves
UI: do not close game with gameOver popup
UI: animate card with placeholder on target position
Engine: delay updates

Engine.update: pull one card instead of n cards

Bug: instructions play any card

Rule: BatCassidy: draw cards on auto-damage by dynamite -> effect of .explodeDynamite + no remaining damage

Refactor Player.bangCardNames, Player.missedCardNames

UI: display deck cards count

Rule: add all randomised discard pile to deck if empty

Challenge + Editing: copy all properties to avoid losing data

Engine: Refactor Update + executionTime {0, 1} -> grouping updates where totalTime = 1

StartTurn: Refactor + blackJack

Update animation + revealHandCard: zoom

Update animation + flipDeck: zoom

Bug: discard beer should definitely remove the bang challenge

LuckyDuke: Player + flippedCardsCount + 1

Fix luckyDuke resolve dynamite by flipping two cards

UI: other moves button

SidKetchum: Can discard 2 cards for 1 life -> valid move + new action button (+1)

Rule: endTurn + discard all excess cards once

Rule: remove MoveName.play

Rule: Challenge.Name = Move.Name

UI: open rules in Safari

Move.Name -> Text Description

Move.Name -> SFX

Setup firebase

Simplify State

GameDatabase expose Observable<State>

GameUpdate.playerSetBangsPlayed(playerId, count) x2

GameUpdate.playerGainHealth(playerId, health)

GameUpdate.playerLoseHealth(playerId, health, damageEvent)

Engine v6

GameEngine(GameLoop)

GameLoop: dispatch(after update.delay)

Bug game loop

Track event game_over (outcome, winner, players_count)

RxDatabaseProtocol

RxUpdateExecutorProtocol: execute(update, database): Completable

Firebase: decode state

Firebase State + Observation

Firebase all transactions

Better error handling

Do not support duplicate cards in the same collection (with stagecoach_9_spades) -> stagecoach_8_spades

GamSubjects. Remove allPlayers

Use Result<T> completion object

Add extra level for game state

Remove fatalError on DAO

Fix Bug saloon not working

Remove subjects from engine

Fix crash animate update.playerDiscardInPlay

Firebase ExecutedMoves + Observation

Firebase ValidMoves + Observation

Firebase ExecutedUpdates + Observation

Rotate game viewed as another player

Create remote game if needed

Remote game has no CPU players

Join game by choosing playerId

remove UserPreferences.shared

Game: mark started

Firebase: MenuViewController: show connected user photo

Firebase flattening user status

Firebase: GameViewController: show player photo : game.users:
[playerId, WUserInfo]

UI: Menu: choose role

Architecture: DI system using Resolver

Firebase: Verify memory release (detach listener)

MoveMatcherProtocol + clarity

Challenge: optional properties

UI: require sign in only while play online

UI: hide player scores

Firebase: Sign in anonymously + custom name

UI: show roles at game beginning

Figures special abilities

Improve attribute

Jourdonais: Has a barrel at all times -> player's ability

PaulRegret: Has a Mustang at all times -> player's ability

RoseDolan: Has an Scope at all times -> player's ability

WillyTheKid: No limit on Bangs per turn -> player's ability

CalamityJanet: Bangs can be played as Misses, and Misses as Bangs -> new validMove

StabTheKiller: Others need 2 misses to counter his Bang -> Challenge + neededCounter

LuckyDuke: Flip 2 cards on a draw and choose 1 -> change how resolve works (dynamite, jail, barrel)

Trigger an effect on event

BartCassidy: Lose health, draw a card -> effect of .pass or .explodeDynamite

Elgringo: Draws card from player damaged him -> effect of .pass

SuzzyLafayette: Draw a card when hand is empty -> effect of .any

VultureSam: Takes all cards from dead players -> change how eliminate works

Can choose special move

SidKetchum: Can discard 2 cards for 1 life -> valid move + new action button (+1)

Changes phase1

BlackJack: Show second draw, if red suit, draw another -> effect of .startTurn

JesseJones: Can draw first card from other player -> N valid moves for each players card or deck

PedroRamirez: Can draw from discard -> 2 valid moves for discard or deck

KitCarlson: Draw 3, keep 2, put 1 back -> 3 valid moves for each combination of top 3 deck cards

Engine v7

Cards can be scripted

Use Pods + Tests

Beer: onHitEliminating

each effect of an ability must be applicable unless it is marked
optional

GameState: State

GameMove: Move

Update: Event

Move(name, args)

EffectFunc: (args, ctx) -> [Event]

EventFunc: (args, ctx) -> ctx

Attribute as String

StateImpl

Test dependencies using Resolver

DataBaseImpl

MoveArgs => Dictionary<Key: [String]>

Figure => Card + type=figure + attribute { bullets={number}}

Player + Override card ability

Engine, Loop, EventQueue, Timer

Active moves as event

Override abilities using attributes: playBangAsMissed,
playMissedAsBang

Restricted database

Arrange player according to observer identifier

Event wait delay

EventManager define characteristic (updateFunc, duration) of each
Event

Setup

Resource + loadCardList

discardBeer is a beer card ability

state abilities

State.outcome => state.winner

Engine. CurrentTask => .running
NextTurn as triggered
Resources + default.json groups all inherent abilities
Engine + triggered ability prioritization
Engine + setup(preferredRole, preferredFigure)
AI Agent(AI, RoleEstimator)
RoleEstimator(sheriff, abilityScores)
AI(MoveEvaluator)
MoveEvaluator(abilityScores, RoleEstimator, RoleRelationship)
RoleRelationship()
Game simulation
onYourTurnPhase => onPhase + isYourTurn
isPhase = 2 => + verify hits is empty
Clean integration
Fix ordered deck cards
Engine + Hit.name
/!\ ResourcesLoader: remove throws
/!\ test cardsHaveValidAbilities
/!\ eliminate event => should happen before discarding all cards or
changing turn
/!\ Next player is matched by looping initial order
Triggered startTurn
Triggered nextTurn
DatabaseUpdater
EventMatcher(duration, emoji) **events.json**
MoveSelector Tree
/!\ Cannot play jail against sheriff
/!\ Cannot gain reward if self eliminating as outlaw
/!\ Cannot start turn if hitNotEmpty
/!\ Cannot drawsAllCardsFromEliminatedPlayer if is eliminated
Anim Renderer
AnimationMatcher
MediaMatcher
/!\ Fix move evaluation

/!\ fix discard (prison) as help, discard any weapon as attack, discard dynamite as neutral

/!\ PlayBangOnDuel, PlayBangOnIndians as Bang card ability => We want separate play(card) VS discard(card)

Engine AbilityMatcher. ActiveMoves(in: state)

Engine PlayReq isHitName refactor

Engine PlayReq onDiscardHand remove