<<interface>> iBid

+ bids: Int

+ executeBidBonus(): Int

Extends

iBidDefault

<<interface>> iFieldDependent

+ bp: Int
+ bid: Int

+ fieldCheck(string cardName): Bool

Extends

iFieldDefault

Amour Card

1

+ bid: int
+ bidPlus: iBidBehavior

Ally Card

1

1

+ bp: Int
- bids: Int
+ questCheck: iQuestDependent
+ cardCheck: iFieldDependent
+ bidPlus: iBidBehavior

Test Card

1

+ min: Int
+ questCheck: iQuestDependant

<<interface>> iQuestDependent

+ bp: Int
+ bid: Int

+ questCheck(string Quest): Bool

Extends

iQuestDefault

Turn

+ players: List<Player>
+ storyCard: Card

+

Game Controller

+ players: List<Player>
+ nextTurn: Turn
+ turnCount: Int
+ storyDeck: Deck
+ adventureDeck: Deck

+

1

2,..4

2

Deck

+ type: String
+ count: Int
+ deck: List<Card>
+ discard: List<Card>

1.,...*

Rank Card

+ bp: int

Extends

Card

+ type: String
+ name: String

Extends

Extends

Extends

Extends

Extends

Extends

Extends

Extends

Weapon Card

+ bp: Int

Foe Card

+ minBP: int
+ maxBP: int

Quest Card

+ stages: Int
+ foe: String
+ quest: iStoryQuest

1

Tournament Card

+ shields: Int
+ tournament: iStoryTourn

1

iStoryTour

+ shields: Int

+ execute(List<Player

Event Card

+ event: iStory

Player

+ name: String
+ score: Int
+ rankCard: RankCard
+ rankDeck: Deck
+ strategy: iStrategy
+ List<Card> hand
+ List<AllyCard> activeAllies

+

1

1

<<interface>> iStrategy

+ field: Type

+ executeFoeStrategy(int bp, List<Card> hand): List<Card>
+ executeTestEncounter(int bid, List<Card> hand): List<Card>
+ executeTournamentEncounter(List<Card> hand): List<Card>

Extends

Extends

iStrategyCPU

iStrategyPlayer

iStoryQuest

+ stages: Int
+ foe: String

+ execute(List<Player> players): void

nament

> players): Type

Extends

Extends

Queen's Favour

+ execute(List<Player> players): void

Pox

+ execute(List<Player> players): void

King's Call to

+ execute(List<Player>

```
                                          ┌─────────────────────────────────┐
                                          │       King's Recognition        │
                                          ├─────────────────────────────────┤
┌─────────────────────────────┐          ├─────────────────────────────────┤
│     <<Interface>> iStory     │          │ + execute(List<Player> players): void │
├─────────────────────────────┤          └─────────────────────────────────┘
├─────────────────────────────┤
│ + execute(List<Player> players): void │  ┌─────────────────────────────────┐
└─────────────────────────────┘          │   Prosperity Through the Realm   │
                                          ├─────────────────────────────────┤
                                          ├─────────────────────────────────┤
                                          │ + execute(List<Player> players): void │
                                          └─────────────────────────────────┘
```

Interface: `<<Interface>> iStory`
- `+ execute(List<Player> players): void`

King's Recognition — Extends
- `+ execute(List<Player> players): void`

Prosperity Through the Realm — Extends
- `+ execute(List<Player> players): void`

Court Called To Camelot — Extends
- `+ execute(List<Player> players): void`

Arms — Extends
- `players): void`

Chivalrous Deed — Extends
- `+ execute(List<Player> players): void`

Plague — Extends
- `+ field: Type`
- `+ execute(List<Player> players): void`

Extends (×7 relationships)