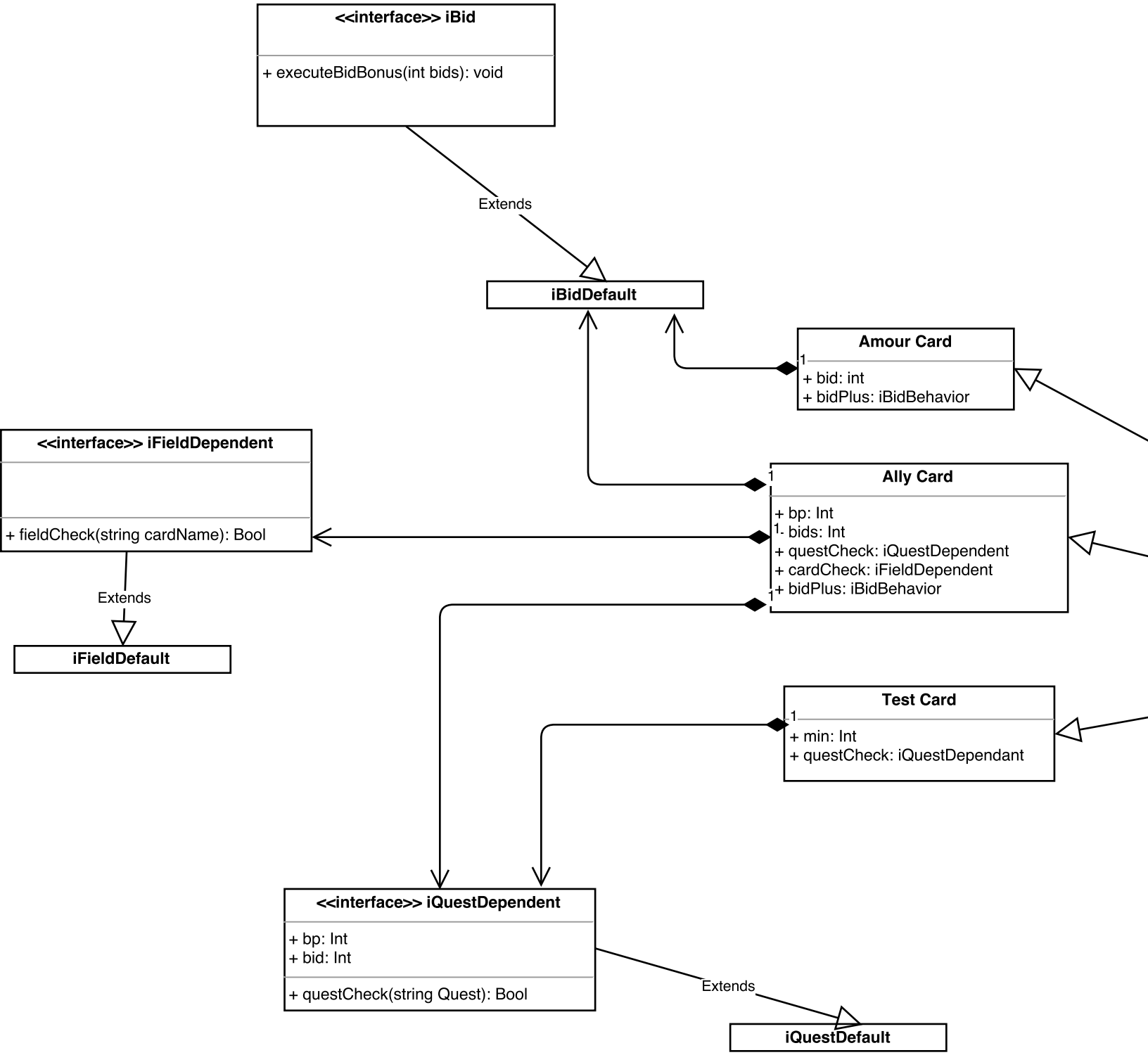
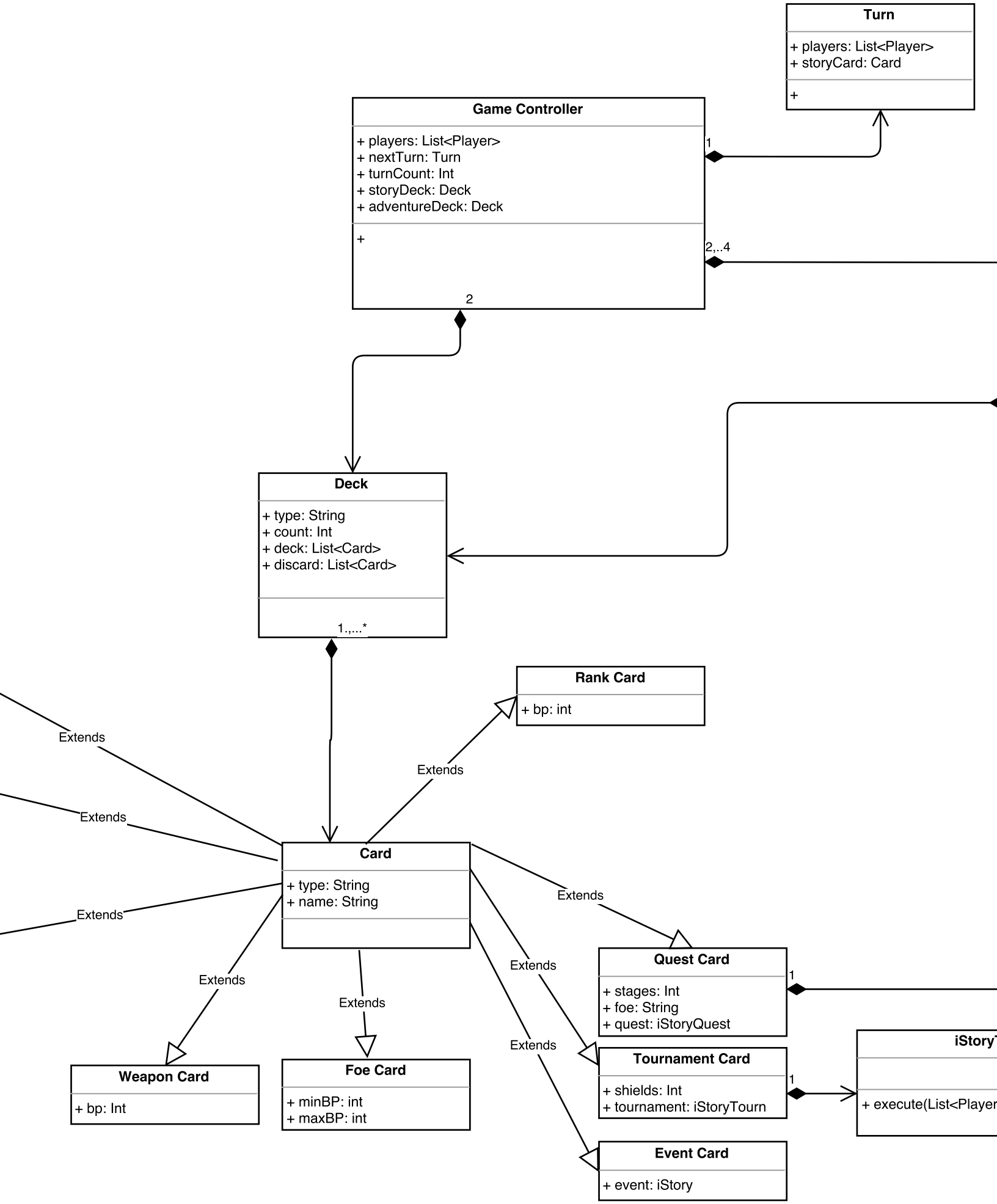
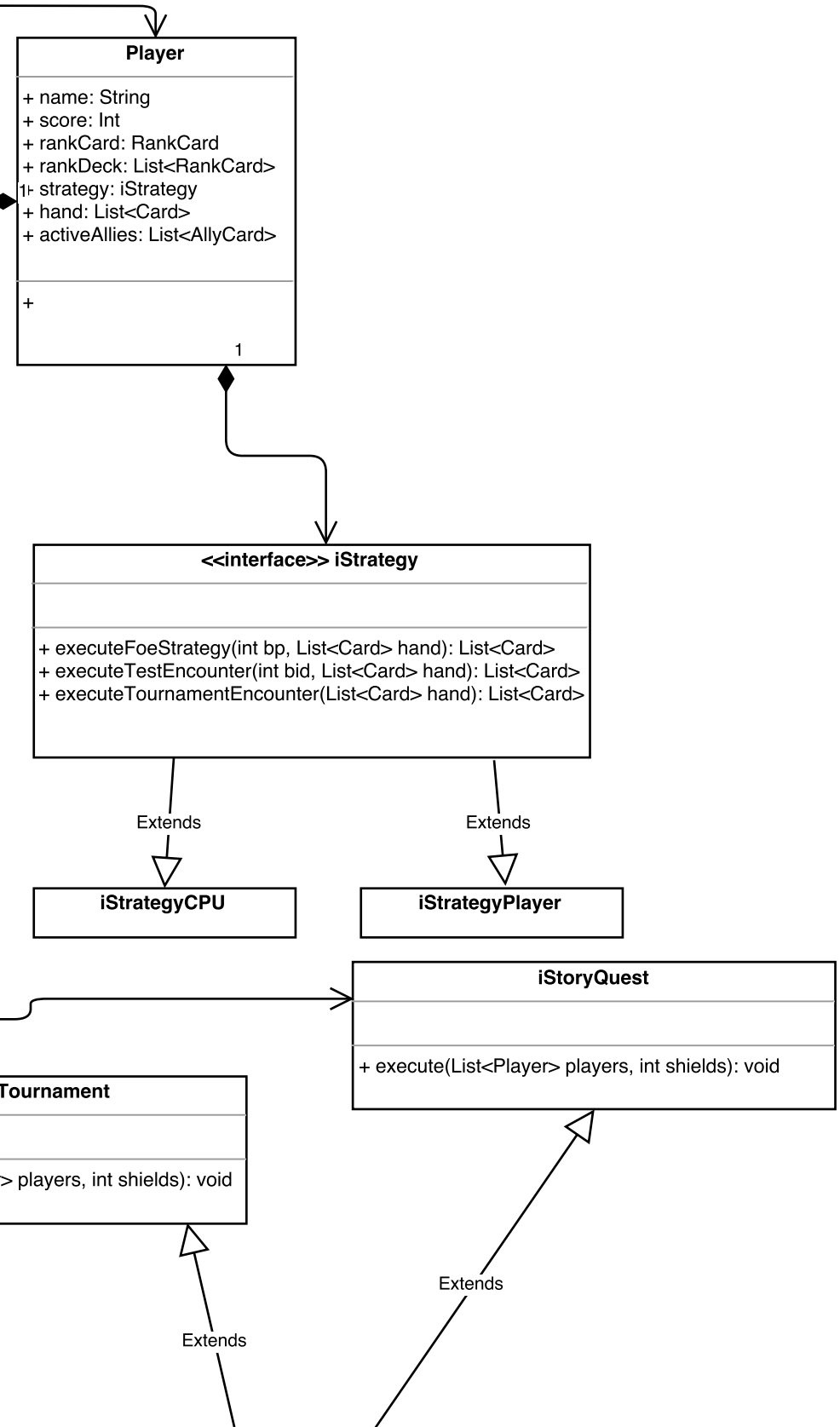
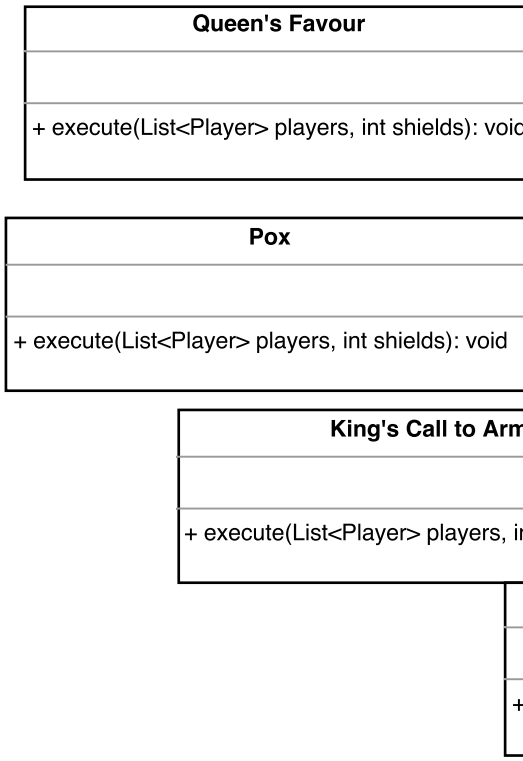
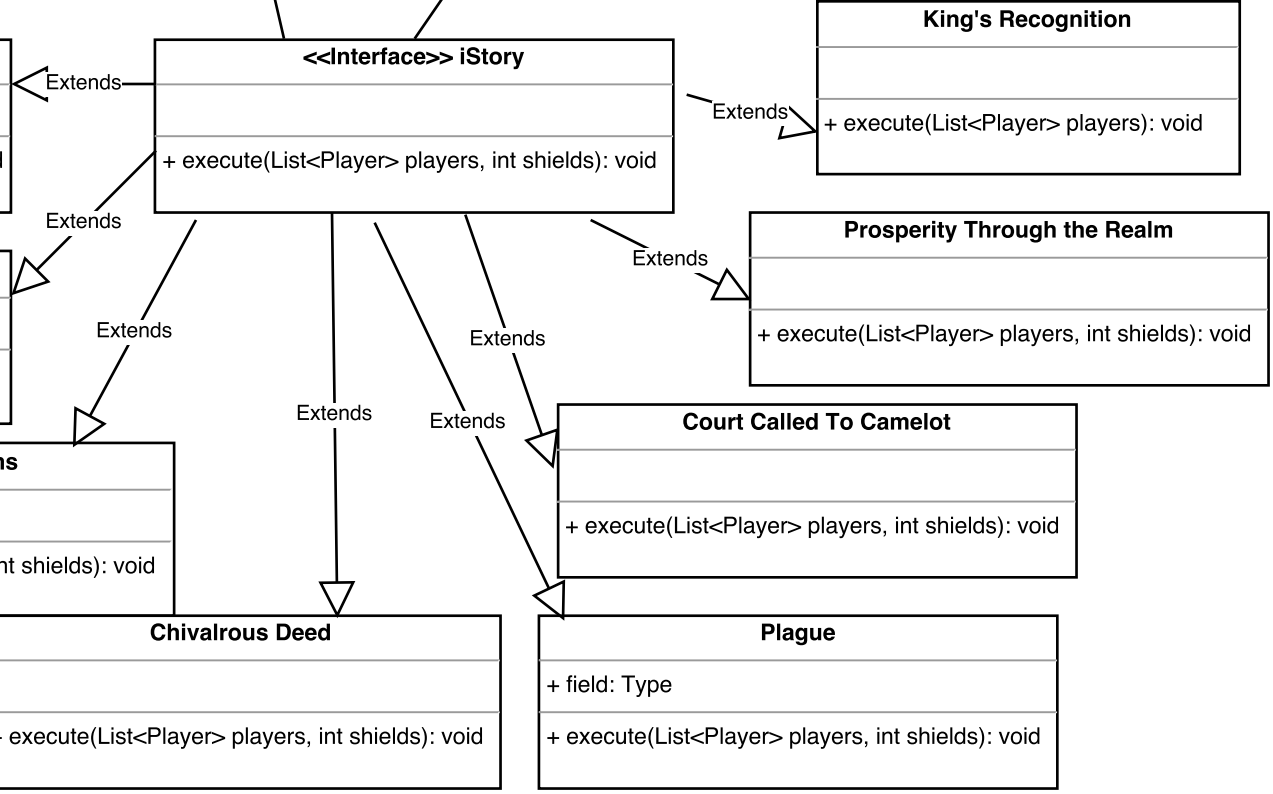

<<interface>> iBid
+ executeBidBonus(int bids): void
Extends
iBidDefault
Amour Card
1
+ bid: int
+ bidPlus: iBidBehavior
<<interface>> iFieldDependent
+ fieldCheck(string cardName): Bool
Extends
iFieldDefault
1
Ally Card
+ bp: Int
1- bids: Int
+ questCheck: iQuestDependent
+ cardCheck: iFieldDependent
1+ 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
Extends
Extends
Extends
Extends
Card
+ type: String
+ name: String
Extends
Extends
Extends
Extends
Extends
Quest Card
+ stages: Int
+ foe: String
+ quest: iStoryQuest
1
Tournament Card
+ shields: Int
+ tournament: iStoryTourn
1
iStory
+ execute(List<Player
Weapon Card
+ bp: Int
Foe Card
+ minBP: int
+ maxBP: int
Event Card
+ event: iStory


Player
+ name: String
+ score: Int
+ rankCard: RankCard
+ rankDeck: List<RankCard>
1 + strategy: iStrategy
+ hand: List<Card>
+ activeAllies: List<AllyCard>
+
1
<<interface>> iStrategy
+ 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
+ execute(List<Player> players, int shields): void
Tournament
> players, int shields): void
Extends
Extends


Queen's Favour
+ execute(List<Player> players, int shields): void
Pox
+ execute(List<Player> players, int shields): void
King's Call to Arm
+ execute(List<Player> players, i


King's Recognition
+ execute(List<Player> players): void
<<Interface>> iStory
+ execute(List<Player> players, int shields): void
Extends
Extends
Extends
Extends
Extends
Extends
Extends
Extends
Prosperity Through the Realm
+ execute(List<Player> players, int shields): void
Court Called To Camelot
+ execute(List<Player> players, int shields): void
Chivalrous Deed
+ execute(List<Player> players, int shields): void
Plague
+ field: Type
+ execute(List<Player> players, int shields): void