Sorcery UML Diagram main() Adeline Su, Julia Zhu, Sherry Feng controller, GameController MVC Architecture GameController - gm, GameManager - td, TextDisplay * Card * gd, GraphicsDisplay - name, string SUBJECT SUBJECT - type, CardType // enum class: Spell, Minion, + go() // to start the game OBSERVER DESIGN Enchantment or Ritual PATTERN - desc, string **OBSERVER** GameManager // any other methods??? - observersDisplay, std::vector<Display*> - observersTrig, std::vector<TriggeredAbility*> - p1, Player + getName(), string + getCost(), int - p2, Player displayMsg(), void displaySorceryBoard(), void td, TextDisplay gd, GraphicDisplay DECORATOR DESIGN PATTERN activePlayer, Player* displayMinion(), void - nonactivePlayer, Player* * Spell * * Minion * * Enchantment * * Ritual * - activationCost, int + startTurn(), void - charge, int - trigAbility, TriggeredAbility + endTurn(), void + attackMinion(), void + attackPlayer(), void // do sth + getAttack(), int TextDisplay GraphicsDisplay + getDefense(), int + activateAbility(), void // "use" + getAbility(), int + getAbility(), Ability theGDisplay, Xwindow + discard(), void + play(), void // update observers if needed - borderSize, const int describe(), void sorceryXCoord, const int + hand(), void ... // many other coordinates, measurements + board(), void + notifyObserversTrig(), void + attatchObserversTrig(), void * EnchantmentDec - attack, int # next, Minion* - defense, int - action, int - ability, Ability Player - name, string - id, int // 1 or 2 (not sure if needed) + getAirElemental(), DefaultMinion* + getEarthElemental(), DefaultMinion* * Ability * - magic, int - deck, Deck + ... // each minion targetPlayer, std::vector<Player*>targetMinion, std::vector<Minion*> hand, Hand - player, Player* - board, Board - grave, Graveyard GiantStrength Enrage Haste ritual, Ritual* + applyAbility(), void +play(), void OBSERVER DESIGN PATTERN OBSERVER ActivatedAbility * TriggeredAbility * MagicFatigue Graveyard type, TriggerType // enum class Deck Hand Board - theDeck, std::vector<Card*> theHand, std::vector<Card*> - theBoard, std::vector<Minion*> - theGrave, std::Stack<Minion*> + applyAbility(), void // notify + applyAbility(), void + drawCard(), Card BoneGolemTrigAbility FireElementalAbility DarkRitualAbility AuraofPowerAbility StandstillAbility PotionSellerAbility NovicePyromancerAbility MasterSummonerAbility ApprenticeSummonerAbility