

Javadoc

Main.java - Main Program

The main program that handles all of ui and ux design as well as handling objects from other class constructors (game, players, etc)

Class Main
Properties Int intPort Int intHelpMenuPage Timer Maintimer String strP1Name String strP2Name Game game Boolean blnIsHost JFrame theMainFrame JPanel MainMenuPanel JButton HostButton JButton JoinButton JTextField nameField JTextField IPAddressField JTextField PortField JButton StartGameButton JLabel StatusLabel JButton returnMenuButton
public void SwitchTabs(JPanel thePanel) public void SendSystemMessage(String message) public BufferedImage getImage(String strImagePath) public void StartGame(){ public void actionPerformed(ActionEvent event){ public void keyTyped(KeyEvent event){ public void keyPressed(KeyEvent event){ public void keyReleased(KeyEvent event){ public void focusGained(FocusEvent event){

Game.java - Game class

Main.java utilizes this class for the main gameplay and mechanics, such as phase changing, etc.

Class Game	
Properties	
private PlayerClass p1 private PlayerClass p2 public boolean bInStarted public boolean isInitializationPhase private String currentPhase private JAnimation animationPanel private SuperSocketMaster ssm private Main mainInstance private boolean bInIsHost private List<AttackAction> attackQueue private boolean isProcessingAttacks private int intCurrentAttackIndex String strBigDeck[][] String strEvoDeck[][] String strP1Deck[][] String strP2Deck[][] String strSqDeck1[][] String strSqDeck2[][] int intCount String strLine int intRandom int intRow int intRow2 String strTempName, strTempCost, strTempHP, strTempAttack, strTempSigil, strTempOrder boolean bInDecksFilled int intRow3 int intRow4 int intRow5 BufferedReader thefile	
Methods	
public void startGame(){ private void initializeDecks(){ public void nextPhase(){ public boolean playerDrawCard(int intPlayerNumber){ public boolean playerDrawSquirrel(int intPlayerNumber){ public void playerReady(int intPlayerNumber){ public String getCurrentPhase(){ public PlayerClass getP1(){ public PlayerClass getP2(){ private void executeAttackPhase(){ public void executeAttackPhaseFromNetwork(){ private void processNextAttack(){ public void onAttackAnimationComplete(){ private void completeAttackPhase(){ private void performAttack(PlayerClass attacker, PlayerClass defender, int intSlotIndex,	

private PlayerClass p1
private PlayerClass p2
public boolean bInStarted
public boolean isInitializationPhase
private String currentPhase
private JAnimation animationPanel
private SuperSocketMaster ssm
private Main mainInstance
private boolean bInIsHost
private List<AttackAction> attackQueue
private boolean isProcessingAttacks
private int intCurrentAttackIndex
String strBigDeck[][]
String strEvoDeck[][]
String strP1Deck[][]
String strP2Deck[][]
String strSqDeck1[][]
String strSqDeck2[][]
int intCount
String strLine
int intRandom
int intRow
int intRow2
String strTempName, strTempCost, strTempHP, strTempAttack, strTempSigil, strTempOrder
boolean bInDecksFilled
int intRow3
int intRow4
int intRow5
BufferedReader thefile

public void startGame(){
private void initializeDecks(){
public void nextPhase(){
public boolean playerDrawCard(int intPlayerNumber){
public boolean playerDrawSquirrel(int intPlayerNumber){
public void playerReady(int intPlayerNumber){
public String getCurrentPhase(){
public PlayerClass getP1(){
public PlayerClass getP2(){
private void executeAttackPhase(){
public void executeAttackPhaseFromNetwork(){
private void processNextAttack(){
public void onAttackAnimationComplete(){
private void completeAttackPhase(){
private void performAttack(PlayerClass attacker, PlayerClass defender, int intSlotIndex,

```
CardClass attackingCard, boolean isBottomAttacking){  
private void checkScaleDifference(){  
public boolean playCard(int intPlayerNumber, CardClass card, int intCardSlot){  
public void syncReturnToDrawing(){
```

PlayerClass.java - Player Class

Class CardClass

Properties

```
public String strName  
public BufferedImage imgCardImage;  
public int intHealth;  
public int intAttack;  
public int intCost;  
private SigilClass sigil = null;  
public boolean blnRevealed = false;
```

Methods

```
public void attack()  
public void sigilActivate(PlayerClass player, PlayerClass opponent) {  
public SigilClass getSigil() {  
public void setSigil(SigilClass sigil) {  
public boolean hasSigil() {  
public CardClass(String strName, BufferedImage imgCardImage, int[] arrStats, SigilClass  
sigil) {  
public CardClass(String strName, BufferedImage imgCardImage, int[] arrStats, String strSigil)  
{  
private SigilClass parseSigil(String strSigil) {
```

SigilClass.java - Abstract Class extends CardClass

Class

SigilClass extends CardClass

Properties

```
protected String strDescription;  
protected String strName;  
protected String strSigilActivation
```

Methods

```
public abstract void activateSigilEffect(CardClass card, PlayerClass player, PlayerClass  
opponent, int intSlotIndex);  
public String getName(){  
public String getDescription(){  
public String getActivationPhase(){
```