

Woo

INSTANCE VARIABLES

```
ArrayList<String> images // paths of a bunch of images of cards
ArrayList<Integer> positions // [xcor,ycor] of each picture in images
ArrayList<Integer> attackingCards // tracks which cards are attacking/being attacked
PImage bg; // background image
boolean bover = false; // true whenever the mouse is over an image
boolean locked = false; // true if the image is being dragged
ArrayList<PImage> image1 = new PImage [images.length];
float newx, newy; - // new coordinates of mouse after the dragging
int whichImage; - // which image the mouse is currently over
boolean p1turn=true // true when first player's turn, false otherwise
boolean p2turn=false // opposite of p1turn
Player nicolas // bottom player
Player chris // top player
boolean spellMode // checks if the user is going to use a spell
int turnCounter // counts number of turns
boolean printEndOnce // used for printing last statement only one time in draw
```

METHODS

```
void setup() -sets up images and positions
void draw() - sets up background, border squares, images
void mousePressed() - changes locked based on bover
void mouseDragged() - draws the reprinting of the moving image, also adds a green border
void mouseReleased() - sets locked back to false when the user stops dragging
void mouseClicked() - adds to attackingCards and puts a border on selected cards, also used
for ending turn and loading weapon
void keyPressed() - used to execute actual attack sequence when attackingCards is full
void printCurrentHand1() - prints currentHand on first turn for nicolas
void printCurrentHand2() - prints currentHand on first turn for chris
void printCurrentHanda() - prints currentHand on non-first turn for nicolas, having the first two
methods only ran into some NullPointerExceptions
void printCurrentHandb() - prints currentHand on non-first turn for nicolas, see
printCurrentHanda
void fill() - fills each deck with basic monster cards
int cardOver() - returns index of card in images (arrayList) that is over a specific monster slot,
1 method for each slot
int cardOverWep() - returns index of card in images (arrayList) that is over the weapon slot, 1
method for each slot
void processCards1() - processes monsters on nicolas's side, fills nicolas.monsters with the
cards over the slots
```

void processCards2() - processes monsters on chris's side, fills chris.monsters with the cards over the slots  
void displayMonsters1() - displays nicolas's monsters after they are registered, not draggable so they don't belong in the other display mechanism  
void displayMonsters2() - displays chris's monsters after they are registered, not draggable so they don't belong in the other display mechanism  
void fillAttackingCard1() - fills first attacking slot, card will have a red border on the field  
void fillAttackingCard2() - fills second attacking slot, card will have a green border on the field  
void fillSpellCard() - fills attacking spell slot, card will have a yellow border on the field  
Card attackMonster(Card attacker, Card receiver) - changes HP and mana based on attacker, Card returned is what the receiver becomes after the attack  
void loadWeapon() - loads weapon for each player, 1 for each player  
void pallyPower() // paladin Hero Power  
void shamanPower() // shaman Hero Power

Player
<p>INSTANCE VARIABLES</p> <ul style="list-style-type: none"> <li>- int currentHP // current HP</li> <li>- int maxHP // max HP (so effects won't heal past the max)</li> <li>- int currentMP // current MP a player has this turn</li> <li>- int maxMP // max MP a player has for the game (increases each turn)</li> <li>- int deckDamage // damage taken if there are no more cards to draw</li> <li>- ArrayList&lt;Card&gt; currentHand // current hand of player</li> <li>- ArrayList&lt;Card&gt; monsters // monster cards currently on the board</li> <li>- ArrayList&lt;Card&gt; deck // deck of cards for the user</li> <li>- Card weapon // current weapon that is equipped</li> <li>- boolean usedHeroPower // checks if HeroPower has already been used this turn</li> </ul>
<p>METHODS</p> <p>Player() - sets currentHP to 30, maxHP to 30, current MP to 0, maxMP to 1</p> <p>int size() - returns size of currentHand</p> <p>void drawCard // draws a card from deck</p> <p>void addMonster // add a monster to the board</p> <p>void incHP // healing player</p> <p>void decHP // player takes damage</p> <p>void incMP // increase maxMP each turn</p> <p>void decMP // decrease currentMP when a card is placed</p> <p>String hpLine() - returns currentHP / maxHP</p> <p>String mpLine() - returns currentMP / maxMP</p> <p>String toString() - returns hp,mp and weapon data of player</p>

Card
<p>INSTANCE VARIABLES</p>

PIImage picture - loaded picture of card  
String path - path of card  
int value - damage/restore number of card  
int type - monster/spell/weapon  
int curentHP - a card's current health  
int maxHP - a card's max health  
int cost - mana cost  
boolean shown - only shows your hand if it is your turn  
Boolean attackedthisTurn - checks if card attacked this turn already

#### METHODS

Card (PIImage portrait, String paf) - initializes HP, cost, etc., based on paf  
String toString() - prints HP, MP, and name of card  
void incHP(int value) // increases HP by value  
void decHP(int value) // decreases HP by value  
void attackMonster() // monster attacking a monster  
void attackPlayer() // monster attacking a player