Señor Power APCS2 pd 4
Nicolas Naing, Chris Yu 6/1/17

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
float x_offset_drag; // drag variable that makes movement smoother
float y_offset_drag; // drag variable that makes movement smoother
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

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 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, 14 of these methods 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

Card attackMonster(Card attacker, Card receiver) - changes HP and mana based on attacker, Card returned is what the receiver becomes after the attack

Player

INSTANCE VARIABLES

- int currentHP // current HP
- int maxHP // max HP (so effects won’t heal past the max)
- int currentMP // current MP a player has this turn
- int maxMP // max MP a player has for the game (increases each turn)
- ArrayList<Card> currentHand // current hand of player
- ArrayList<Card> monsters // monster cards currently on the board
- ArrayList<Card> deck // deck of cards for the user

METHODS

Player() - sets currentHP to 30, maxHP to 30, current MP to 0, maxMP to 1

int size() - returns size of currentHand

void addMonster // add a monster to the board

void incHP // healing player

void decHP // player takes damage

void incMP // increase maxMP each turn

void decMP // decrease currentMP when a card is placed

void attackMonster(Card monster, int damage) // player attacks monster with a weapon

void attackDir(int damage)

Card

INSTANCE VARIABLES

PImage picture - loaded picture of card

String path - path of card

int value - damage/restore number of card

enum type - spell/monster/weapon

int hp - amount of hp on card (monster only)

METHODS

Card (PImage pikture, 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 monsterAttack() // monster attacking a monster
```