

Hearthstone Card Recognizer

Board recognizer for the game Hearthstone using image processing methods.

Authors

- Danilo de Moraes Costa - NUSP 8921972
- Guilherme Zanardo Borduchi - NUSP 8937458
- Lucas Silveira de Moura - NUSP 8937267

Project Objective

Hearthstone is a popular collectible card game, launched in 2014, with more than 70 million unique players. With more than 1000 playing cards, our project aims to process an image from the playing board, and try to detect which cards you are holding, which monsters and weapons are on board and how many cards your oponent has, from which point it will be possible to present various informations to the user, such as who is currently winning the game, what combination of cards can be played and how can the monsters interact.

Input Images

The input images will be composed of screenshots of games being played, and also a large database containing every playable card in the game, to be able to detect them on the screenshots.

The screenshots will be obtained by playing the game and capturing them in different situations (for example, when the board is full of monsters and also when it is empty). We don't need a large number of screenshots, since the biggest challenge of this project is identifying which of the more than 1000 playable cards are in play.

The database of cards will be created via a simple Python scraper for the [Hearthpwn](#) website, which gets the URLs for the images on the site and saves them on the authors' computers.

Figure 1 is an example of the screenshots that will be used. Note how the monsters can have different shapes and particle effects, and also how the cards in your hand get rotated and more grouped together as you draw more.

On Figure 2 we have an example of the images that will compose our card database. They are going to be matched against the screenshots previously shown. Also note how the scales of the images are very different, and how once a monster is played, only its picture is shown on the board (creating another challenge).



Figure 1: The board of the game Hearthstone.



Figure 2: One of the playable cards in Hearthstone.

Reaching the Objective

To reach our objective, we first have to scrape the [Hearthpwn](#) website and create the database of cards that will be matched against the screenshots. This is done once, and from then on we simply use the saved images.

We need to test each of the 1000 card against the scene to see which ones are present. We will probably need another form of processing to reduce this number, since the matching operation is very costly (suggestions needed).

Since our project deals with matching images on a scale, color and rotation variant environment, we intend on using keypoint detectors (SIFT, SURF or ORB) to extract the relevant points on the image, and then match them using Brute-Force or FLANN-based matchers. Finally, if enough matches are present, we consider that the object is in the scene.

Once we have information about which cards are on the scene, we can then use simple logic to determine stuff such as who is currently winning and possible plays that can be made.

First Results

We already coded and tested the scraper. We just need to run it for all the 1000 cards, which is going to take some time.

We also implemented a prototype that matches a single card against a scene, with varying results. Some cards were matched successfully, even with the rotation, scaling and color variants we mentioned. Other cards, however, could not be matched, and we intend on improving this prototype to minimize the misses (cards that aren't matched). The last problem is cards that are mismatched. This is even worse than not finding a card, since it will lead to wrong statistics.

We give two examples of our prototype below. The green lines connect keypoints on the object to the same keypoints found on the scene. The white rectangle denotes the object the algorithm matched on the scene.

Figure 3 is an example of a successful match. The card was found on the hand of the player.

Figure 4 is an example of a mismatch. The card we want to match is on the scene, under the big "Lyla the Sunshard" card, but the algorithm mismatched it with another card that shares some characteristics, such as same attack and health.



Figure 3: Successful match.

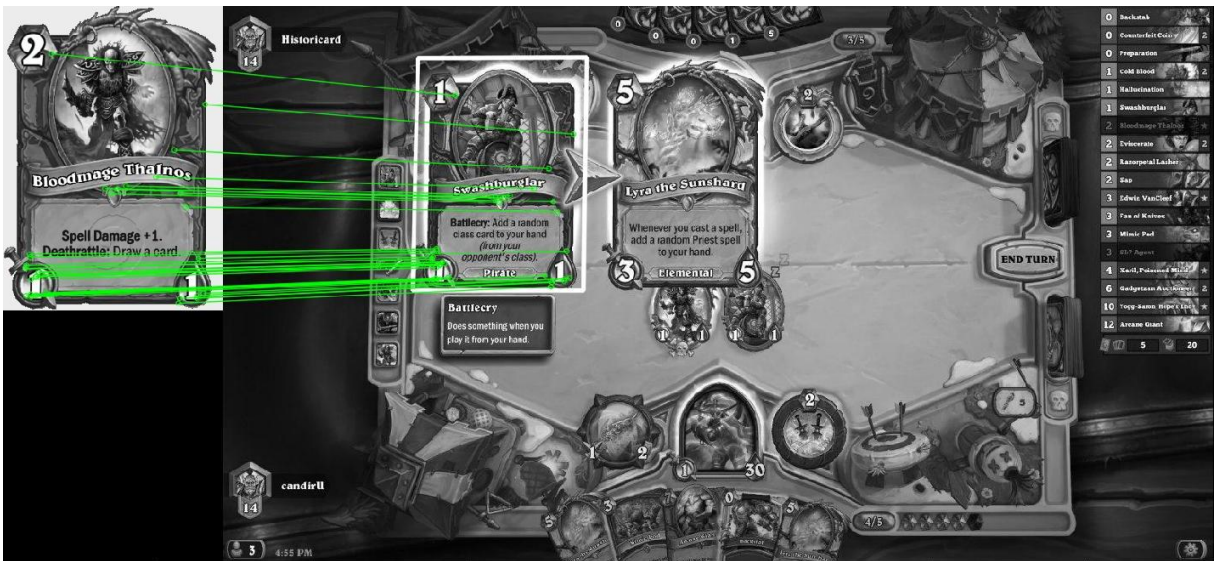


Figure 4: A mismatch.