UNIVERSITY OF SÃO PAULO
INSTITUTE OF MATHEMATICS AND COMPUTER SCIENCES
Gustavo Bouzaz Paixão - 8936947
Luis Paulo Falchi Justino - 8937479

IMAGE PROCESSING PROJECT
FINAL REPORT

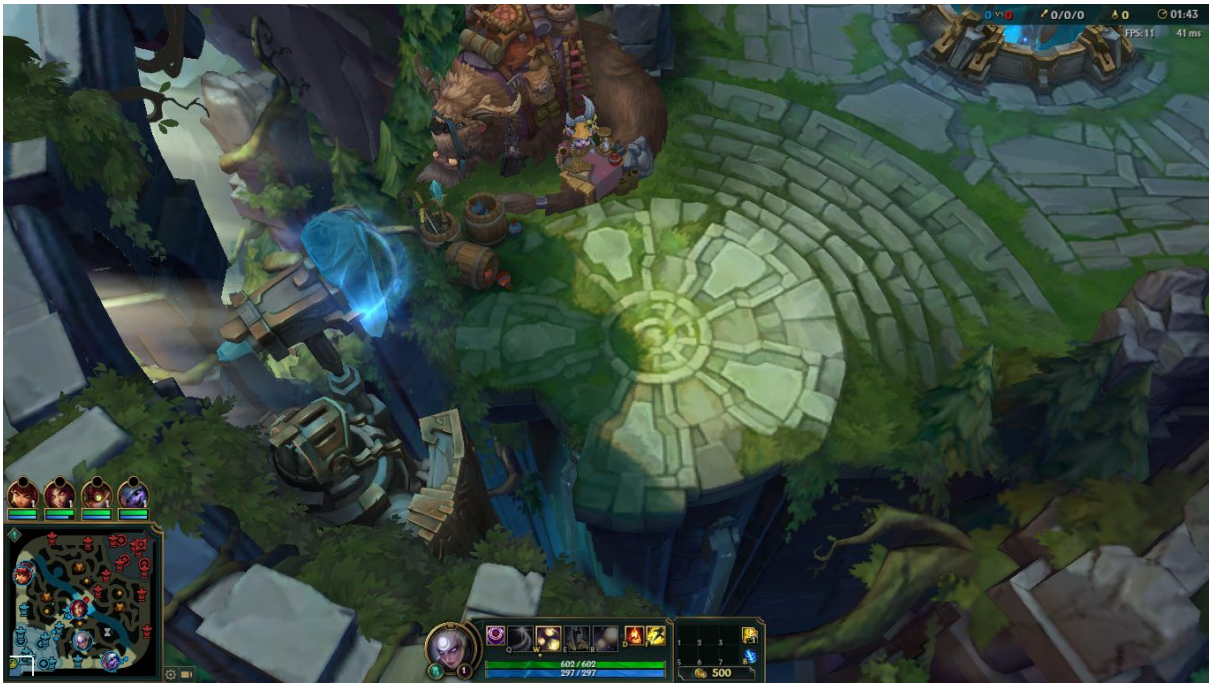
São Paulo
June 2017

1. MAIN OBJECTIVE

Event detection in a match of League of Legends, from the treatment of images of the game's minimap, obtained in real time.

2. INPUT IMAGES

There are two types of images used in this project. The first one is a simple screenshot of the game in real time. Below, there is an example:



The other one is the portrait (or square) of a champion (character of the game), that can be downloaded at https://developer.riotgames.com/static-data.html. An example:

3. PROJECT STEPS

First of all, all champions portraits were downloaded and for each image obtained in a game:

1. Trim it to get the game's minimap, an 200x200 image;
2. Identify the champions in the match, using the portraits as comparison models;
3. Once we found the champions, we can obtain their coordinates in the map;
4. With the coordinates, identify events in the match (objective conquests and *ganks*).

4. RESULTS

In the beginning, the portraits were reshaped and cut into circles in order to compare them with the return of the OpenCV's function *HoughCircles*. Below, there are two images after applying this function:

As we can see, this method did not found all possible champions, given the circle in the center of the second image wasn't identified. There are circles that don't represent any of the champions and these ones are returned randomly.

Even with wrong returns, the circles were compared with the processed portraits to identify the champions in the match, but the results weren't the expected, probably because of the circles poor resolution. In order of that, we decided to try another method.

In a second attempt, we chose the Template Matching (function *matchTemplate*) from OpenCV to find the characters. This function has a parameter for a method, which can be: CV_TM_SQDIFF, CV_TM_SQDIFF_NORMED, CV_TM_CCORR, CV_TM_CCORR_NORMED, CV_TM_CCOEFF or CV_TM_CCOEFF_NORMED. Tests were realised for each method.

Consider the following models for comparison:

These are, respectively: Fiora, Ahri, Alistar, Soraka and Brand.

Also consider the following minimap picture:

After applying the function for different templates and using the CV_TM_CCOEFF method, we obtained the following images as results:

The first four of them are correct, once Ahri, Soraka and Brand were found and Fiora wasn't, respectively. As Alistar's model was used as template for the last image, it is wrong.

Below, there are output images where the method is different for each one of them (CV_TM_SQDIFF_NORMED, CV_TM_CCOEFF, CV_TM_CCOEFF_NORMED, CV_TM_CCORR, CV_TM_CCORR_NORMED, CV_TM_SQDIFF, respectively), also considering Alistar's model as template.

Only in the first image Alistar was found.

5. CONCLUSION

We got good and bad results. But there were some problems that hampered the development of the project, like:

- Low resolution of the images may affect the accuracy of the methods;
- Obtain a collection of images in real time;
- Minimap images may vary a lot, considering that there is shadows, towers, minions, monsters and other elements in the game;
- In cases where there is champion overlapping, also there is loss of information, making it impossible to identify some champions in the minimap. The image below has two cases of champion overlapping: