

Reimplementing a Strategy Game Interface

Manuel Menezes Morais
manuel.menezes.morais@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2015

Abstract

Almansur is a multiplayer turn based strategy browser game developed throughout the last decade. This project aimed to create a new version of Almansur, focusing on improving the usability of the interface, making it easier to teach new users, and more efficient to use for the current ones. This document describes the development process of this new version. First the existing game was analyzed, creating a conceptual model, a list of possible tasks in the interface, and comparing it with other games of the same genre. This interface was then tested by player observation and with the Nielsen Heuristics to identify its current usability faults. A new interface was then created, first on paper, then in a simple interactive prototype, and finally in its full implementation, which iterated multiple times through development. Finally the resulting work was tested and compared with the previous version, using both old and new players to the game.

Keywords: Browser Game, Strategy Game, Usability, User Interface

1. Introduction

Almansur [1] is a turn based browser strategy game, built in HTML 4 and Flash, played in a similar way to others like the Sid Meier's Civilization series [2] or the Paradox Entertainment [3] Grand Strategy games series [4]. The current interface of Almansur is arguably the current biggest hurdle to new players, being confusing and outdated in some ways.

This work aims to improve the usability and flexibility of the user interface of Almansur, by making it more user-friendly, applying Human-Computer Interactions techniques, and recreating it on newer technologies, more specifically using HTML5 and WebGL.

To accomplish this task, the methodologies explained on [5] were followed, being necessary to:

- Analyze the current user interface of Almansur, listing requirements to replicate its functionality, and add any new needed ones;
- Do usability testing of the current interface, creating scenarios and metrics to be able to compare to what will be proposed;
- Analyze the user interface of similar games of the same genre, or that simply share some common interface situations, looking for design

patterns and state-of-the-art solutions for some of the problems of the current interface;

- Create and implement a prototype of the proposed user interface. This prototype will be iteratively tested and improved;
- Test both the new and the old versions of the interface with old and new users, applying the metrics defined before and comparing results between each version.

2. Related Work

This section analyzes the current interface of Almansur. For this, a conceptual model was created and the current possible tasks were analyzed. The interface was also compared with other games in the same genre or with similar traits, trying to find interface design patterns that can be used to improve the usability.

2.1. Conceptual Model

Fig. 1 describes the concepts, and the interactions between them, of the version 2 of Almansur in diagram form.

2.2. Player Experience

The main target audience of Almansur are people who play videogames at least occasionally and already had contact with strategy games. This includes subgenres such as: browser based strategy, strategy RPG, real time strategy, grand strategy

in some key areas, like the usage of map overlays, the overall layout of the interface, the way terrain is displayed on the map, the way borders between territories are represented, the way units are represented, how movements and/or orders are issued to units, how cities are managed, the way messages are exchanged between players, and how documentation and help is shown to the player. Between the tested games, the ones that were shown more relevant were Civilization III to V, Rise of Nations, Crusader Kings II, Europa Universalis IV, Cities: Skylines, Travian and Advance Wars.

The conclusion from this analysis is that for most situations each game has a different way of doing it, adapted to its rules, and as such cannot be directly mapped to Almansur. It can however be kept that displaying the user interface near the corner of the screen, map overlays and colored borders are good design patterns used among critically acclaimed games of the genre and can be implemented.

3. Usability Testing

To evaluate the current user interface of Almansur and identify its usability faults, the Nielson's Heuristic evaluation [6, 7] was used. It is usually considered that three to five evaluators are needed to find 70% of the usability faults [8, 9], being three used during this work.

The analysis resulted in a finding of 37 breaches of interface, spanning 7 of the Nielsen Heuristics: Consistency and Standards, Visibility of System Status, User Control and Freedom, Error Prevention, Recognition Rather than Recall, Flexibility and Efficiency of Use and Help and Documentation.

To find out more of the issues of the current players with the interface, a group of two long time veteran players and three players that had the game for a week, was also observed playing their turns. The problems found and criticisms were actually relatively similar between the veterans and newcomers, closely matching the Nielsen Heuristic analysis already discussed.

4. Interface Prototyping

The interface of the version 2 of Almansur mirrors the way the game has evolved over the years. Originally Almansur had 4 pillar areas, Land, Diplomacy, Economy and Military. The management page (which comes from version 1 of Almansur) is organized in these four areas, with the player being able to manage each one of them in the respecting pages. There was a 5th button at the time with the map, however the map was only to consult the surroundings and not to make major

actions on the game.

With version 2 of Almansur the map was reworked in a much more complex interface, with the player being able to filter the map and make actions over the the 4 areas mentioned before. The increase of information in the map made the game much easier to play, however this map does not completely implement all of the actions that were available in the management pages. The result is that there are ways to manage the 4 areas in the management page, but also some other ways to manage each of the four areas in the map, with some features only being available in one of the sides. Having this mix of old and new is what made the interface be in the convoluted state it was before this work.

The version 3 of Almansur, proposed in this thesis, aims to blur the lines between the 4 areas, removing the management pages and center all of the interface in the map, removing all external pages and make that info available in windows that appear on top of it. This offers a interface more in line with desktop games and less with browser games, making it both more familiar to expert players of the genre, and to newcomers due to info being more accessible in the map.

The new interface also has to support tablets. Although its not specifically designed for them, this means that we had to make sure that all of the interface and interactions work on a touchscreen, and the mouse-overs are not essential to play the game.

The new concept for the interface stated on was initially explored through a paper prototype, where the conclusion to center it on the map was reached. This prototype already had most of the concepts of the interactive prototype, however it was not finished and quickly scrapped due to the dynamic nature of the interface and the large quantity of screens that would be needed to be drawn.

After having the main ideas lined out in paper, the concept has been put to test by creating a interactive prototype of them. This has been done with JustInMind Prototyper [10], and the result can be consulted on [11].

The new interface, although seamless in transition, is composed of two views, the map view and the city view. These views share most of the interface elements, having other specific ones depending on the context. In the map view the player sees, through different overlays of information, the sta-

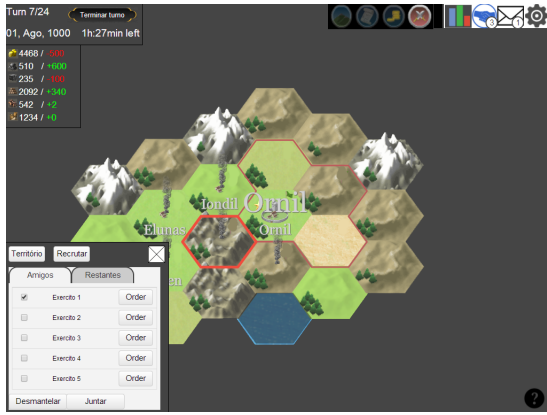


Figure 2: Sample image of the new Almansur interface prototype on the Map View.



Figure 3: City view on the prototype

tus of his currently explored or owned territories (Fig. 2), and on the city view the player can see and manage each specific territory and its buildings (Fig. 3). Common from both views are elements to change the game settings, message other players, do diplomacy and check statistics on the top-right, and elements to see the state of the game and the current player resources on the top left.

5. Implementation

This section describes the results of implementing the prototype described in section 4.

5.1. Architecture

The version 2 of Almansur is implemented using a MVC architecture[12], built on Ruby on Rails [13]. All of the typical webpages, which includes: the initial pages before login that promote the game, the account pages, and the ingame management pages, are common pages created in a typical MVC architecture, where the user makes a HTTP GET request to a controller, the controller gets data from models, then feeds that data into views to generate a HTML/CSS/Javascript page, and this page is returned to the user browser. Any link or

form used in these pages makes the browser change context and repeat the process.

There is however one page that works in a different way, which is the page that has the map. This is loaded in a normal way, however the page contains a Flash app[14] for the map. This app, after the page is loaded, then proceeds to load a series of what we will call streams, to get the necessary information to build the map and its interface. Each stream is an asynchronous call to the server controllers or helpers (ajax) returning information in a proprietary plaintext format. After the map is fully loaded, all interaction will then proceed in one of two ways: or also using ajax to send a POST or GET, using the response to update the map, or using a normal http call which will change the browser back to one of the normal pages.

For the version 3 of Almansur, both the management pages and the flash map page were removed. Instead, the map pages were added. These are single pages which focus on having a map with the size of the screen, and a interface on top of it generated with javascript and ajax calls, to offer the behavior explained on 4 and shown on Fig. 4. This way the user has the perception of always staying in the same page, creating a more intuitive interface. These pages use a expanded version of the same stream api of the flash map for updating the map, to not break compatibility with the artificial intelligence. The interface parts are built or directly in the javascript code using jQuery, or using ajax to get html code from a controller.

Each map page is a response of a normal HTTP GET that comes with the Pixi.js framework[15], the maplib, which is a library of javascript code that is shared amongst all map pages, and specific javascript code for that map. Currently only two of these pages exist. These are the game_map, which is basically most of the new ingame interface, and the turn_map, which shows the turn log, which will later be explained on 5.2. This was however done because there are multiple other situations where the maplib can (and will) be used, like in the admin view of the map or to create a map editor for the admins.

Pixi.js is a 2D rendering framework for web created around the concept of delivering great performance. This is used in Almansur to create a scenegraph and manage the graphic objects life cycle. The main reason for choosing this framework is that it offers completely seamless fallback from WebGL to canvas, meaning a increased browser

support considering that WebGL implementations are recent[16]. This means however that the game will stop supporting some old browsers still in use, like Internet Explorer 8, due to not having canvas support[17].

The way that the maplib works is that it manages the interface life cycle alongside Pixi.js. The map specific code needs to implement multiple callbacks that are used by the maplib in the life cycle, and to override camera behavior when needed. The maplib also offers functions to: draw objects on the map, open windows on top of the map, display error and success messages, load the information streams from the server, and initialize the map based on the information streams that were loaded.

5.2. Iteration and Final Version

After collecting opinions from the users about the interactive prototype produced in section 4, development was started in the actual game interface (Fig. 4), which was heavily more time consuming than expected. Because the previous version map was implemented in Flash, and one of the objectives was to remove its dependency, the entire map had to be programmed from scratch, and this task took near 6 months of full time development (including testing and debugging).



Figure 4: Map view in the military overlay in the final version

The implementation was constantly iterated and shown to users for feedback during development, with some key areas that received major changes from what was shown in the prototype:

- The way armies are represented in the map had four different versions, ending in one that each army is represented by a single flag, where its style represents the size and the color represents the relation of the player with the owner of the army;
- How army information is represented in the interface was simplified. The tabs of the army listing of a territory were merged into a single

list, and when with a army selected the interface was also simplified from three tabs to two;

- Information was added to the map representing the pending orders from an army. Each stationary order received a new symbol that is shown on the territory where it is going to be executed. The movement orders now, besides the arrow, also show a small number with how long the movement will take, and the arrow is colored representing the speed of the movement. The movement is also now pathfinded showing the exact path it will take;
- A new map overlay was created with general information about each territory, like the population happiness or the crossing time;
- The message and diplomacy menus merged into a single menu;
- A new much more flexible and intuitive system was added to choose a message receiver instead of a plaintext box;
- A turn log was added, where the player can see a simplified state of the map in each of the 30 days of the simulation;
- The tutorial was rewritten, making it more clear, simple, harder to break, and much shorter.

5.3. Tablet Support

Although not the main target of the work, one of the proposed objectives was for the interface to work in tablets. The interface was always laid out in a way that would fit into a normal 1280x800 tablet screen.

Supporting tablets also created some additional implementation challenges due to how the interaction is done in a touch screen. This was mostly problematic with the zoom mechanic due to the expected behavior of a scroll wheel not having the correct feeling when on a touch screen. A zoom mechanic using both fingers for pinching was then implemented in a seamless way with the scroll wheel zoom.

6. Testing and Results

This section describes the methodology used to test the implemented interface and the results of these tests. The testing was always done on a PC, in two different sets.

6.1. First Contact

The first set of tests was made with new users, with the objective of testing the attractiveness and intuitiveness of the interface for them. A testing session of about 1 hour per person was done with

22 volunteers, 11 with the version 2 interface, and 11 with the version 3 interface. These were all people that did not know about Almansur but had at least already played strategy games.

The test consisted first of 3 questions to identify their profile and to chose which of the groups the tester would end up into, asking how long they played videogames per week, the age, and which strategy sub-genres he/she plays or had played in the past. After that, the user was asked to progress through the tutorial of the game up to a certain point with the time being measured. Next the user was asked how comfortable were they with continuing to play the game without support. After that they were asked to try to make some tasks in the interface, where the number of clicks and the time taken was measured. Finally, the users answered the System Usability Scale test [18], which consists of a set of 10 abstract questions made to the user regarding the usability of the interface, and then calculates a final value of 1 to 100 depending on the answers.

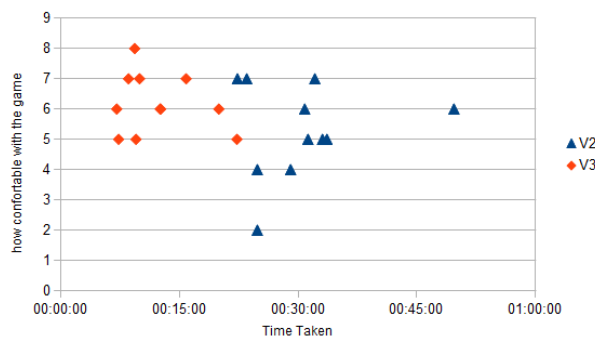


Figure 5: Time taken progressing through the tutorial compared with how comfortable the user were.

The gathered results were compared between the version 2 and version 3 and were overall positive, showing that the version 3 alongside the new tutorial are more adequate to new players. In Fig. 5 a point graph can be seen where the time taken in the tutorial is in the horizontal axis, and the answer to the first question, asking how comfortable players were with continuing exploring the game without guidance, in the vertical axis. The graph shows a average time of 30 minutes and 28 seconds for version 2, and 12 minutes and 16 seconds for version 3. It can also be seen in the figure that there are big variations in the time taken by each player, with version 3 varying between 7 minutes and 4 seconds and 22 minutes and 15 seconds, and version 2 between 22 minutes and 19 seconds and 49 minutes and 41 seconds.

Considering that the average comfortability for version 2 is 5.3, and for version 3 is 6.2, the result is quite positive and the players consider that understood more of the game in near a third of the time.

The task results were however worse, with version 3 showing slightly lower values. There are various possible reasons for this. The group for version 2 may be slightly better, and if we consider that when people received the tasks they had already spent 2.5x more time with the interface in average in version 2, they probably had already spotted how to do some of the tasks they were asked for. The only worrying result here is that almost half of the people were not able to do the task of sending a friendship proposal in both versions. By watching their behavior and asking them, the problem is that most do not consider intuitive that a diplomatic relation change is sent attached to a text message. The interface needs to indicate better this fact.

For the System Usability Scale, version 2 results vary between 22.5 and 70, with an average of 46. Version 3 results vary between 47.5 and 87.5, with a average score of 63, showing some increase even with a shorter tutorial.

6.2. Extended Contact

The second set of tests were made with users that already had played the game for some time. These were a group of 15 veteran players, after a 15 turn game with the interface, and a group of 27 newcomer players (most in common with the 6.1 group), after a 20 turn game with the new interface. The questionnaire for the veteran and newcomer players were however slightly different.

The first part of the questionnaire consisted of a list of 28 interactions with the game that the user had to rate. The newcomers had to rate in a absolute 7 point scale between terrible and great. The veterans had to rate in a relative 7 point scale, compared with their experience with the version 2 interface, between much worse and much better. The veteran players were also asked how do they rate the new interface of the game comparing with the version 2 one in the same 7 point scale. The newcomers were instead submitted to the System Usability Scale test again. Finally both groups were asked how did they rate the new interface taking all their experience into account answering in a 1 to 9 scale.

From the 15 players in the veterans group, 8 answered the questionnaire. All the 27 players of the newcomers group answered the request, how-

ever only 15 responded to the questionnaire, with the remaining 12 stating that they did not play enough of the game to be able to answer the given questions. Feedback from the newcomer users for the interface was overall positive, having a average rating of 5.1 across all 28 interactions. Only 5 of the 28 interactions had the second quartile going below 4, meaning that for the remaining 23 at least 75% of the testers rate the interaction in a positive way. The negative interactions are points that need to be addressed. Feedback for the veteran users, although not as high as expected, was also positive with a average rating of 5 (slightly better). Across the 28 interactions, 7 had a median of better, 13 of slightly better, and 8 of similar. Only 3 of the interactions had the second quartile going below 4, meaning that for the remaining 25 at least 75% of the testers considered the interface better.

The System Usability Score results were negative, with the average score decreasing from the 63 in the tutorial to 53. While this can simply be because of not being exactly the same group of people, some of the users were informally questioned about this after the results were obtained. The answer was that it was not really the interface that was at fault, but the game ending up being different than what they expected when played for an extended period of time, which affects some of the of the questions.

When asked to compare the overall experience of the new interface with the old one, three of the veteran players answered with 5 (slightly better) and five with 6 (better). This is a higher result than the one obtained in the individual questions, showing that, although there still are problems in the interface, the overall experience became better in version 3. The answer to the last question emphasizes this even more. When asked to rate the overall experience with the new interface in a scale of 1 to 9, the majority of the answers were given between 7 and 8, for both the newcomer and veteran players.

7. Conclusions

Almansur is a multiplayer turn based strategy browser game developed throughout the last decade. This worked started with the objective of creating a more efficient and user friendly interface for the game, that would also be able to support tablets.

To reach this, first an analysis of the current game was made, creating a conceptual model, listing requirements and comparing with other existing games. Its usability was also evaluated to

find the version 2 interface faults, observing users playing their turns and doing a Nielsen Heuristic evaluation.

With these results a new interface was then prototyped, implemented, iterated and then tested. A new more robust and user friendly interface was created for Almansur and successfully tested with both new and old players, having had an increase in satisfaction with both groups, and making the game more appealing to attract new users.

The objectives proposed were successfully completed, however the results although positive were not as great as could be expected. The major reason for this is that what this work wanted to achieve made it necessary to make profound changes in the technology used and the implementation structure, having spent a great amount of time reimplementing features that were common from both the version 2 and the version 3 interface. This work however laid out the foundation to continue improve the usability and the overall gameplay of Almansur.

For the usability of the interface there is still some work to be done. The feedback and results of the testing should be taken in account and slight changes implemented. There are still also a few bugs runtime bugs in the turn log. There are also a few screens that were not given the attention that would probably need, due to the enormous side of the interface and the development hurdles discussed on 5.2.

For the overall development of Almansur and outside the scope of this thesis there is still much work that can be done for the future. The account pages still need to be reworked. The created map library can be implemented in multiple parts of the game outside of the in-game interface, like in the land selection screen or to create a map editor. The overall aesthetical design of the interface needs to be polished by a designer to make it more appellative to new users. The in-game documentation still needs improvement and is the Nielsen Heuristics flaw that was less addressed in this work. Finally, the overall gameplay rules and balance need to be polished to make the game concept realize its full potential.

References

- [1] Almansur. <http://www.almansur.net/>. [accessed 14-October-2015].
- [2] Sid Meier's Civilization V . <http://www.civilization5.com/>. [accessed 14-October-2015].

- [3] Paradox Interactive. https://en.wikipedia.org/wiki/Paradox_Interactive. [accessed 14-October-2015].
- [4] Grand Strategy wargame. http://en.wikipedia.org/wiki/Grand_strategy_wargame. [accessed 14-October-2015].
- [5] H. Sharp J. Preece, Y. Rogers. *Interaction Design: Beyond Human-Computer Interaction (3rd Edition)*. John Wiley & Sons, 2011.
- [6] Jakob Nielsen. Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 152–158, New York, NY, USA, 1994. ACM.
- [7] Jakob Nielsen. Usability inspection methods. In *Conference Companion on Human Factors in Computing Systems*, CHI '95, pages 377–378, New York, NY, USA, 1995. ACM.
- [8] Jakob Nielsen. How to Conduct a Heuristic Evaluation. <http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>. [accessed 14-October-2015].
- [9] R. L. Mack J. Nielsen. *Usability Inspection Methods*. John Wiley & Sons, 1994.
- [10] JustInMind Prototyper. <http://www.justinmind.com/>. [accessed 14-October-2015].
- [11] Almansur interface prototype. <http://web.tecnico.ulisboa.pt/ist170623/almansur>. [accessed 14-October-2015].
- [12] Model View Controller Architecture. <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>. [accessed 14-October-2015].
- [13] Ruby On Rails. <http://rubyonrails.org/>. [accessed 14-October-2015].
- [14] Adobe Flash. <http://www.adobe.com/pt/products/flash.html>. [accessed 14-October-2015].
- [15] Pixijs. <http://www.pixijs.com/>. [accessed 14-October-2015].
- [16] WebGL Browser Support. <http://caniuse.com/#feat=webgl>. [accessed 14-October-2015].
- [17] Canvas Browser Support. <http://caniuse.com/#feat=canvas>. [accessed 14-October-2015].
- [18] Measuring Usability With The System Usability Scale. <http://www.measuringu.com/sus.php>. [accessed 14-October-2015].
- [19] G. Abowd R. Beale A. Dix, J. Finlay. *Human-Computer Interaction (3rd Edition)*. Prentice Hall, 2003.