Report Name

Author Names
Team D
Dim3
Fiona Buyers, Christopher James, Ryan Wells
{1003648b, 1003019j, 1002253w }@students.glasgow.ac.uk

ABSTRACT

A multi-player game designed to encourage a balance between personal gain and the well being of a group.

1. AIM OF APPLICATION

1.1 Application Purpose

The application itself is a multi-player game designed to encourage users to work effectively as part of a group. When users register they will be randomly assigned to one of four factions. Whilst playing the user will aim to increase not only their individual score, but the score of their faction. To do this, the user will have to buy buildings to expand their 'city' and increase their population. There will be several different kinds of buildings; farms, barracks, studios and labs; each earning the user points in a different category (food, military, art and science, respectively) and all these buildings will earn money as time progresses. Points earned by the user will be added to the total for the relevant faction and the four faction totals will be displayed, showing which faction is the strongest in each area. Crisis events will occur throughout the game which will cause buildings of a certain kind to become less or more effective, for example famine or war, so in order for the user to have a successful city and gain the maximum number of points, the city they have created should be as well balanced as possible. More accomplished users will be able to unlock different personas when they have enough money, giving the user different multipliers for each of the points categories, affecting the amount of points earned for the faction. The majority of personas will not only feature positive multipliers, but also some that are detrimental to point production. This will allow users to choose what their city will focus on in terms of the types of points they wish to earn, but will also encourage users to continue playing as regular expansions of cities will be required to enable a user to afford a change in persona.

1.2 Functional Requirements

- The user's individual score should be tracked.
- The individual score should affect the appropriate team's overall score.
- Upon registration, the user should be randomly assigned a faction.
- The amount of money a user currently has should be tracked and the user should not be allowed to buy buildings or personas that cost more than they have, and when an item is bought the amount of money possessed should be altered appropriately.
- Users should earn money and points from their current buildings at set time intervals.
- What is the purpose of the application?
- Eg. The application is an academic search engine called AcaSe and is it is based upon the PuppyIR Framework[**?**], which has been used to construct other such services[**?**, **?**]. The main purpose of this web application is to provide a customized interface to services such as Google Scholar and MS Academic Search.
- What are the assumptions about the aims and objectives?
- Describe the design goals and objectives of the application.
- What are the constraints of the project?
- Functionality List: i.e. what is the required and desired functionality?
- Reflective Questions:
- Is the scope of the application appropriate?
- Are the design goals realistic/achievable?
- How complex is the application?
- Is distribution across the web appropriate?

Copyright is held by the author/owner(s).
DIM3 2010, Glasgow, UK
ACM X-XXXXX-XX-X/XX/XX.

2. CLIENT INTERFACE

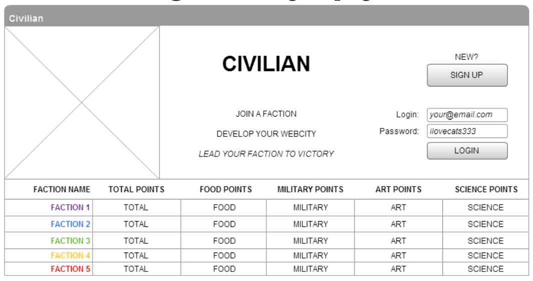
2.1 User Interface

The user interface was primarily designed to provide a fluid and natural experience to the user. When considering the user experience, care was taken to ensure that the interface was attractive to users but still provided enough functionality on each layer such that users could fulfil all the relevant tasks they wished to without having to change page an excessive amount of times.

The interface mostly followed the wireframes originally drawn up, but changes were made following feedback from the presentation earlier in the Semester. The initial design of the application (shown in Figures 1 and 2) had two main pages, a login page, allowing users to login, register and view the current faction scores, and a game page, showing the current users statistics and the buildings the currently owned, along with the current faction scores. Before implementation, this was changed slightly so that the current faction scores are shown in a separate page and both the login and game screens were redesigned slightly, these are shown in figures 204834 - 7174751.

There are three main screens: the index page, the game page and the faction scores page. The index page, provides options to login, register or view the current faction scores. The faction scores page shows the faction that currently has the most money, and also the greatest number of points for each of the categories. This page is accessible from within the game and without logging in, allowing unregistered players the ability to monitor the factions' scores. The main game page shows the current number of buildings each user has, displays their current points and how much money they will earn each turn. Users will also be able to purchase buildings for their city, and personas from this page.

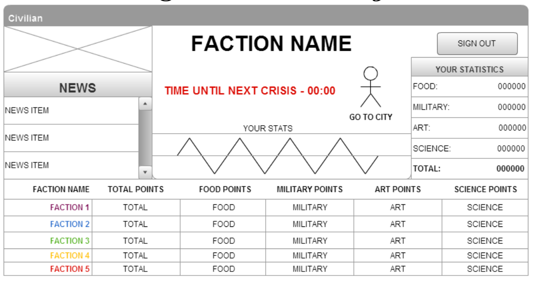
Figure 1: ***login page***



Login: *your@email.com*
Password: *ilovecats333*

FACTION NAME	TOTAL POINTS	FOOD POINTS	MILITARY POINTS	ART POINTS	SCIENCE POINTS
FACTION 1	TOTAL	FOOD	MILITARY	ART	SCIENCE
FACTION 2	TOTAL	FOOD	MILITARY	ART	SCIENCE
FACTION 3	TOTAL	FOOD	MILITARY	ART	SCIENCE
FACTION 4	TOTAL	FOOD	MILITARY	ART	SCIENCE
FACTION 5	TOTAL	FOOD	MILITARY	ART	SCIENCE

Figure 2: ***Game Page***



YOUR STATISTICS	
FOOD:	000000
MILITARY:	000000
ART:	000000
SCIENCE:	000000
TOTAL:	**000000**

FACTION NAME	TOTAL POINTS	FOOD POINTS	MILITARY POINTS	ART POINTS	SCIENCE POINTS
FACTION 1	TOTAL	FOOD	MILITARY	ART	SCIENCE
FACTION 2	TOTAL	FOOD	MILITARY	ART	SCIENCE
FACTION 3	TOTAL	FOOD	MILITARY	ART	SCIENCE
FACTION 4	TOTAL	FOOD	MILITARY	ART	SCIENCE
FACTION 5	TOTAL	FOOD	MILITARY	ART	SCIENCE

2.2 Dynamic Components

Due to the nature of the application, frequent updating of the database is required. The majority of the updating done to the client side uses Ajax GET requests, some at a timed interval, some whenever a user performs a specific event, such as purchasing a building.

- Draw a wireframe of the user interface
- this may require several wireframes depending on the complexity of the application and the interfaces
- Describe the user interface.
- i.e. Label key input and output components: describe them.
- Provide a Walkthrough and explain the user interactions with application.
- i.e. use cases AAAAAAAAARGH
- Describe the interactions associated with the dynamic components on the user interface.
- What calls are required to dynamically update the data on the client side?
- How does the user interface help the user achieve their goal, or complete their task?
- Is the user interface intuitive, appealing, usable, etc?
- What technologies are used on the client side?
- What are the reasons for your choices? i.e. what is the advantages and disadvantages of using this technology?
- What other options are there?

3. APPLICATION ARCHITECTURE

- N-Teir Architecture Diagram - chris has this
- i.e. data flow diagram between the interface/client, middle ware, and backend services/data repos
- Describe the data model i.e. what data needs to be stored or persisted by the application?
- What are the relationships within the data model.
- i.e. use ER diagram and explain. - chris I DONT KNOW, we never had one really
- Describe the backend services used (if any).
- Reflective Questions:
- How have you ensured that there is a separation of concerns?
- What other technology could have been used instead of django?
- What are the advantages of using a Web Application Framework over other technology?
- And, what are the disadvantages?

4. MESSAGE PARSING

- On the architecture diagram, Identify and label the main messages that will be parsed through the application.
- or alternatively (and preferably) include sequence diagrams to denote the sequence of communications parse between clients and servers.
- Describe the messages that are parsed back and forth through the application.
- For the main transactions - describe the payload of the messages
- i.e. What are the contents of the messages? i.e. include sample XML, XHTML, JSON, etc of one or two messages.
- What is the format of the messages?
- Why this format?
- What other formats could be used, what are the advantages and disadvantages of these other formats?

5. IMPLEMENTATION NOTES

- Views - What are the main views that you have implemented and what do they do?
- URL Mapping Schema - what is your URL mapping and schema?
- External Services - what external services does your application include and what handlers did you include?
- Functionality Checklist (which functionality is completed) - everything except crises
- Known Issues (what kind of works, what kind of errors to do you get)
- What technologies have been used and are required for the application. Include a list or table of all the technologies, standards, and protocols that will be required.

6. REFLECTIVE SUMMARY

For the Implementation Report Only:

- What have you learnt through the process of development?
- How did the application of frameworks help or hinder your progress?
- What problems did you encounter? - timing?
- What were your major achievements?

7. SUMMARY AND FUTURE WORK

- Summary of application and its current state.
- Include a list or table of all the technologies, standards, and protocols that will be required.
- What are the limitations?
- Plans for future development - crisis events, more personas and buildings.

8. ACKNOWLEDGEMENTS

Our thanks to Lief and Euan for their comments, suggestions and interest they showed in the project throughout the duration of the course. And our thanks to the peer reviewers for their feedback. Thanks also go to Tom Whitely for his evaluation of the interface and the suggestions he made.