

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. The default persona is the 'civilian' persona, which provides no multipliers for points or money, users who have purchased a persona are able to return to 'civilian' status for free whenever they choose.

1.2 Assumptions

What are the assumptions about the aims and objectives?

1.3 Design Goals and Objectives

Describe the design goals and objectives of the application.

1.4 Constraints

What are the constraints of the project?

1.5 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.
- Users should be able to purchase several different kinds of building and persona if they have enough money.
- Users should be able to return to civilian status for free whenever they want.
- The faction scores page should be accessible both from the login page to users who are not logged in, and from the main game screen.
- The faction scores page should show the percentage of the total number of points each faction has.

1.6 stuff to do

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?

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.

Figure 1: Login Page

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

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

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.

1. Takes the user back to the index page.
2. This will take the user to their game page.
3. Logs the user out.
4. Each bar shows the distribution of points between the factions. This one shows the total points distribution.
5. This shows the percentage of art points the team D have.

Figure 3: Total Faction Points Page

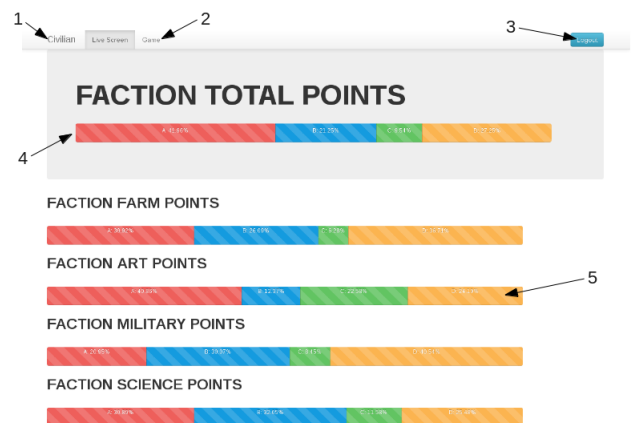
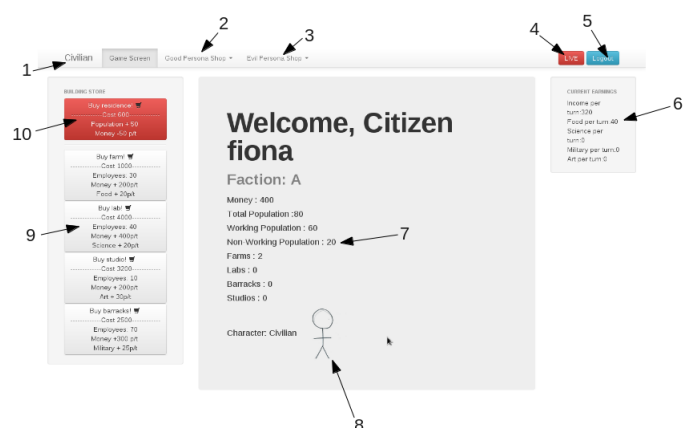


Figure 4: Game Page



1. Takes the user back to the index page.
2. Allows the user to purchase good personas (ie. citizen, gentleman, warrior, scientist).
3. Allows the user to purchase evil personas (ie. pirate, evil doer, emperor, politician).
4. Takes the user to the faction scores screen.
5. Logs the user out.
6. Shows how many points and how much money is earned per turn with the user's current buildings and persona.
7. Shows the number of buildings, the amount of money and the employment information of the city's population.
8. Shows the user's persona.
9. Allows the user to buy new buildings. Also shows how much the building costs, the profits of the building and the points gained from the building gained each turn.
10. Allows the user to buy a residence and shows the population increase from buying the building and the cost of running the residence.

Figure 5: Index Page



1. Allows the user to log in.
2. Lets new users register.
3. Takes the user to the faction scores page.

The final design has 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.

2.2 Dynamic Components

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?

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.

2.3 stuff to do

What technologies are used on the client side? What are the reasons for your choices? i.e. what are the advantages and disadvantages of using this technology? What other options are there?

3. APPLICATION ARCHITECTURE

3.1 N-Tier Architecture Diagram

Figure 6: n tier diagram



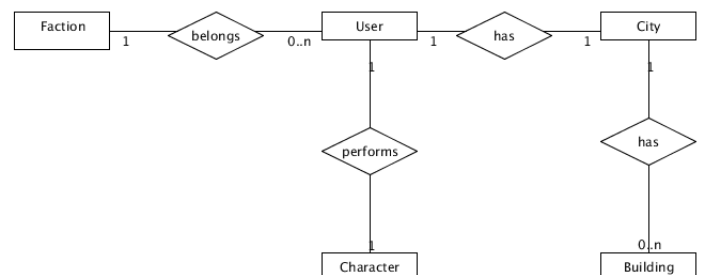
N-Tier Architecture Diagram - chris has this ***need diagrams*** i.e. data flow diagram between the interface/client, middle ware, and backend services/data repos

Describe the data model i.e. what data needs to be stored or persisted by the application? i dont even know what this is

3.2 Entity-Relationship Diagram

What are the relationships within the data model. i.e. use ER diagram and explain. ***need diagrams***

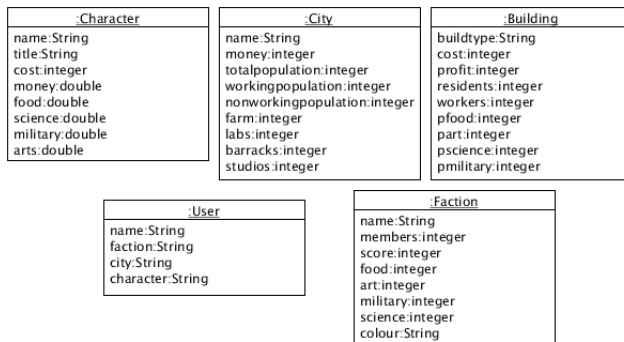
Figure 7: ER diagram



3.3 Backend Services

Describe the backend services used (if any).

Figure 8: *Lovely Tables*



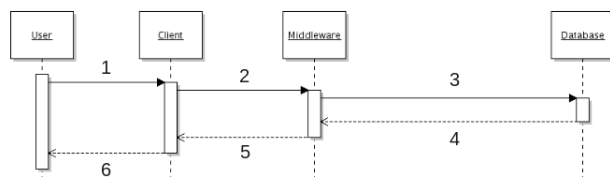
3.4 stuff to do

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

4.1 Sequence Diagrams

Figure 9: *Sequence Diagram*



1. User clicks on desired item.
2. Client sends HTTP GET request to middleware via Ajax
3. Middleware access database using ORM query.
4. Database returns info to middleware.
5. Middleware passed information to client.
6. Client changes info with Ajax.

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. ***need diagrams***

Describe the messages that are parsed back and forth through the application.

4.2 Message Payload

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.

4.3 Message Format

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

5.1 Views

Views - What are the main views that you have implemented and what do they do?

5.2 URL Mapping Schema

URL Mapping Schema - what is your URL mapping and schema?

5.3 External Services

External Services - what external services does your application include and what handlers did you include?

5.4 Functionality Checklist

Functionality Checklist (which functionality is completed) - everything except crises

5.5 Known Issues

Known Issues (what kind of works, what kind of errors to do you get)

there are no known issues, except that everyone will want to play our game now.

5.6 Technologies Used

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

The application has all the initial features implemented, except the crisis events. Users are currently able to purchase buildings of each type initially specified (farm, barracks, studio and labs) and the initial eight personas. The money earned each turned and faction points are updated every minute.

Currently, the main limitation of the game is that after a while, users manage to earn money at a greater rate than they can spend it. This has led to users with millions of credits they are unable to spend and led to some of the more avid users having thousands of buildings and completely dominating the faction leaderboard and no other teams being able to catch up.

In the future, the turn length would be increased in order to prevent the accumulation of enormous amounts of money. The crises event feature would also be implemented and more buildings and personas would be made available. It was also considered that users might eventually be able to split off from a faction to form their own. time until next turn. some kind of graphical thing. increase turn time when logged out

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. We would also like it noted that we appreciate the input of several others who made our application incredibly competitive and verging on violent.