UpStage

Introduction

UpStage is an online platform which allows global collaborators to put on digital performances. UpStage was envisioned in 2004 by Helen Varley Jamieson and Vicki Smith. In 2007, AUT Teams began developing UpStage. Since UpStage is an open-source software, it has not been exclusively worked on by AUT, there has been a number of other contributors such as Martin Eisenbarth who implemented features such as the new video streaming functionality that was merged in semester 2, 2013.

Objectives

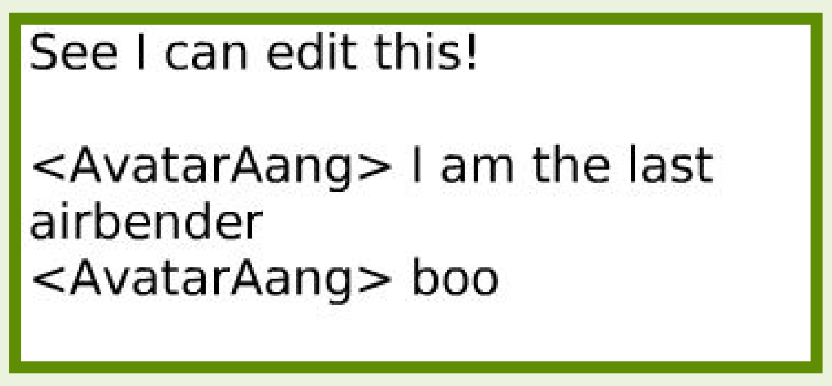
For semester 2, 2013 the main goals of the project were to merge in the video streaming function and the new media management system created by Martin Eisenbarth and resolve high priority tickets to create UpStage v3.0 in time for the 10th birthday festival in January, 2014. Unfortunately, testing in November revealed that the software v3.0 was not ready for release.

The main focus in semester 1, 2014 was stabilizing UpStage by resolving issues that the clients had deemed as vital to be fixed for the release of v3.0. As well as developing on UpStage, the team began an investigation that looked at what is wrong with UpStage, how the outdated technologies it relies upon affects it, the current functionality of UpStage and why a new software is recommended.

Accomplishments

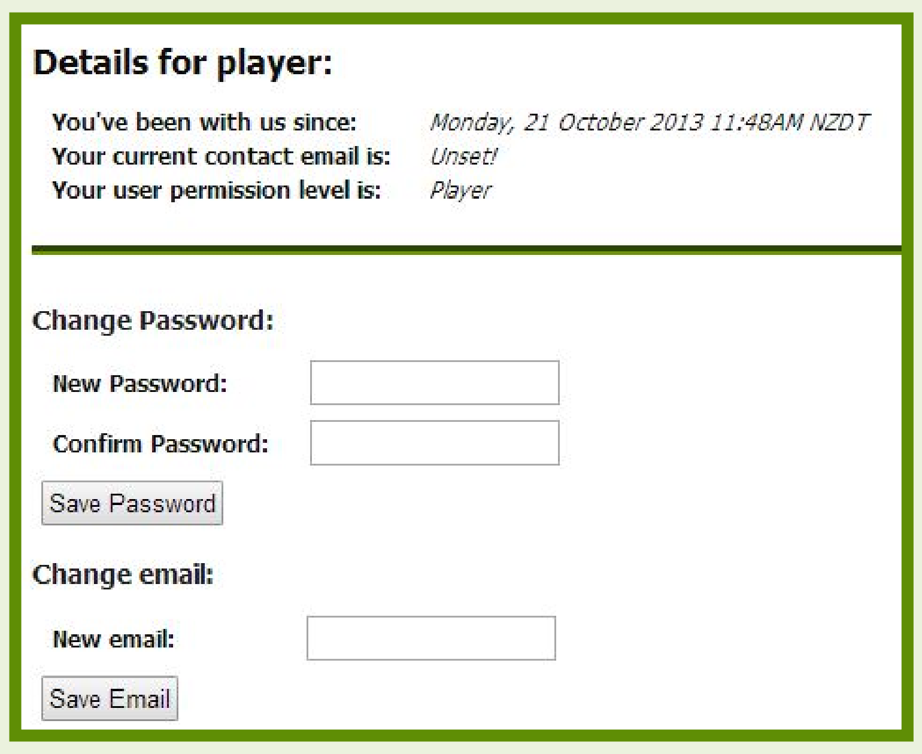
Semester 2, 2013

- Merged in the new video streaming and media management system
- Redefined user permission levels
- Updated User and Server Manual
- Search and filter media implemented
- Mute/unmute streaming avatar locally/globally
- Revert unsaved changes on stage edit page
- Editable chat opening message



See I can edit this!

<AvatarAang> I am the last airbender
<AvatarAang> boo



Details for player:

You've been with us since: *Monday, 21 October 2013 11:48AM NZDT*
Your current contact email is: *Unset!*
Your user permission level is: *Player*

Change Password:

New Password:
Confirm Password:

Change email:

New email:

Semester 1, 2014

- Chat properly loaded
- Logging out functionality
- Stage not cleared when assigning media
- Player can edit own profile
- Minor cosmetic changes
- Created motivational document for the move to a new product

Practices

During our project we used a form of agile development: SCRUM.

Our sprints we scheduled were 2 weeks long, however we found that no work was being done in the first week. To combat this issue, we found that using a tighter scheduling regime worked better. We would schedule each week of our sprints, assigning tasks and estimating story points in team meetings. After each sprint, the team would acceptance test the software. The lack of automated tests led to this process taking 5 people an entire day, and often it revealed issues previously unnoticed.

In semester 1, 2014 this process fell apart. This happened because we did not have the level of skills required to correctly estimate the amount of work that would need to be done on a task at the start of the sprint.

At the start of each semester with the addition of new team members, pair programming was used in order to help the new members help understand the code.

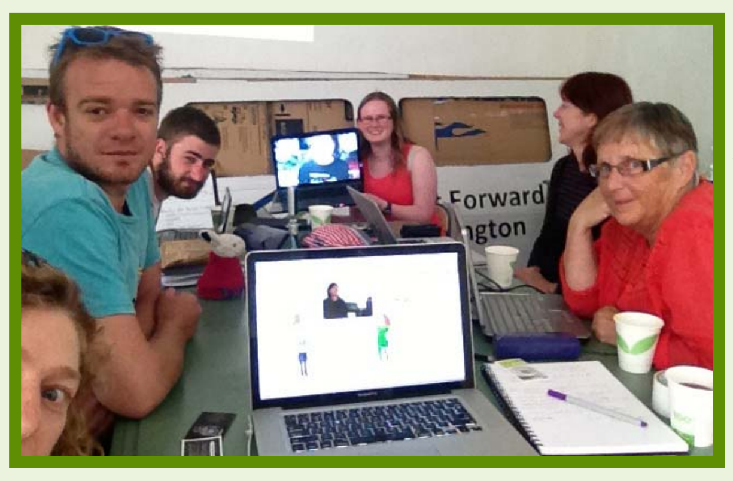
Learning

- Working in a semi-professional environment
- Basic understanding of Python, ActionScript and JavaScript
- Complexities of working with an existing software base
  - ⇒ Reading code others have written
  - ⇒ Remembering to comment everything you change
  - ⇒ Refactoring while developing
- Version control using GitHub
- Developing without an IDE

Challenges & Solutions

• Working with pre-existing complex software	Pair-programming
• Lack of automated testing	Batch acceptance testing as a team
• Code doesn't adhere to coding standards	Refactoring code while developing
• Scheduling sprints around assignments and other commitments	Created a calendar and scheduled sprints accordingly
• Long-distance clients	Communicated via email, irc chat, GitHub issues and meetings on UpStage
• Motivation ⇒ With knowledge that new software was being developed to replace UpStage, working on it was disheartening as we knew that our work was going to be obsolete	Worked on researching the new software alongside developing UpStage
• Reimaging the development server and reinstalling UpStage ⇒ The installation material wasn't as clear as it should be and some of the packages needed are outdated and have been renamed	Used online resources to find what the outdated package names had been changed to and updated the installation manual for future teams

Future of UpStage



UpStage celebrated its 10th birthday in January, 2014 and we were able to go down and join in the celebrations. As a part of the celebrations, there was a meeting held to discuss the future of UpStage and where to go after UpStage v3.0 is officially released. It was found that UpStage no longer meets Helen and Vicki's original expectations to do with accessibility, the code is no longer maintainable and the technologies that it uses are out of date.

To remedy this, Helen and Vicki came to a decision that after UpStage v3.0 is stable, the AUT Team will move to investigating a new software. The motivational document that explains a need for the change was created in semester 1, 2014. From here, teams will be able to investigate which technologies, architectures and languages would be best for the new software before starting to develop it.

Semester 1, 2014

Nikos Phillips (1117467)

Vanessa Henderson (1127036)

Clients: Helen Varley Jamieson
Vicki Smith

Supervisor: Anne Philpott

AUT
UNIVERSITY