**Motivation for re-newed UpStage program**

**Last modified: 31/05/2014**

**Version Number: 3**

**Contents**

**1. Current State of UpStage**

**2. Goals for New Project**

**3. UpStage’s Functionality**

**4. Research Areas for New Project**

**5. New Project Metrics**

**6. Other Consideration**s

**1. Current State of UpStage**

**User base**

UpStage is currently being used by artists, schools, universities and community groups from different parts of the world. The two main user bases are artists and students. There are some differences between the way artistic performers and a performer from an educational institute use UpStage. The artists mainly use pre-loaded media as opposed to teachers who more often use real time interaction such as the chat and drawing functions and streaming video.

**State of Code**

It has reached the point where the code has become extremely difficult to work with and maintain as well as test. There is a lack of inbuilt automated testing framework to run test suites. Due to a lack of having an automated test suite, it takes extremely long amounts of time to test, and bugs are not easily found. Some of the original code does not follow naming conventions and many parts of the UpStage code lack development documentation. As Martin mentioned on the UpStage developer log, developing on Upstage is similar to reverse engineering where in terms of iterative process, the development can be broken down to the following:

1. Code analysis and debugging
2. Modifying the code base
3. Testing the modifications

The state of the legacy code makes it very hard to go back and make enhancements or fix bugs.

**Accessibility and Portability**

Another factor driving change is accessibility/portability. The definition of portability has changed since the first version of UpStage was released. By today’s standards to be accessible and portable the software needs to be available for mobile devices such as Android and iOS as well as accessible from a desktop. There is a serious need for a complete redesign of the software to keep it up to date and relevant for the modern user, and to enable it to be vastly more extensible and maintainable.

**Resource**

There are no funds for creating the new software as far as we know. All human resource are supplied by open source community, subscriber of UpStage IRC channel and AUT students/staff. Martin Eisenbarth, a developer who previously helped AUT students with development work, will likely not be able to aid the project in the near future.

**Technologies used (limitations and/or known issues caused by it)**

Flash

* Flash player must be installed on the user end.
* Less maturity compare to other video formats
* Restricts the language to ActionScript
* The usage on websites are declining.
* Current version of flash does not work on mobile. ( maybe possible to run with newer version with flashLite installed on the device)

Client-Server Model

* Code executed at client end, rather than server. Prevents UpStage Version 3 running on mobile devices.
* Appears to be thin client - fat server, but perhaps not as thin at the client end as it should be. (Code is executed on the client machine i.e. in the browser).

AJAX

XML

**Language used**

Python with Twisted (is a package/library which is outdated) - Client

ActionScript 2 - Server

Javascript - client

HTML

CSS

**Current Limitations of UpStage**

* Non-repeatable bugs
* Extremely difficult to extend
* Low modularity
* Tight coupling?
* Extremely difficult to refactor
* Non-strict naming conventions
* Unhelpful documentation and comments in code (has gotten better with recent with AUT teams)
* Obsolete technology (i.e. ActionScript, Twisted, Flash etc..)
* Not available on mobile devices
* No automated testing facility

**2. Goals for New Project**

The goal for the new project is to provide software that offers the same features and functionality as the current version, but with the underlying code conforming to standard naming and layout conventions, being modular, loosely coupled, well documented (useful comments among code), and well tested (having an automated testing facility). This will all make it much easier to add and extend functionality to the program, identify and remove bugs

Listed:

* Reconstruct the flexible nature of the stage to incorporate different users (artists, students etc..).
* Retain all current functionality of UpStage.
* Be easily maintainable.
* Strictly follow coding standards (documenting code and naming conventions).
* Be highly extensible.
* Follow a carefully selected design pattern and architecture.
* Be developed to a delivery model that could make the application available for portable devices.
* Store information in a database (rather than XML files).
* Use newer more relevant technology such as HTML5 (with no flash) and CherryPy rather than Twisted, or an object oriented language.
* Utilise an automated testing suite.

**3. UpStage’s Functionality**

**Audience**

* A link to find user’s local time in the homepage
* Can enter a stage
* Can send text messages in the stage
* Can register to be a Player
* Sign in with the registered account

**Player**

* Has the same functionalities as audience
* Can log out
* Will have tool bar in the stage if the player is assigned to this stage
* Can put avatars in the stage
* Can move the avatars in the stage
* Can stop the avatars in the stage
* Can select a prop at the bottom right side of the window
* Can select a background at the bottom left side of the window
* Can clear the avatars in the stage
* Can play audio in the stage
* Can turn volume up/down for audio in the stage
* Can stop a single audio
* Can stop all audio
* Can rename their current avatar
* Can draw line with chosen color in the stage
* Can select the type of line to draw
* Can change the thickness of line to draw
* Can clear the line

**Creator**

* Can log out
* Create a new Stage
  + Textboxes for entering the stage’s full name and short name for url
* Edit a stage
  + Textboxes for modifying the stage’s full name and short name for url
  + Textbox for modifying the splash message
  + Can choose if the debug messages are printed on the backdrop
  + Can change the props and backdrops’ color
  + Can change the chat window’s color
* Upload media so that it can be used in stage
  + Choose the type of media for uploading: avatar, prop, backdrop, audio or video-avatar
  + Textboxes for entering media’s name and tags
  + For avatar, a voice and be chosen
  + The chosen voice can be tested
  + The number of frames can be selected
  + A local file can be chosen to attach
  + Choose the stages to assign the media to.
  + For prop, the number of frames can be selected
  + Choose the stages to assign the media to
  + For backdrop, the number of frames can be selected
  + Choose the stages to assign the media to
  + For audio, choose the type of this audio: sound effect or music
  + A local file can be chosen to attach
  + Choose the stages to assign the media to
  + For video-avatar, choose from existing streams in the /media/video directory
  + Choose the stages to assign the media to
* View an uploaded media
  + Display specific media by choosing filters
  + Add more filter to display an eligible media
  + Remove the filter
  + Reset button for reset the searching filter and result
  + Search by media’s tags
  + Edit an uploaded media by modify name, voice and the stages to assign
  + A button to display the detail of the media
  + A button to save changes
  + A button to delete the media
  + Select delete the media even if in use
* Edit players:
  + Textboxes for modifying the password, confirm password and email
  + A button to save the new password
  + A button to save the new email
* Enter a stage
  + Can send text message in the stage
  + Be assigned to a stage :
  + The message sent can be read out as a chosen voice
  + Has the tool bar(same tool car as player’s) in the stage if the creator is assigned to this stage
* Can create a new player account
  + Textboxes for entering the username, password, confirm password and email address
  + Select the type of the new player account: player, maker, unlimited maker, admin or creator
  + A button to save
* Can edit existing player details
  + Can display the detail of all existing player accounts
  + Can select one of them to change email, password
  + A button to update the changes
  + A button to delete the player’s account
* A link to edit page mode
  + Can edit the homepage
  + Can edit the workshop
  + Can edit the player page
  + Can edit the stages page
  + Can edit the sign up page
  + A button to submit the changes
  + A button to reset the page to be default

**Maker**

* Has the same functionality as creator except creating new player and editing existing player details
* Has upload limit

**Unlimited Maker**

* Has the same functionality as creator except creating new player and editing existing player details
* Has no file size upload limit

**Admin**

* Has the same functionality as creator
* Has upload limited of a file size

**4. Research Areas for New Project**

* Architecture
* Programming languages
* Design patterns
* Open source community
* IDEs
* Automated testing suites
* Versioning control software
* Research methodology

**5. New Project Metrics**

* Cost
* Maintainability (ease of testing)
* Extensibility (future proofing)
* Compatibility with existing components
* Compatibility with other new components
* Ability to use with existing UpStage source code
* Feasibility

**6 Other Considerations**

**Ongoing projects**

As upstage is an ongoing project, there are new features added constantly. The development team should consider if these features will be easy to implement with new architecture, languages or design pattern.

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Modified By |
| 20/05/2014 | 1 | Created main structure of document. Added known status of Upstage from proposal document (2014\_Sem1) | Takuma |
| 22/05/2014 | 2 | Added headings for important sections and bullet points which must be elaborated | Takuma |
| 23/05/2014 | 2.1 | Added more heading such as technology and language used. Bullet points should be elaborated | Takuma |
| 30/5/2014 | 2.2 | Added existing functionality/mechanism | Xiangyu |
| 31/05/14 | 3 | Combined with James’ work on the rationale and motivation, and proof read. | James |