YouDanMu

Sprint 2

Incremental Testing and

Regression Testing Log

Team Member: Naiwei Zheng, Yucong Ma,
Yibo Gou, Ge Yan,
Jiaqi Zhu

1 Classification Of Components

1.1 Define Components

a. Incremental Testing Form

We use Top-down form of incremental testing since our modules are highly dependent to each other and our plan of implementation of project are top-down, meaning many of the sub-modules are not finished in Sprint 1, it is easier to design and implement for us to perform Top-down testing rather than bottom-up testing.

b. Modules

i. YDM

Main entrance of the program, the realization and initialization for all other modules, communicate with Browser Extension to listen events and assign action to each modules

- Input: Events from Video Providers
- Output: Events to signal other modules
- Parent Dependency: None
- Child Dependency: Danmaku Providers, Danmaku Timeline, Renderer, Video Providers, Danmaku, Browser Extension

ii. Danmaku Timeline

Store the timeline of Danmaku and host function that perform play, pause, speed change of the timeline. Communicate with Renderer as the data source of contents to be rendered

- Input: List of Danmaku fetched from Danmaku Providers
- Output: Animation frame update events
- Parent Dependency: YDM, Danmaku Providers
- Child Dependency: Danmaku

iii. Danmaku Providers

An interface to be implemented to be classes for loading and parsing the sources of danmaku.

- Input: Video identifier
- Output: List of Danmaku from the specific Danmaku source
- Parent Dependency: YDM
- Child Dependency: Danmaku

iv. Video Providers

An interface to be implemented to be classes that store the information of current video player states and events

- Input: Third party events (e.g. YouTube internal events)
- Output: Common YDM events
- Parent Dependency: YDM

- Child Dependency: Third party interface (e.g. YouTube API)

v. Browser Extension

The outward extension of the plugin to communicate with browser, hook up events listeners, inject javascript code and interact with DOM elements.

- Input: Events from browser
- Output: Common YDN events
- Parent Dependency: YDM
- Child Dependency: Browser API

vi. Danmaku

The class to be instantiated to store a single danmaku content, styles and state

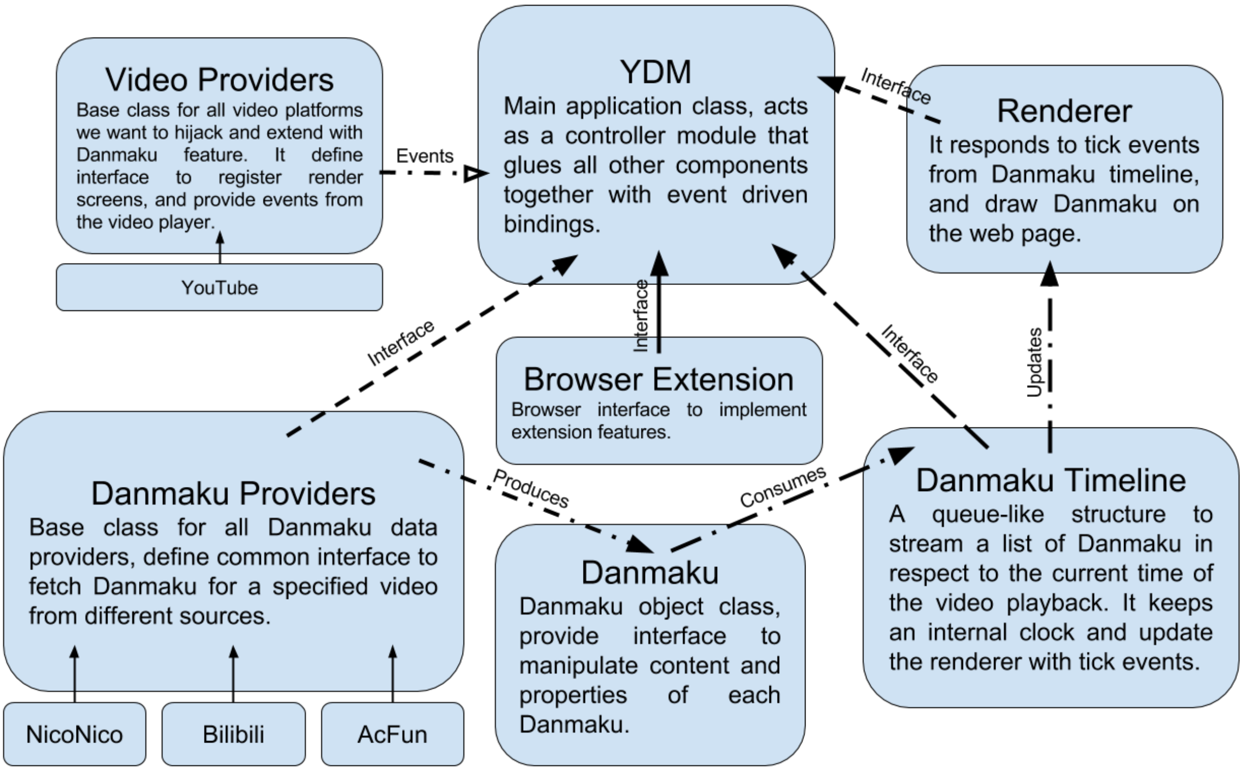
- Input: Danmaku content and properties
- Output: Danmaku object encapsulate the DOM element
- Parent Dependency: YDM
- Child Dependency: None

vii. Renderer

Convert our data to DOM elements that are rendered by the browser. React to resize and state change of the player.

- Input: Danmaku Timeline tick events
- Output: Draw Danmaku on the web page
- Parent Dependency: YDM, Danmaku Timeline
- Child Dependency: Danmaku

c. Diagram


Video Providers
Base class for all video platforms we want to hijack and extend with Danmaku feature. It define interface to register render screens, and provide events from the video player.
YouTube
Events
YDM
Main application class, acts as a controller module that glues all other components together with event driven bindings.
Interface
Renderer
It responds to tick events from Danmaku timeline, and draw Danmaku on the web page.
Interface
Browser Extension
Browser interface to implement extension features.
Interface
Interface
Updates
Danmaku Providers
Base class for all Danmaku data providers, define common interface to fetch Danmaku for a specified video from different sources.
NicoNico
Bilibili
AcFun
Produces
Danmaku
Danmaku object class, provide interface to manipulate content and properties of each Danmaku.
Consumes
Danmaku Timeline
A queue-like structure to stream a list of Danmaku in respect to the current time of the video playback. It keeps an internal clock and update the renderer with tick events.

1.2 Which form of incremental testing did you follow

For our incremental testing, we used top-down testing with the help of stubs. This is because most of the leaf-node modules in our project are depending on the environments, and should integrate with third party modules. Such as the Danmaku Providers classes, they invokes API from different websites but eventually returns a common data type to our other modules. The YouTube API hijack the YouTube API and listen to web page events but eventually provides a set of common events to other modules. With top-down testing, we can replace these leaf-node environment depending modules with dummy stubs that only responds to the common interface, without actually invoking their party interfaces. This way, we can free our testing environment from the actual production environment, for instance, we can test most of the modules merely under the command line, event without running a browser.

2 Incremental and Regression Testing

2.1 Automation

We do use automation to assist our incremental and regression testing. Since our source code is written in TypeScript, we want to benefits from the type checking and IDE static code analysis by writing our test cases also in TypeScript. Thus we need to compile our test scripts along with our source code. To do this we used Gulp.js, a stream-interfaced asynchronous JavaScript task runner; node-typescript, a package that stream the source files to the TypeScript compiler, and several other toolings.

We choose Jasmine.js as our test framework, with assistance with Chai.js as an environment independent assertion library, Mocha.js as the task runner in browser environment, and coverall.js as the code coverage reporter.

The testing strategy is to write a set of test cases for every single module. We refactored our modules into single files, and we can dynamically substitute other modules with dummy stubs with a static testing interface. This way we can just import a module, replaces other modules with stubs, then write assertion with set of test cases to conduct testing on a single module.

For regression testing, we wrote a Gulp job that watch on source code changes, with each change, it automatically rerun the compilation and testing process, and produce code coverage report. Thus every fix will be tested immediately and the automation process would efficiently assist the development iteration.

2.2 Defect log

* Severity: 1 – Workaround 2 – Important 3 – Critical

Module	danmakuProvider

Incremental Test Log

Defect No.	Description	Severity	How to correct
1	Danmaku parser has a specific routine that cannot be escaped once the file download from the server is incomplete and cause critical crash to the page when loading new danmaku	1	Added exception for parser to throw and the exception is being catched by the danmakuProvider module and prompt to error message to the user interaction interface
2	When trying to switch the current Danmaku provider, if the current parsing process is incomplete or stuck, user cannot switch the Danmaku Provider before the process is complete	1	rewrite the interface of the danmaku provider. The function now accept a callback the callback is called once the process is complete. The caller of the function can drop the existing call and start a new once without stuck the routine.
3	It is assumed that the text download from the danmaku source are encoded in Unicode. If it is not, the file will not be decoded correctly	2	The parser now accept an optional argument to select the parsing encoding.

Regression Test Log

Defect No.	Description	Severity	How to correct
1	utf8, utf16 and unicode are supported by the parsing natively only. Some Japanese source using	1	Use an external text-encoding module to decode these unsupported encoding natively

	euc-jp iso-2022-jp are not supported by the parser.		
2	encoding by some rare encoding such as big5, euc-jp iso-2022-jp have extra header or footer that are out of concern by the parser	2	Discard these footer and header in the parsing process.

Module	Danmaku

Incremental Test Log

Defect No.	Description	Severity	How to correct
1	Font size can not be set successfully, because we use SVG TEXT element to render which not support to change via size attribute.	1	Set the fontSize via stylesheet.
2	Special character(<,>.etc.) in text will be recognized as HTML syntax. Because we use innerHtml which will not process escaping.	3	Use textContent which automatically escape the inner text.
3	Cannot set color succssefully, because SVG TEXT element dose not support changing color via stylesheet color attribute.	1	Using Fill attribute instead.
4	Cannot set X value of TEXT element will cause mismatch. Because the X value should be a string instead of a number.	2	Using to string to convert it to String. Using parseFloat to convert float.
5	TEXT element style cannot be rendered correctly. Because when we create TEXT element ,we do not specific SVG namespace.	2	Using createElementNS when create TEXT element, and specific namespace.
6	Cannot calculate width and height of TEXT element. Because only the element which has been added into web page will have width and height.	2	Before acquire width and height, add the TEXT element into a parent execeed screen boarder.

Regression Test Log

Defect No.	Description	Severity	How to correct

1	If the TEXT element has already been rendered into Screen, when we try to get width and height, this element will disappear. Because TEXT element can only be linked with on parent, when we link it with another parent, it will be removed from current location.	2	Using cloneNode to make a copy target element and add it into another parent.
2	When we use cloneNode, the width and height we acquire is always zero. Because, cloneNode have a deep option, when we use default call will only copy the shallow attribute instead of innerText, so the element copy has not Text	1	Using true a deep option's parameter, which will make it recursively copy and will contain Text.
3	When we encountered Danmaku with long context(grows larger than a point) the width will not grows with it , because we use offsetWidth attribute which will only calculate the element in the border of screen.	2	Using clientWidth instead of offsetWidth.

Module	Danmaku Timeline

Incremental Test Log

Defect No.	Description	Severity	How to correct
1	Interval tree iterator will return a data without a type once the iteration is finished.	3	Define a data be null once the iteration is finished.
2	If the interval tree is empty, the remove function will try to search node in the tree without a root node which is undefined before the first node been inserted and crash.	3	Check if the root node is undefined if so return false directly.
3	When try to remove a node from a tree which only have one node, will cannot access the parent(null) and crash.	2	Create a special case for the situation which only one node existed in the tree.
4	If there is no node in the interval tree, the iterator returned by the interval tree will start by one element that is undefined.	2	Function that return iterator will not return iterator if there is no node in the interval tree.
5	The data attribute in interval tree, we will check the data when we try to remove node. We can successfully insert node if the data is undefined. But we cannot remove it correctly.	3	Check the data before any insertion, undefined is not allowed.

Regression Test Log

Defect No.	Description	Severity	How to correct
1	Special move case will not update, height of the tree.	3	update the height of tree to 0
2	During insertion, if the high value is small than low value, insertion will be successfully processed which will cause a negative max value.	3	Make this kind of insertion invalid.
3	After removing the last node in the tree, the rebalance process will try to balance an undefined.	3	Check if the root node is already undefined before rebalance.
4	If the last node has been removed from the tree, search function will return an array with an undefined element instead of empty array.	3	Check before return, if the tree node is undefined, return an empty array.
5	If there is no node in the tree that satisfied condition of the node to be removed, the remove function will return null instead of undefined.	3	If no node could be removed, return undefined.

Module	Renderer

Incremental Test Log

Defect No.	Description	Severity	How to correct
1	Segments structure tries to query an available segment for a given length that overlaps the least number of other segments. But the number of overlaps is wrong because we used a for...in loop over a Set from ES6 that the TypeScript compiler cannot translate into ES5.	3	We use the older version syntax of ES5 to loop the Set.
2	When we try to combine two intervals to calculate the overlaps, the result would be wrong because we do not update the previous and current pointers.	3	Update the pointers.
3	When we use requestAnimationFrame to get the currentTime, we assume that the time slice from the previous frame, but actually it's a timestamp from performs.now.	3	We need to cache the previous time stamp and also the start time of the video, then calculate the time slice we need.

Regression Test Log

Defect No.	Description	Severity	How to correct
1	The calculation of time slice is not in secord but ms.	2	Divide the time slice by 1000 to get secords.

Module	Browser Extension

Incremental Test Log

Defect No.	Description	Severity	How to correct
1	When trying to recieve from the browser, it will wait forever.	3	Add a timeout to stop waiting.
2	If the plug-in is unloaded, during the message transmission process, the service will lost contest and cause crash.	3	Check if the contest is loaded.

Regression Test Log

Defect No.	Description	Severity	How to correct
1	When the plug-in is unload, it will not return an undefined.	2	change the function to return an error instead of undefined.
2	When the timeout stop the waiting process, it will return undefined.	2	change the function to return an timeout error instead of undefined.

Module	Setting view

Incremental Test Log

Defect No.	Description	Severity	How to correct
1	Display content will be editable which allow user to change context in it and crash.	3	Set display content to not editable.
2	User can input negative font size,0, or float number and cause crash.	3	Set the invalid value which will nt be accepted.
3	Some Radiobuttion an be unselect, and the type in setting will be undefined.	3	Prompt user that those Radiobutton is required.

Regression Test Log

Defect No.	Description	Severity	How to correct
1	Even the Message has been prompted. Seting process is already finished.	2	Set a default type for those RadioButton.
2	When encounter invalide value, the value will be set to default value instead of previous saved value.	2	Set those value to previous saved value.

Module	Post Danmaku View

Incremental Test Log

Defect No.	Description	Severity	How to correct
1	If the user try to post a danmaku with a length that exceed max_safe_int, When we try to get danmaku length will cause a crash.	3	Limited the max_length of danamku to max_safe_int.
2	User can select multiple color for Danmaku.	2	Deselect the previous selection if user select new color.
3	If the user try to post a danmaku with a font size that exceed max_safe_int, When we try to get danmaku length will cause a crash.	3	Limited the max_font size of danamku to max_safe_int.

Regression Test Log

Defect No.	Description	Severity	How to correct
1	Max_safe_int is an extremely large value wich canot not be posted.	3	Limited the max_length of danamku to reasonable value.