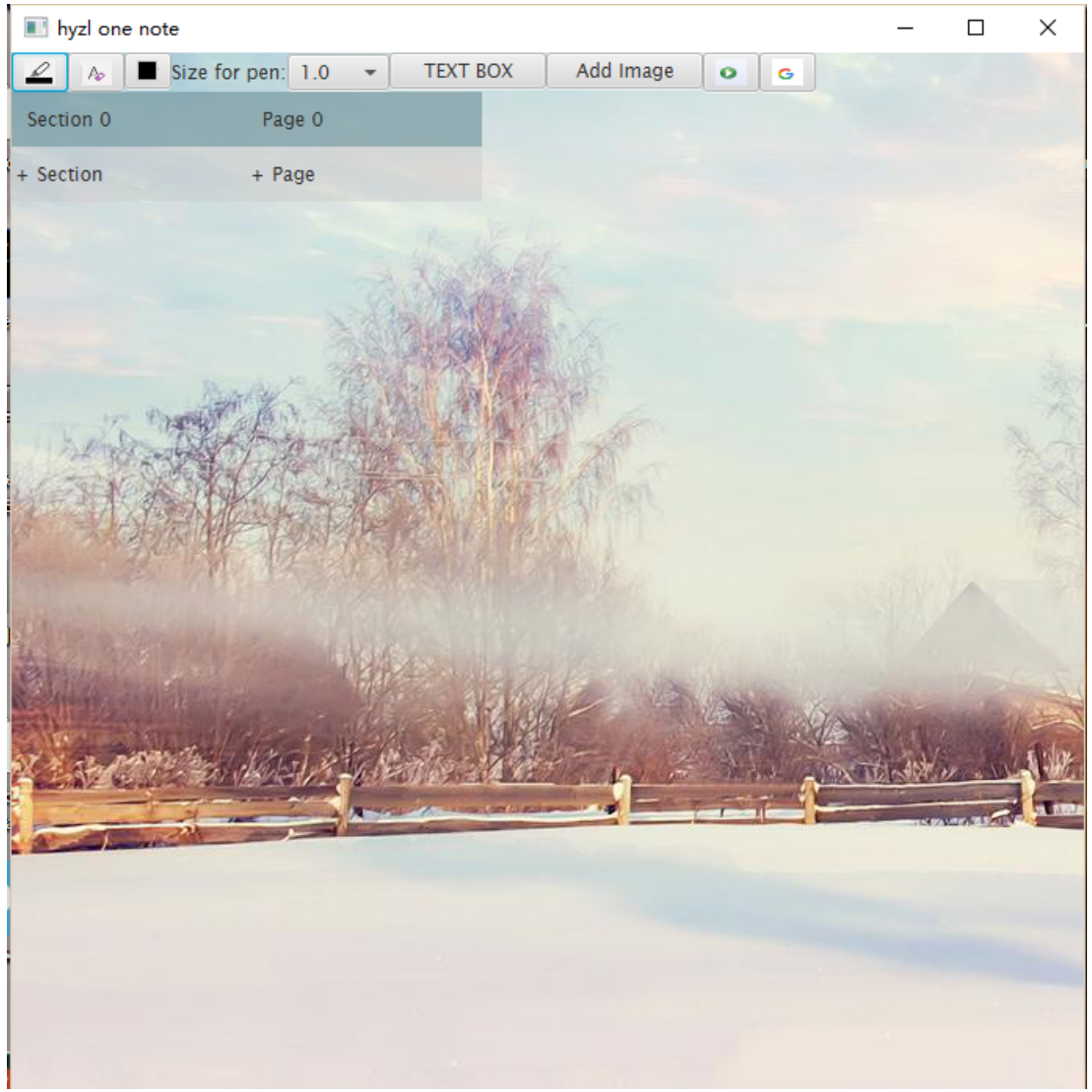
FunNote

BY JIACHENG YANG, GUOXIN HUANG, WENKAI ZHENG, SHU LIN



Topic

Model/View/Controller

Pen

Eraser

Image

Video

Textbox

Webpage

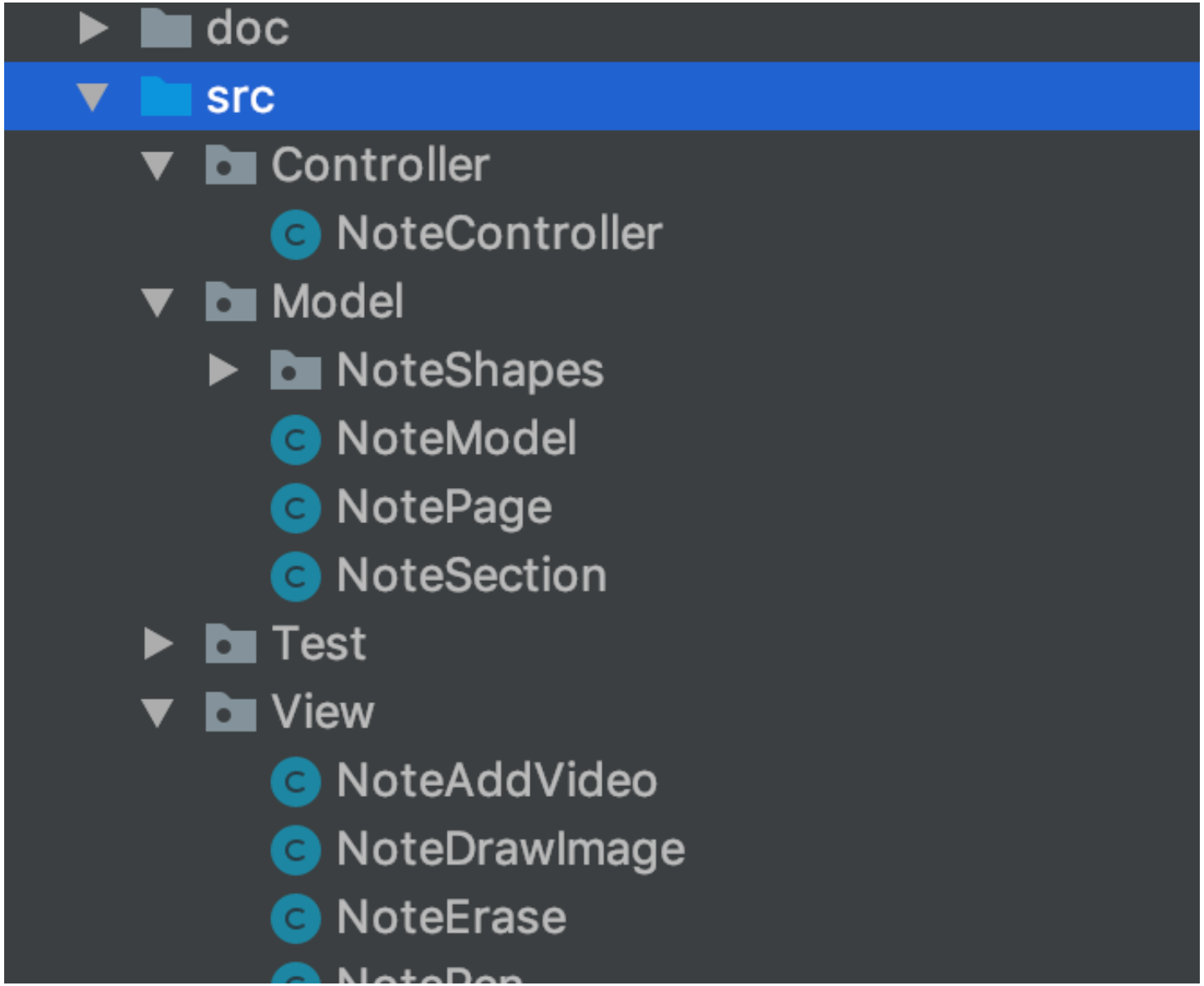
Save/Load

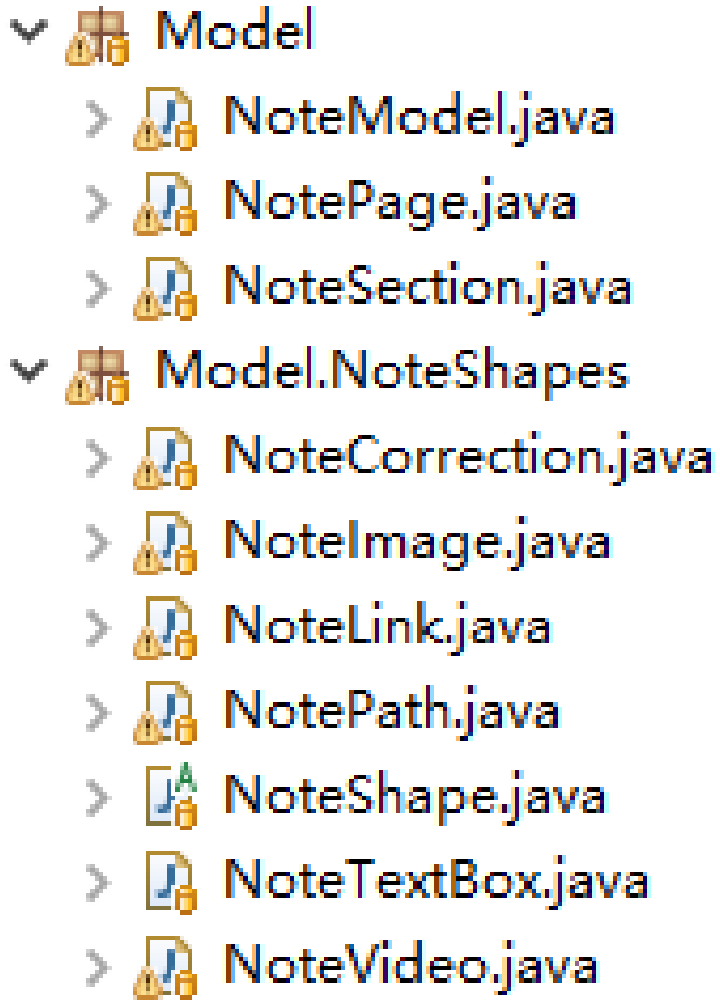
Section/Page

MVC

The MVC is the capstone of our program.

Each of us could modify and adding features to the program without affecting other

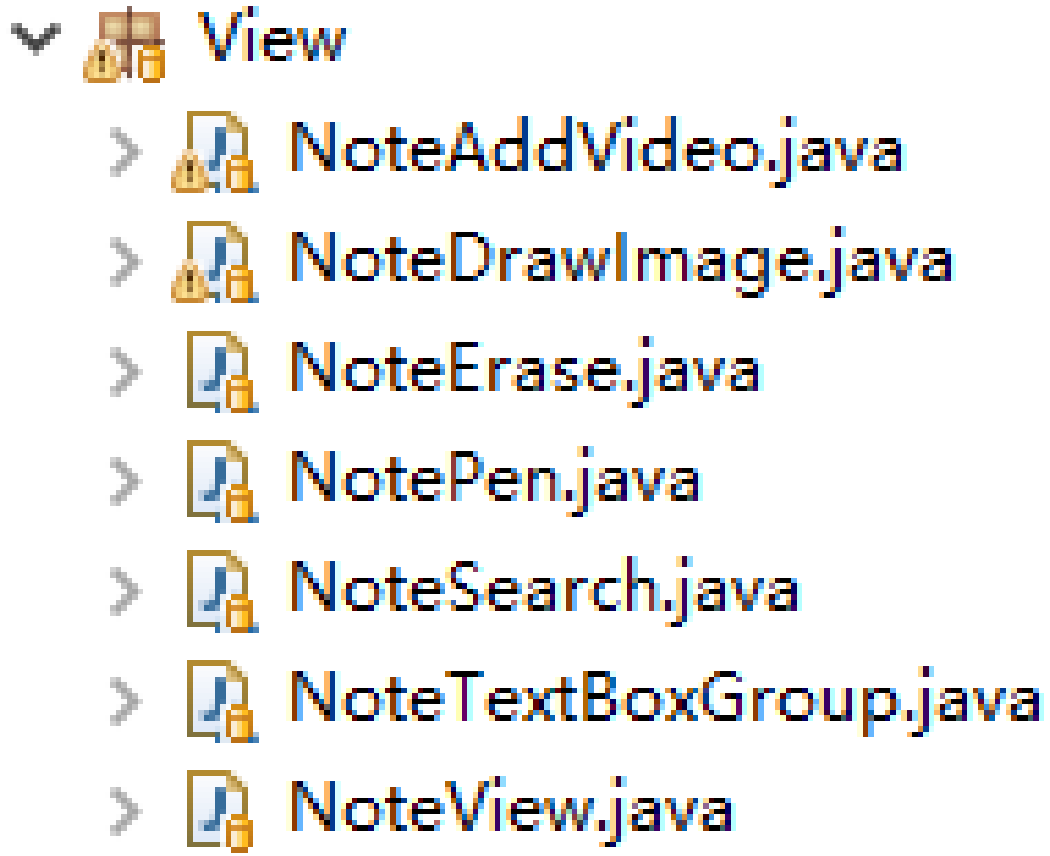



Model
NoteModel.java
NotePage.java
NoteSection.java
Model.NoteShapes
NoteCorrection.java
NoteImage.java
NoteLink.java
NotePath.java
NoteShape.java
NoteTextBox.java
NoteVideo.java

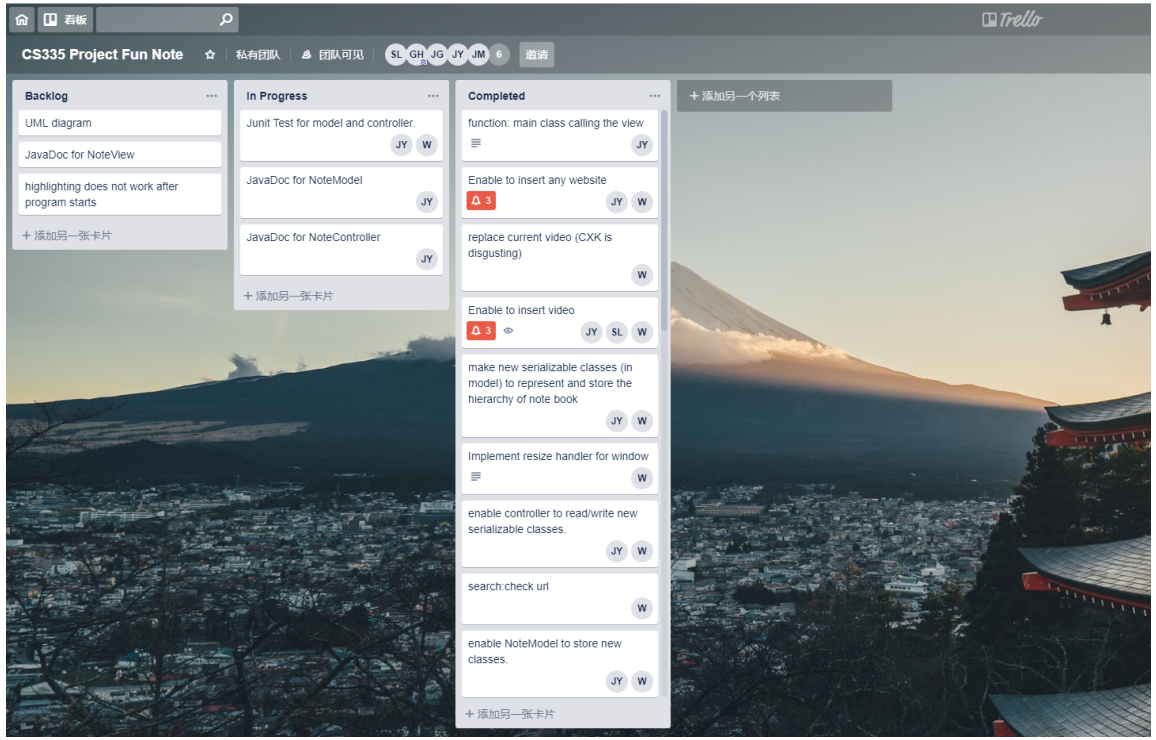
Model

```java
public class NoteController {
	private NoteModel model;
	private int progress = 0;

	public NoteController(NoteModel model) {
		this.model = model;
	}

	public void newSection() {
		int id = model.getNextSectionIdWithIncrement();
		model.getSectionMap().put(id, new NoteSection(id, "Section " + id));
	}

	public void newPage(int sectionId) {
		NoteSection section = model.getSectionMap().get(sectionId);
		int id = section.getNextPageIDWithIncrement();
		section.getPageMap().put(id, new NotePage(id, "Page " + id));
	}

	public void setSectionName(int id, String name) {
		model.getSection(id).setName(name);
	}

	public void setPageName(int sectionId, int pageId, String name) {
		model.getSection(sectionId).getPage(pageId).setName(name);
	}

	public void removeSection(int sectionId) {
		model.removeSection(sectionId);
	}

	public void removePageSecion(int sectionId, int pageId) {
		model.getSection(sectionId).removePage(pageId);
	}

	public int getNextSectionIdWithIncrement() {
		return model.getNextSectionIdWithIncrement();
	}

	public int getNextPageIdWithIncrement(int sectionId) {
		return model.getSection(sectionId).getNextPageIDWithIncrement();
	}

	public String getColor() {
		return model.currentColor;
	}

	public void setColor(String color) {
		model.currentColor = color;
	}

	public void addShape(NoteShape noteShape) {
		model.addShapeToCurrentPage(noteShape);
	}

	public void setCurrentFunctionality(NoteModel.Functionality functionality) {
		model.currentFunctionality = functionality;
	}

	public boolean isCurrentFunctionality(NoteModel.Functionality functionality) {
		return model.currentFunctionality == functionality;
	}

	public NoteModel.Functionality getCurrentFunctionality() {
		return model.currentFunctionality;
	}

	public double getPenSize() {
		return model.getPenSize();
	}

	public void setPenSize(double s) {
		model.setPenSize(s);
	}

	public void saveModelTo(File file) throws IOException {
		try (ObjectOutputStream outputStream = new ObjectOutputStream(new FileOutputS
			outputStream.writeObject(model.getSectionMap());
			System.out.println("Your note is saved...");
		}
	}

	public void readModelFrom(File file) throws IOException {
		HashMap<?, ?> sectionMap;
		try (ObjectInputStream inputStream = new ObjectInputStream(new FileInputStrea
			sectionMap = (HashMap<?, ?>) inputStream.readObject();
		} catch (ClassNotFoundException e) {
			System.out.println("Cannot load your note...");
			e.printStackTrace();
			return;
		}
		System.out.println("Loading your note...");
		double size = sectionMap.size();
		double counter = 0;
		int nextSectionId = -1;
		int currentSection = -1;
		model.setSectionMap((HashMap<Integer, NoteSection>) sectionMap);
		List<Observer> observers = model.getObservers();
		for (Object o : sectionMap.values()) {
			if (((NoteSection) o).getID() < currentSection || currentSection == -
				currentSection = ((NoteSection) o).getID();
			}
			for (Observer observer : observers) {
				((NoteSection) o).addObserver(observer);
			}
			if (nextSectionId < ((NoteSection) o).getID()) {
				nextSectionId = ((NoteSection) o).getID();
			}
			counter++;
			progress = (int) Math.round((counter / size) * 100);
			System.out.println(progress + "% loaded");
		}
		model.setCurrentSectionId(currentSection);
		model.setNextSectionId(nextSectionId + 1);
		model.currentFunctionality = NoteModel.Functionality.NULL;
	}

	public void deleteShape(NoteShape noteShape) {
		model.deleteShapeFromCurrentPage(noteShape);
	}

	public int getProgress() {
		return progress;
	}

}
```

Controller


View
NoteAddVideo.java
NoteDrawImage.java
NoteErase.java
NotePen.java
NoteSearch.java
NoteTextBoxGroup.java
NoteView.java

View



Trello

We use the Trello to coordinate our team, to assign the tasks, and it makes sure that everything is following the plans.

Basic Features

Drawing

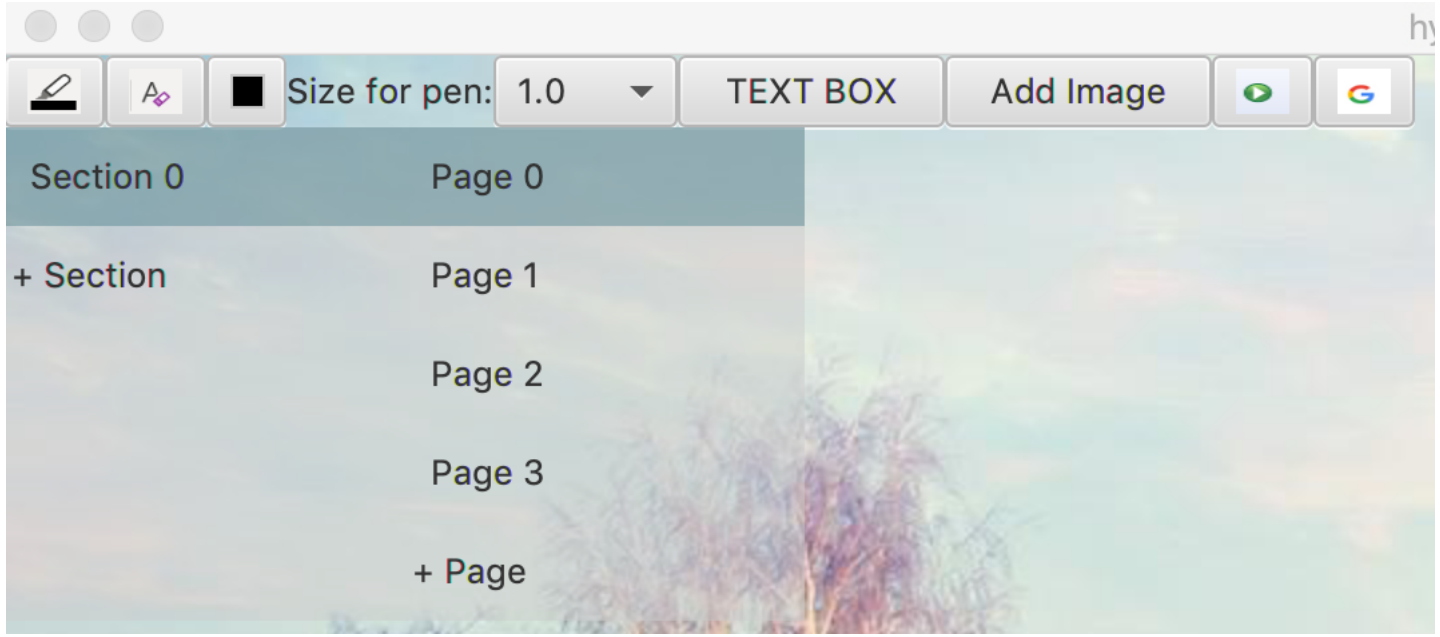
Eraser

Add Texts

Add Images

Change Text size/colors

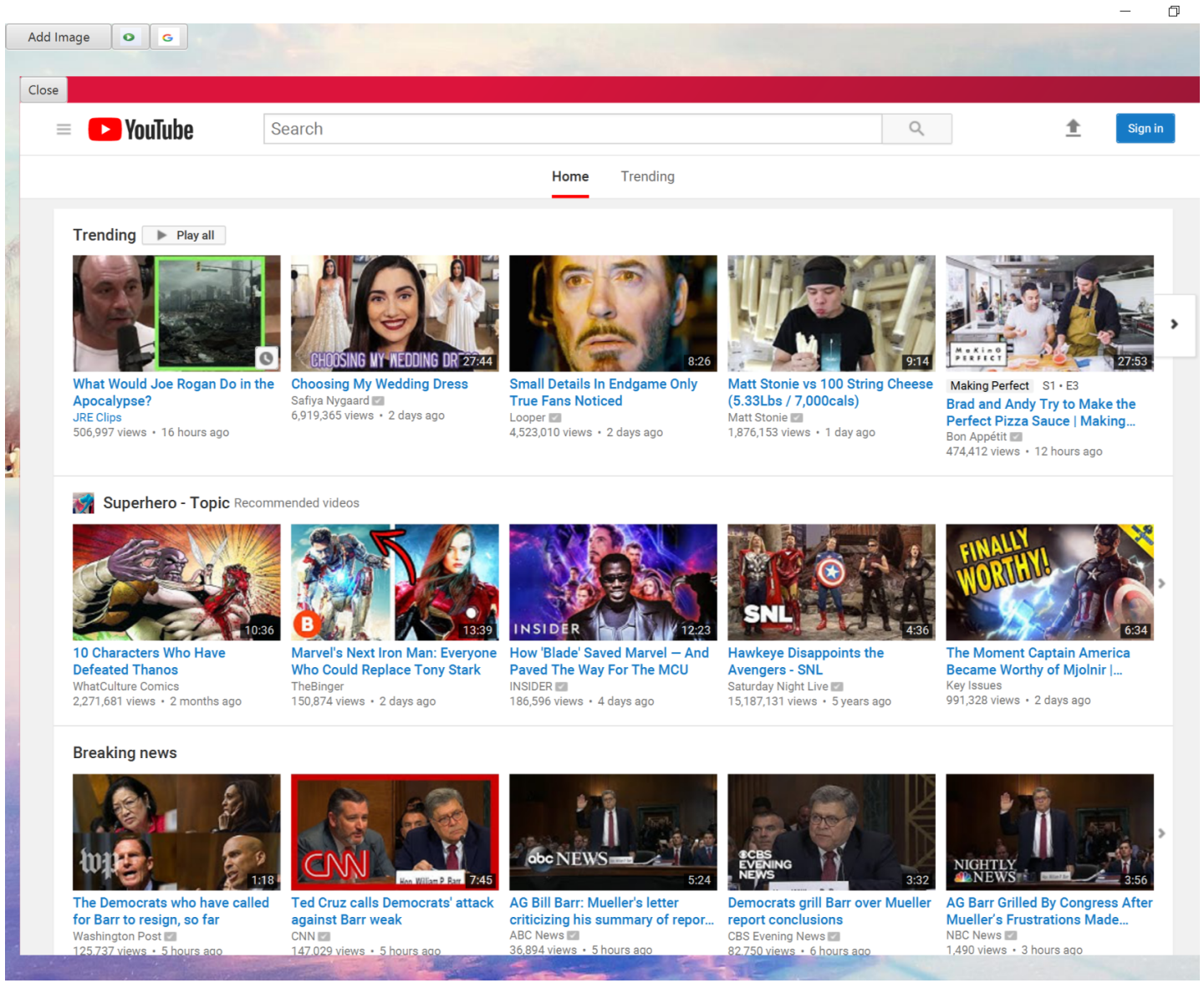
Switch sections/pages



Wow Factors: Built-in Web Browser

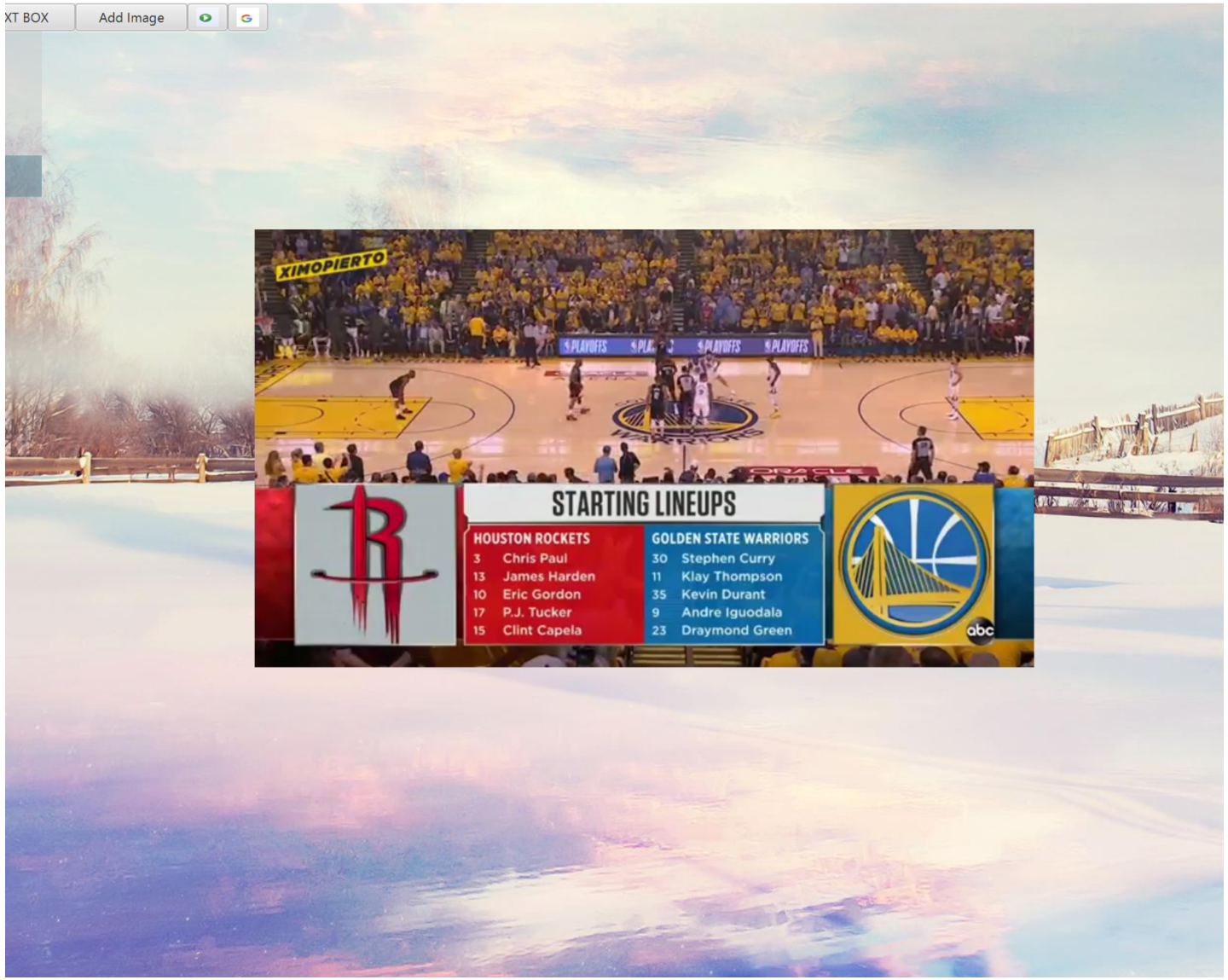
We choose to build a function that allows the user to open a webpage within our app. And the app would remember the latest pages that the users viewed. With this feature, users are abled to save some webpages inside their notes.

And with the Web Browser, this is app is not longer a simple note taking app, users can play games through web pages in the app.



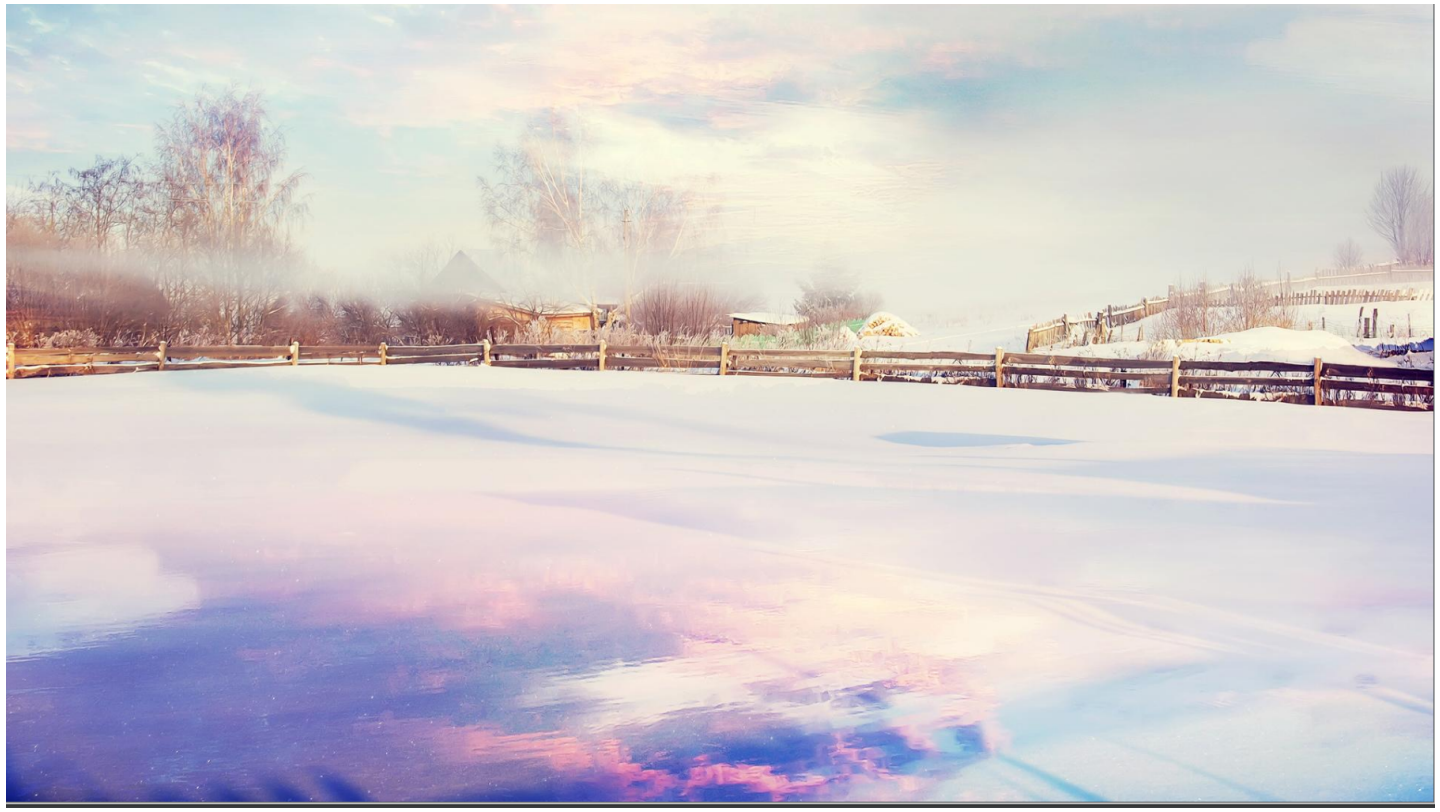
Wow Factors: Built-in Video Player

Users are allowed to import videos into the notes and they could save/delete the video easily. It is nice to have this feature since the format of notes are not simply texts or drawings.



Design Choices: Background Image

We put a background image in for the users. The purpose of this choice is to make the note app looks nicer, and give the users a relax atmosphere.

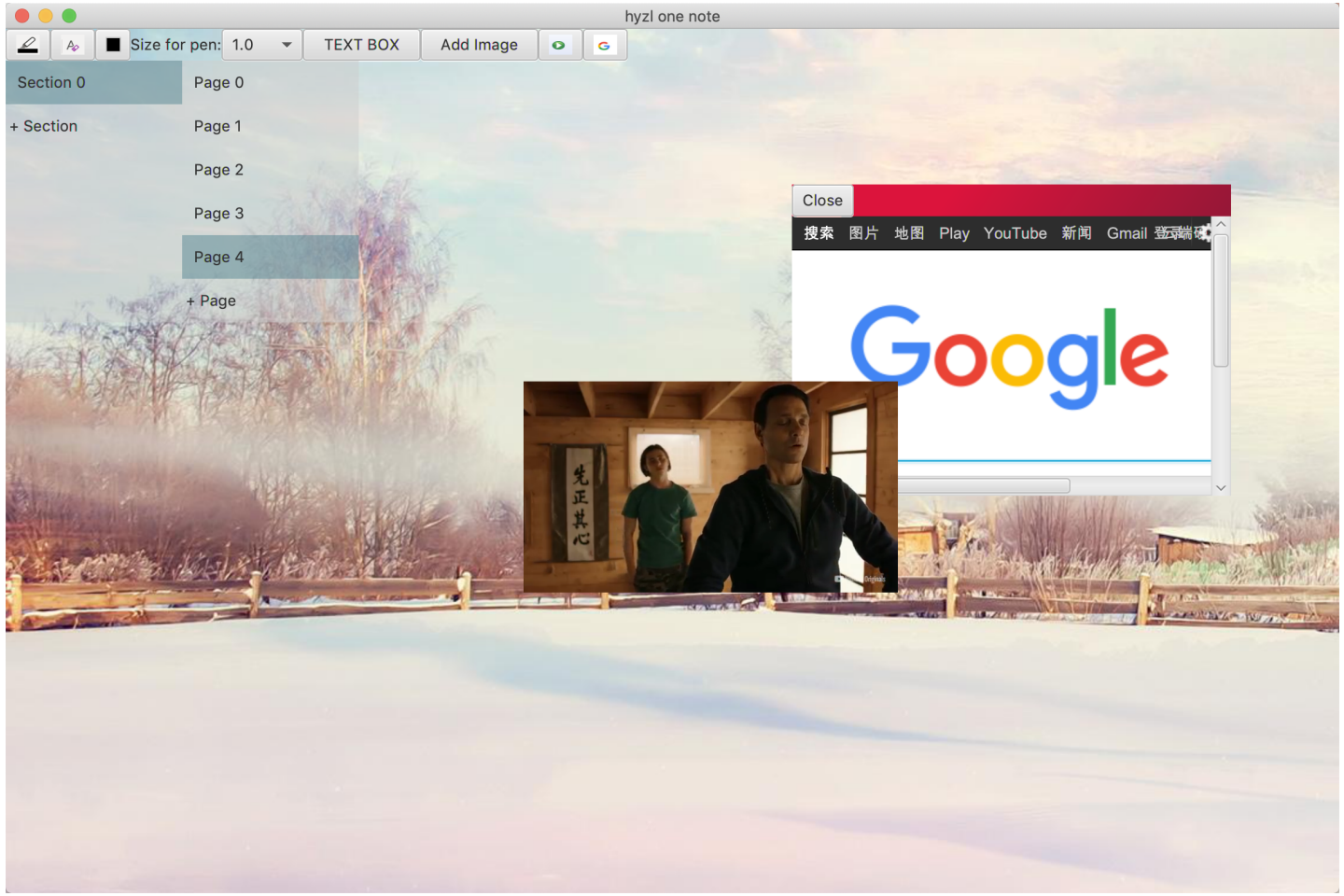


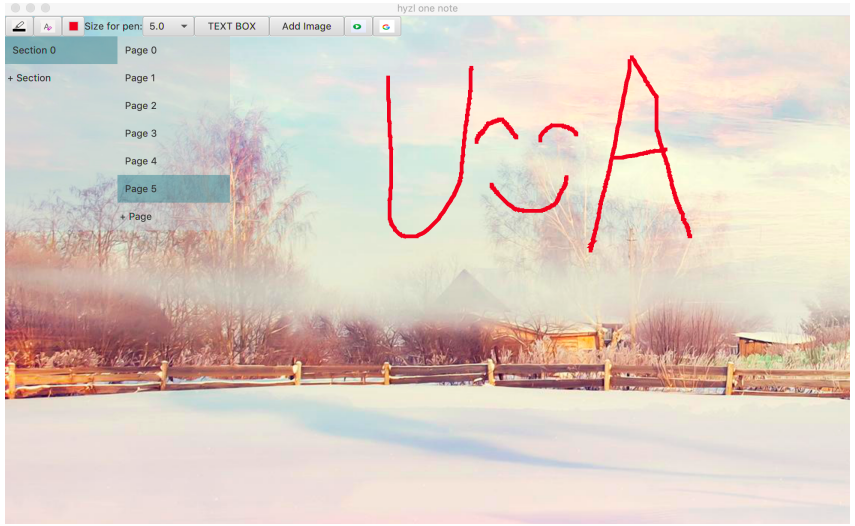
Design Choices: the Buttons

We horizontally align the buttons, such that they looks a lot more tidy. And for each buttons, they all have either image or text to explains the functions of each buttons to the users.

Design Choices: Draggable video/Web Browser

The video and web browser can be dragged so that the users can put them at any places they, it is easier to user, the users can organize the content of note easily.





Design Choices: Pen/Eraser

The users are allowed to draw and erase on the note, it is very convenient for them to takes notes.

The styles of pen and eraser are also carefully design, it makes the app attractive.

Design Choices: Text Box

We let the users to put the text inside a text box. And they can change the color/size of the text, and they are also allowed to drag the box. The reason we design this is to allow the users to modify the content of the text easily.