# Midterm Report: Parsons Coaching

Da An, Yuan Tian

March 2019

# Contents

# 1 Introduction

Our goal was to work on Parson's Coaching features under Gameful Direct module. The goal mainly consist of several improvement on the existing user interfaces.

During the last month, we started with getting familiar with the code base and data models, and moved on to implement the features that we proposed in the beginning. We also fixed some bugs in "sage-frontend". In the next section, we will introduce in detail about how each feature was implemented by us.

# 2 Implementations

## 2.1 Parson's Coaching Presentation

We made all conversation during an assessment session consistent by moving the hint, instruction and question into a dialog box. Student will feel more engaging with the avatar and can always expect information that they need appears in the dialog box.

In our implementation, instead of making the new messages override the old messages in the old design, we made the new message presentation more like chatting. In this way, students will not only feel more engaging with the avatar but also can always looks back in history to reference the hints and questions that he requested. The new user interface is displayed in the figure 1

## 2.2 Parson's Avatar Time Reminder

The current implementation of time section is detached from the Avatar. Under the guidelines of Gameful Direct Instructions, we made the Avatar associated with the Avatar by adding a time reminder feature to the Avatar. Besides, student will become more engaging during the assessment session.

The avatar can now remind students how much time they left in current session. When current session has 3 minutes left and 1 minute left, the Avatar will pop up a message to remind students. Shown in figure 2

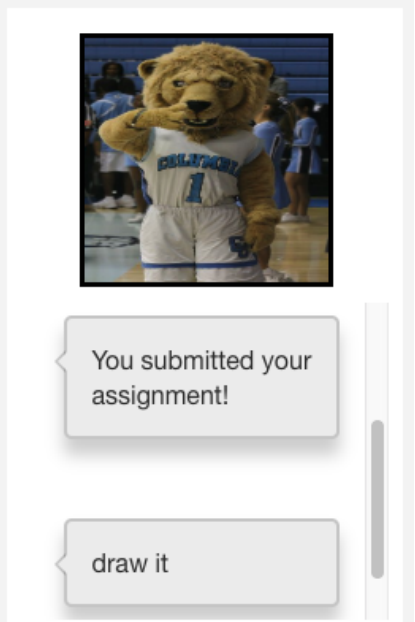


Figure 1: Message Presentation

## 2.3 Parson's Submit Presentation

To enable student to have a better overview about their performance in a particular assessment session. We include a summary section after students finish one session. The summary session currently consists statistics about the number of hint used and the time used. Shown in figure 3

Apart from that, we also included a history comparison section that compares the performance of current session with his/her previous submissions on this question. Students can know how many time they did this assessment and the average score that get from this assessment. However, it turns out this is not a desired feature that listed on TFS. We will remove this features later in the repository.

However, current statistics about hint usage is not perfect. One problem with current implementation is that student may by double-counted for the number of hint they used. If student press "hint" button twice, or student request two hints but got the same hint twice, or there is no hint available. Student should not be considered as use hint twice. To avoid, some cache are needed at front-end so that the system can monitor how many hint it used in total. Also, at current implementation, every time a student click on "hint"

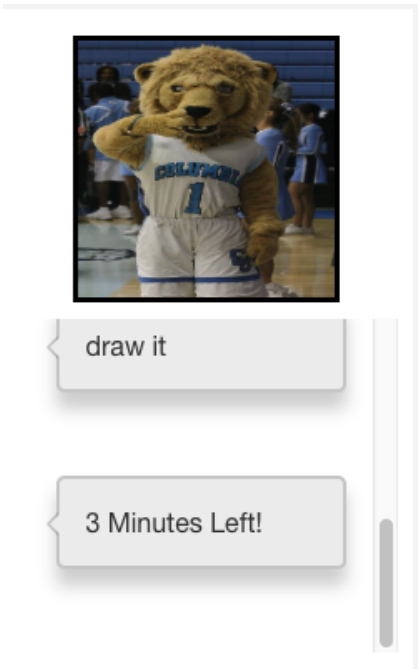


Figure 2: Time Reminder

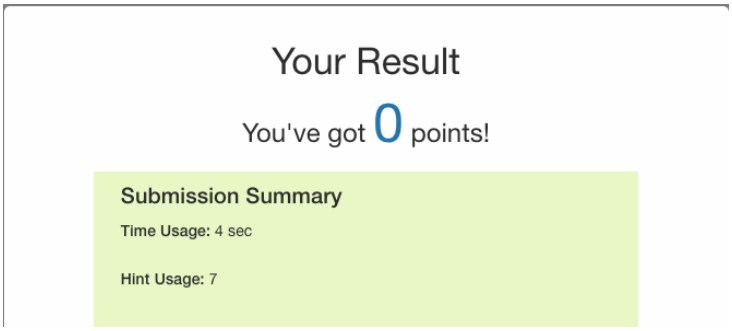


Figure 3: Submit Presentation

button, a HTTP call will increment the hint usage statistic in the database. This will cause inconsistency in front-end statistics and database. We could change it to make one HTTP call at the end of the submission.

## 2.4 Parson's Feedback Authoring

### 2.4.1 Instructor

On instructor side, we added a modal for instructors to write comments for each game which will later be reflected in the avatar session. Instructors will be able to write comments either based on the current comment templates we provide or create new comments. The comments are classified to good move or bad move which will later be shown in avatar sessions based on the students' performance. The comments related to different games from different instructors will later be stored in the MongoDB and can be referred to in the future for study.

If instructors choose to use the existing templates , they will be provided a drop-down menu to choose which type of template to use first, then retrieve the selected template based on the type. After that, instructors will be provided with the comments from the template and they will be able to edit these templates for one time and send them to MongoDB. The process is shown in the figure 4, 5 and 6.

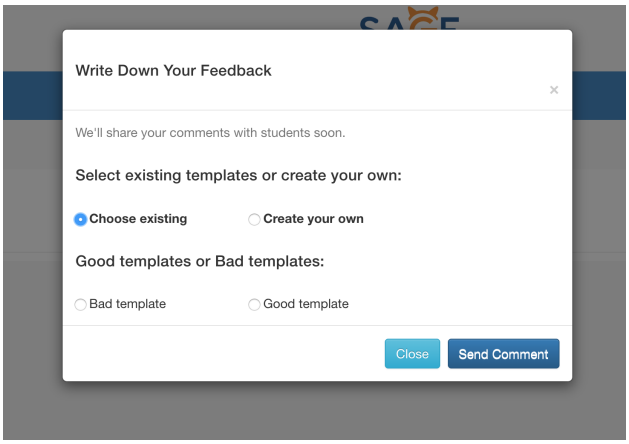


Figure 4: Choose the existing template

If instructors choose to create their own comments, they will be able to add at most five comments each time. If they would like to add more, then they can click the write another button below. After adding, modifying, or deleting comments, they can store these comments by pressing the send comment button. If the comments are sent successful, they can see the change reflected on the page shown in figure 7, 8 .

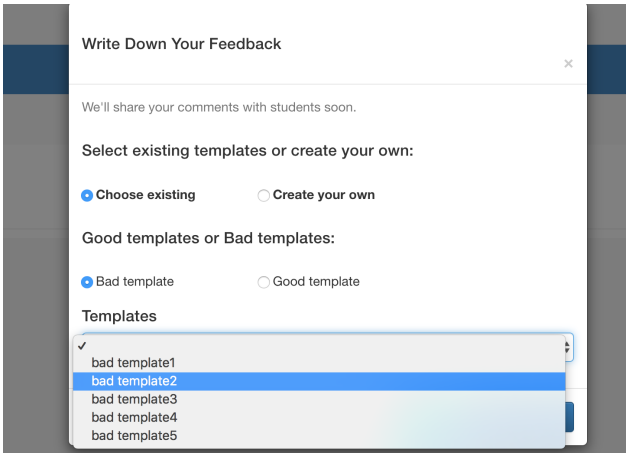


Figure 5: Select the template

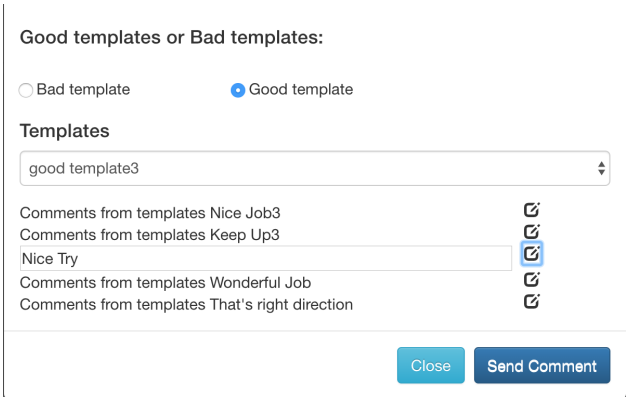


Figure 6: Edit the existing template

The data model to use to store the comments from instructors have four attributes: the instructor id, the game id, the result type and the comment string.

### 2.4.2 Student

On the student side, frontend can detect whether the student makes a good move or bad move by pull data from database, which was created by sage-scratch. Since sage-scratch can only provide a few kinds of feedback, frontend can know the type of moves using a dictionary. After frontend knows the

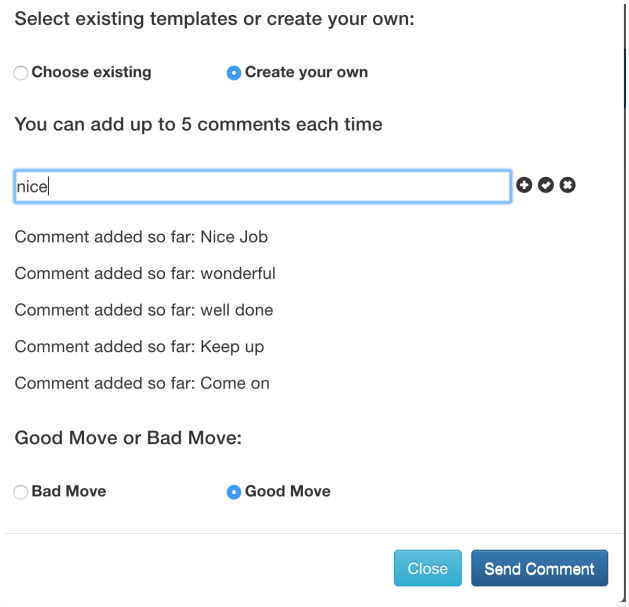


Figure 7: Create comments

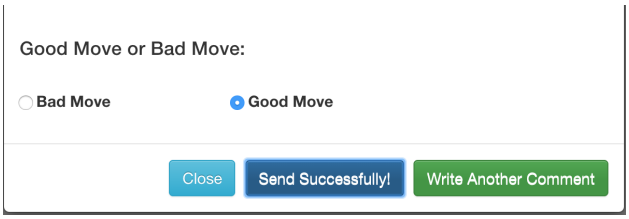


Figure 8: Send comments

student move type, we can then random sample a customized feedback from the database and present this feedback to the student.

# 3 Other Works

## 3.1 Bugs Fixed

We also did some work that was not appears in our initial proposals. We put them in this section as they are not as important as other features.

### 3.1.1 Learning Path Fixed

We fixed the issue when instructor add a new mission/learning path to a class, students cannot receive any updates. The problem was that previous students who developed this part did not implement this features at all. All learning path that was on the student client was hard coded. Now, the students mission page is looks like following figure 9


Figure 9: Learning Path Presentation

### 3.1.2 Homepage Fixed

We fixed the issue that students are not expected to see the quests that he/she hasn't been enrolled on homepage. Previously, students can see all the quests in the database. However, since student not are enrolled in all quests, student cannot access the quest after clicking on those quests. Instead, student should only see the quests that they have access to. We fixed this bug.

# 4 Next-steps

## 4.1 Parson's Submit Presentation

### 4.1.1 Hint Usage Statistic

As mentioned earlier, current implementation about hint usage may cause consistency between front-end and database. Also, the number is not accu-

rate as it has to be associated with the hint content but not just the number of button clicks

### 4.1.2 With Student Metric

In current implementation, the metric, shown in figure 10, are put into a separate page. To make the student more stimulated after each game and follow the goal of "Gameful Direct Instructions", we proposed that we could add the student metric to the summary section after each submission. Students can see how much does the metric changed after this game, which capabilities they learned and by how much. In this way, students can be more encouraged to do more game to makes the metrics look better. However, we would need to consider how to show this improvement in student metrics.

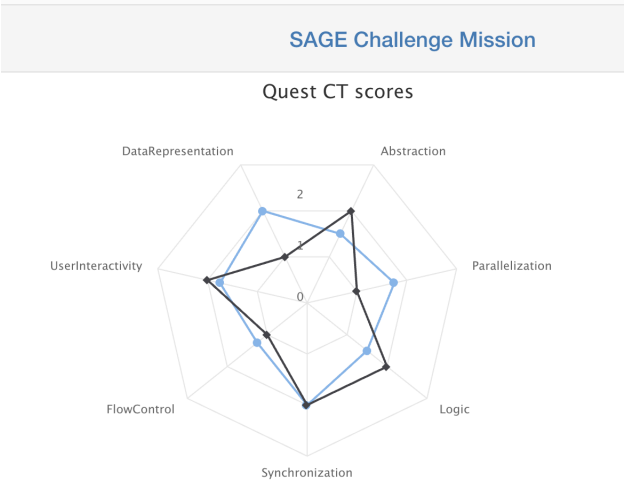


Figure 10: Metrics

## 4.2 Instructor Feedback

### 4.2.1 Feedback Template

We are current making several templates for instructor to provide feedback when student makes a good/bad move. Those templates makes instructors

completes this the feedback much faster. We now store and access some instructions from instructors and students perspective.

However, several tasks in front of us makes a good template suggestion for instructors. First is about the workflow, are we expect instructor to provide feedback for each game? Can we have some default feedback? We are still considering those problems. Second, we would like to makes the template "smart". "Smart" can be achieved in several ways:

1. We would like to combine with the game itself such that we can provide different templates based on the game type. For example, in Parson's puzzle, some game is focusing on "looping", some game is focusing on "condition". There are several ways to implement this. First, we can provide different template based on the games' tag. However, in current data models, there is no tag associate with games, to make this to work, we would need to add tag to each games

2. We would like to make the template become non-static. For example, the template can takes on student name or assessment name. In this way, when the feedback is presented to the students, students will feel more interactive. To be more specific, we want to enable instructors to set the feedback as follows: "Hi {{studentName}}, you did great! " or " Remember the way that we discuss on {{gameName}}". And the student will see: "Hi Da, you did great" etc.. To implement this feature, some work need to be on the instructor side, such as validate the placeholder names, check for ambiguity etc.

3. We would add more some criteria for instructors to filter the template when the templates become more complex with more features, the instructors will not be able to choose the template only based on the name of each template, so we propose to add some filters for teachers to choose appropriate templates.

These features follow the requirement under Gameful Direct Instructions.

### 4.2.2 More Feedback Format

Apart from providing instructors the opportunity to give feedback based on student's individual moves, we propose that we can allow instructors to provide feedback during the submit session. That is, instructors can also

provide feedback after student submit. The feedback can be customized the based on the hint usage, time usage and score.

# 5 Conclusion

Over the past few weeks, we made several implementations to follow the requirement of "Gameful Direct Instructor". We have almost finished all the features that we proposed at the beginning of the semester. However, some features still has lot of room for improvement (e.g. feedback authoring). In the second half of the semester, we will dig deeper into the work that we did fix bugs on the current web pages, and spend some time to add more features.