View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Institutional Knowledge at Singapore Management University

Singapore Management University
Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems

School of Information Systems

10-2014

Enabling National Software Development Competitions to Identify and Enhance Student Mentor Capability in Singapore

Chris BOESCH
Singapore Management University, cboesch@smu.edu.sg

Sandra BOESCH
Pivotal Expert Pte Ltd, sandracboesch@gmail.com

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

Part of the Education Commons, and the Software Engineering Commons

Citation

BOESCH, Chris and BOESCH, Sandra. Enabling National Software Development Competitions to Identify and Enhance Student Mentor Capability in Singapore. (2014). *Ireland International Conference on Education IICE 2014, 27-29 October*. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/2047

This Conference Paper is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email libIR@smu.edu.sg.

Enabling National Software Development Competitions to Identify and Enhance Student Mentor Capability in Singapore

Chris Boesch PhD
School of Information Systems (SIS)
Singapore Management University (SMU),
Singapore
cboesch@smu.edu.com

Sandra Boesch PhD
Pivotal Expert Pte. Ltd.
Singapore
sandracboesch@gmail.com

Abstract

The authors previously developed a system to facilitate the self-directed learning and practicing of software languages in Singapore. One of the goals of this self-directed learning was to enable the creation of student mentors who would be able to assist other students during classroom sessions. Building on this work, the authors extended the platform to support the promotion and coordination of multiple programming competitions including multiple schools systems within Singapore with the goals of identifying, enabling, and mentoring students who might be better prepared to mentor their peers at their school after participating in the country wide competition. This paper covers the challenges, insights, and features that were developed in order to register and select students in Singapore to participate in coding competitions with a limited number of available competition slots.

1. Introduction

Over the past few years the authors have developed and extended an online platform called SingPath [1] to promote the self-directed learning and practice of software languages, and to experiment with ways to make the time students spend together in classrooms more efficient. The authors also introduced adaptive gameplay to allow players to moderate and pick individual difficulty levels to encourage self-directed learning and programming practice [2]. Additionally, quests, a collection of short videos that are shown to students as they solve new problems, can be used to provide relevant tutorials or can be designed to promote curiosity and a more fun learning experience for the students [2].

In order to assess the capabilities of students in classroom settings the authors previously outlined a process to conduct in class lab sessions as competitive tournaments [3]. Research was also conducted to leverage the capabilities of more advanced students by automatically assigning them to mentor less capable or less prepared students [4]. Building on this work, the authors extended the SingPath platform to support the promotion and coordination of multiple programming competitions within Singapore with the objective of identifying, enabling, and mentoring students from multiple school systems so that students might be better prepared to mentor their peers at their local schools after participating in nation wide competitions.

Three limited sitting SingPath programming competitions were launched in early 2014. Due to sitting restrictions, a fair and open process was needed in order to select the students who would be invited to participate and compete in the programming tournaments. The qualification process developed was similar to the second chance admissions tournament [5] that the authors had previously supported. This qualification process, unlike the second chance tournament, did not have a pre-selected student population. Instead, each competition was made available to all qualified students in Singapore and the process called for a completely self-serve system for the students.

2. Education in Singapore

Singapore is a city-state in Southeast Asia consisting of approximately five million inhabitants. The Singapore government spends approximately twenty percent of the national budget on education [6]. Children in Singapore typically attend primary school between the ages of six and twelve years old and then attend Secondary School between the ages of thirteen and seventeen or eighteen years old depending on the secondary diploma earned. Upon finishing secondary school, the majority of students go on to complete either a two-year program at a

Junior College or a three-year program at a Polytechnic University.

The three SingPath programming competitions were announced to promote software development skills among Secondary School, Junior College, and Polytechnic University students. Students in each of these school categories were invited to register for upcoming competitions and to begin solving problems on SingPath.com. Python was selected as the language for the Junior College and Polytechnic University competitions and JavaScript was chosen for the Secondary School competition. A distinct URL link for each competition was provided to a group of Secondary School, Junior College and Polytechnic University instructors in Singapore who were then invited to share the competition links with fellow instructors.

To provide additional incentives to students, a MacBook Air was donated as the grand prize for each competition. These grand prizes were sufficient to drive initial interest and see between fifteen to fifty students from each school category sign-up within the first three weeks of the competitions' software launch. This paper covers the features and process developed and refined to support the soft launch of these three events.

3. Qualification Process

The Events feature on SingPath.com handles the qualifying and sorting process for each programming tournament. The Events feature not only enables players to register for an event but it allows all students who register to see an event ranking of qualified students. These event rankings were initially based on the number of problems that players had solved on one of the paths provided by SingPath.com. On SingPath, paths are an ordered collection of levels, and levels are an ordered collection of problems. Therefore, as students registered for the appropriate event being Secondary School, Junior College or Polytechnic University, and solved problems on the appropriate SingPath path, they were able to see themselves moving up the Event ranking.

The player ranking view provided real time feedback on how each individual was doing relative to all other students practicing either Python or JavaScript to qualify for their specific event. In addition to the ranked list of players, the event registration cutoff point of forty players is clearly shown in the ranking, along with how many problems each player has solved. When the current player is not in the top forty ranking of players for the event, they are able to see themselves below the fortieth player. This view provides the player with feedback on how far they are from catching the fortieth player and qualifying for the programming tournament. For the Junior College and Polytechnic University competitions the cutoff limit was set at forty players due to the networked venue that had been selected. The Secondary School cutoff point was set at twenty students due to additional non-competition activities that were planned for the students. These activities, which involved teaming with local industry mentors, were deemed to be more appropriate for a smaller audience of students.

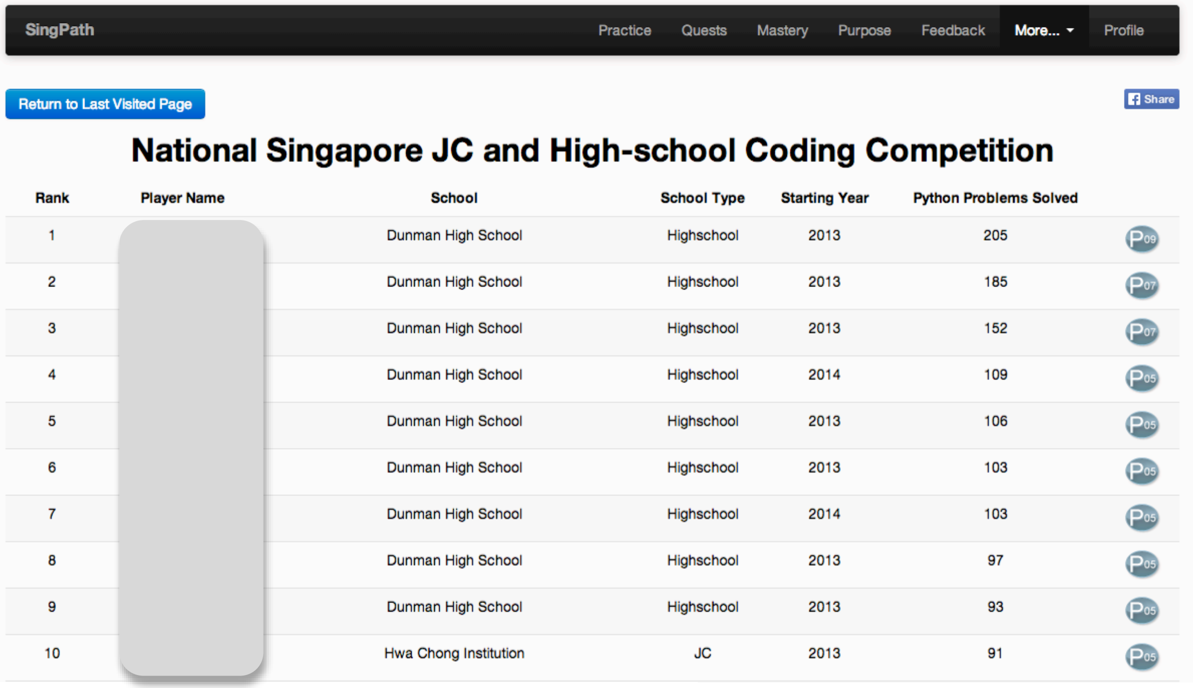
The qualifying process specified that all students who ranked above the cutoff point one week prior to the competition dates would be invited to the programming tournament. All invited students would then have three days to RSVP to their invitations before additional students from below the cutoff point would be invited to compete in the event.

These event qualifying features which implement event registration, rankings, and cut off points were sufficient to motivate eighty-six students to register and begin solving problems during the three week event qualifying soft launch. This figure equated to eighty-six percent of the approximately one hundred competition slots that were available across the three nation wide competitions. This early interest and participation by the students provided initial data to refine the process and qualification approach for the three programming competitions. An initial point of interest during the soft launch was that at least one student from each the Junior College and Secondary School events spent over eight hours practicing their respective software languages. The three-week soft launch data is included in Figure 1.

4. Cumulative Progress vs. Competition Window

While few of the registered students had played SingPath before the tournament registration was announced, it became clear that the process would need to be refined to make sure that the event qualification which was based on the cumulative number of problems that had been solved on SingPath would not give the advantage to students who had previously played on SingPath versus those students who had not used the platform in the past.

In future competitions, any student that had participated in prior tournaments would have a significant head start in qualifying for any forthcoming programming tournaments compared to new comers, if event qualifying were based on cumulative, lifetime number of problems solved. On the other hand, a point was raised by school delegates that tracking cumulative numbers could be beneficial to participants in some instances since it would clearly communicate that students who practice throughout the year and during the school term, could more easily qualify for future events. However, it was also highlighted that this process could also be a barrier to entry for students whose schools do not use SingPath or for students who practice with different paths of problems rather than the ones used for event qualification.



National Singapore JC and High-school Coding Competition

Rank	Player Name	School	School Type	Starting Year	Python Problems Solved	
1	[redacted]	Dunman High School	Highschool	2013	205	P09
2	[redacted]	Dunman High School	Highschool	2013	185	P07
3	[redacted]	Dunman High School	Highschool	2013	152	P07
4	[redacted]	Dunman High School	Highschool	2014	109	P05
5	[redacted]	Dunman High School	Highschool	2013	106	P05
6	[redacted]	Dunman High School	Highschool	2013	103	P05
7	[redacted]	Dunman High School	Highschool	2014	103	P05
8	[redacted]	Dunman High School	Highschool	2013	97	P05
9	[redacted]	Dunman High School	Highschool	2013	93	P05
10	[redacted]	Hwa Chong Institution	JC	2013	91	P05

Figure 1. SingPath Programming Tournament Event Qualification

A possible solution to this head start issue would be to attempt to create an entirely new set of practice problems for each event qualification. In this instance a new path of problems would be created on SingPath and progress on this new path would be used to rank registered students. This approach was considered overly resource intensive since based on the soft launch data, the best students were likely to solve over two hundred problems in just a few weeks in order to qualify for events and the effort required to create and review two hundred new practice problems in a particular software language is substantial.

To address the cumulative progress qualification concerns, the authors added a quest qualification ranking feature that enable event qualification ranking by working through a quest rather than practicing a path. In previous work the authors developed a quest feature, which requires SingPath players to solve five problems before being able to see a short video clip [2]. Since creating new quests only requires the selection of a series of video clips and selecting an existing path of problems, creating new quests to support events is considerably less work than creating new paths of problems. Furthermore, whereas path progress on SingPath is cumulative, working through quests based on a path requires that problems be re-solved even if they have been solved previously. The support of quest event qualification also provides events organizers with an opportunity to present the event participants with a series of videos as they work towards qualifying for events. These clips can be any video publicly available on YouTube or any MP4 file accessible via a public URL link. For the three competitions discussed in this paper, cumulative path progress was chosen since few participating students had played SingPath previously and those who had played before had not solved many problems on the paths selected for the event rankings.

5. Qualifying Rounds

Within three weeks of registering, approximately ten of the Junior College students had already solved over one hundred problems with one student solving over two hundred problems (Table 1).

Table 1. Top Ten Ranked Junior College Students

	Problems Solved	Attempts	Time(hours)	Time(s)/ problem	Attempts/ problem
1	235	988	6.09	93.34	4.20
2	197	1055	8.63	157.74	5.36
3	185	707	7.10	138.11	3.82
4	168	501	4.67	100.17	2.98
5	129	605	10.49	292.64	4.69
6	121	294	7.21	214.40	2.43
7	120	474	2.89	86.73	3.95
8	118	816	5.67	173.10	6.92
9	104	613	6.41	222.04	5.89
10	102	611	4.30	151.67	5.99

The high level of student engagement was encouraging, but it also raised concerns related to the availability of practice problems, the amount of time needed to qualify for the upcoming competitions, and the timing in which different students across the country would become aware of the competitions. Indeed, school delegates raised concerns by contributing that some students might not have as much time to practice and qualify as other students. If the minimum time needed to qualify became too great or the uncertainty of how much time would eventually be required became too great, this could be discouraging to many students who might otherwise participate in the tournaments. This issue would apply to students that learn about the competitions later and must race to catch up to the early starters. A proposed solution to these potential issues is to hold qualifying tournaments leading up to the competition. Rather than only set the number of available slots in the finals, a plan could be put in place to provide every student that achieves some milestone with an opportunity to demonstrate their skills in a proctored environment. This solution would be similar to setting the number of qualifying heats at a track and field event based on the number of participants that were able to run some distance in a pre-determined amount of time. The top finishers for each qualifying round would then be invited to the final competition where the prizes and other recognition would be awarded. Supporting qualifying rounds would also reduce the potential impact of students qualifying for events with too much assistance from their friends and family.

6. Privacy and Fraud Considerations

In order to protect the privacy of the students involved, no actual names or proof of school affiliation were required prior to the day of the final competitions. This also means that it is possible for a student to have someone else doing the qualifying work prior to showing up for the live competition. While getting help in qualifying for events will not help the students once they are in the live, proctored environment, too much assistance could result in some invitations being awarded to less capable students. Additional qualifying rounds while requiring more time and resources would ensure that every student in the final competition had demonstrated an ability to solve problems on their own in a proctored environment. On the day of the competitions, students will be required to show their student ID cards which all Singapore students possess. This process will ensure that only qualified students are able to participate in each competition. During the competitions, students will still be allowed to remain anonymous and should they choose to, go by a pseudonym.

7. Conclusion

During the three-week soft launch of three national programming competitions, the authors observed that eighty-six students registered and proceeded to solve over four thousand software problems. These students were likely motivated by the opportunity to win one of the MacBook Air's offered as prizes as well as opportunities to represent their schools and impress their parents and instructors. Still, other students are likely to have registered just for fun and to see what the competitions were all about. With a limited number of seats available for the final competitions, the authors had to devise a meritocratic way for students to apply themselves in order to claim one of the available seats. As expected, some students solved as many problems as they could, spending over eight hours practicing software in order to claim their top spots in the event ranking. Some students solved just one to three problems while some students registered but solved no problems at all.

These students and the Singapore school delegates helped to highlight potential issues that could arise with registration and qualification of applicants. With these insights, new features and guidelines were implemented to facilitate a meritocratic system that produced diverse student participation from multiple schools.

In addition to supporting the three programming competitions discussed, the event qualifying system provides an easy way to extend the practicing platform developed for academic use to engage with industry partners who might be interested in seeing increased competency in a particular skill such as JavaScript or would simply like to raise their brand awareness among the best student developers. Additional support from industry could help to signal the value of software development skills by local companies and lead to increased awareness, motivation and interest from students.

8. References

[1] SingPath. (2014). Retrieved 15 February, 2014 from www.singpath.com

[2] Boesch, C., & Boesch, S. (2013). Adaptive Gameplay for Programming Practice. 5th Annual International Conference on Computer Science Education: Innovation and Technology (CSEIT 2013).

[3] Boesch, C., & Boesch, S. (2012). Tournament-based Teaching. 4th Annual International Conference on Computer Science Education: Innovation and Technology (CSEIT 2012).

[4] Boesch, C., & Steppe, K. (2014). Automated Mentor Assignment in Blended Learning Environments. 27th

Conference on Software Engineering Education and Training (CSEE&T 2014).

[5] Boesch, C., & Steppe, K. (2011). Case Study on Using a Programming Practice Tool for Evaluating University Applicants. 3rd Annual International Conference on Computer Science Education: Innovation and Technology (CSEIT 2011).

[6] Singapore Budget 2013. (2014) Ministry of Finance, Singapore. Retrieved 15 February 2013 from http://www.mof.gov.sg/budget_2013/expenditure_overview/moe.html