CodeSport: Increasing Participation in Programming Using Coding Tournaments as an Alternative to Hackathons

Michael J. Lee and James Geller
New Jersey Institute of Technology
Newark, New Jersey, USA
{mjlee,james.geller}@njit.edu

ABSTRACT

Though hackathons are successful in attracting large crowds, they may not be sufficiently effective for broadening participation in computing, because they lack appeal for underrepresented groups in computing, and for people with family and job obligations. We propose a contrasting model for creating interest in computing, by making coding a spectator sport. We present an experience report on the design and implementation of a coding tournament, including survey results that informed the design of the system along with post-event questionnaire data from participants, exploring their attitudes towards different coding events. We find that coding tournaments can be an effective and engaging alternative to hackathons, and that they can motivate some audience members to pursue more coding activities, and possibly even participate as competitors in future tournaments.

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

Hackathons; Broadening Participation; Computing Education

ACM Reference Format:
Michael J. Lee and James Geller. 2019. CodeSport: Increasing Participation in Programming Using Coding Tournaments as an Alternative to Hackathons. In *ACM Global Computing Education Conference 2019 (CompEd '19), May 17–19, 2019, Chengdu, Sichuan, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3300115.3309505

1 INTRODUCTION

Hackathons are typically multi-day events, where teams are challenged to create applications containing specific features within a given amount of time (these events are also called game jams when the focus is on creating games) [8, 23, 31]. Many people and organizations are using hackathons to increase exposure to coding, in hopes of generating excitement and broadening participation in computing (as well as networking with potential job candidates for programming jobs). Researchers anticipate that these kinds of events will attract women [26] and people with limited technical background [33]. However, hackathons also have some features that can be discouraging. Johnson [14] highlights three potential impediments [32], specifically discussing potential causes for the low participation in hackathons by certain groups [7, 30]. Individuals from underrepresented groups 1) fear that they will be treated differently from others, 2) lack self-confidence and/or self-efficacy in their technical abilities, and 3) with job and/or family obligations cannot or do not want to participate in coding events that last too long (but many hackathons are multi-day events).

Due to these issues, we propose an alternative coding event in this paper – making programming a one-on-one spectator sport where individuals can choose to be competitors or spectators. This may help minimize the three problems above as follows:

First, team activities (such as competing in a hackathon) can lead to negative treatment (or micro-aggressions) of minority team members, by *any* definition of minority. One-on-one competitions can help minimize these group biases, since they are largely solo activities. Furthermore, because it is a spectator sport, even if it were to be organized as multi-member teams playing against each other, the public nature of the event with an audience would reduce the opportunity for micro-aggressions that might occur in teams working in a more secluded environment.

Second, research has shown that females and minorities may not prefer participating in competitive events themselves (which may draw unwanted attention) [3, 5], but that they are satisfied *watching* competitive events [11]. Therefore, individuals interested in programming, but with (justified or unjustified) lack of confidence or self-efficacy in their own abilities, can self-select to be spectators instead of competitors. Watching may eventually lead to increased willingness to learn more about programming for some of these individuals (even if they do not choose to be participants themselves), especially if they see other people similar to themselves participating and succeeding as competitors.

Third, as a spectator sport, the length of a tournament is limited by the game rules. Consequently, organizers can approximate total times fairly accurately (while keeping them at a reasonable limit), and participants can plan their schedules accordingly.

In this paper, we explain the background behind our vision, and describe results from a questionnaire demonstrating that people would be willing to participate in (competing/watching) coding tournaments. We then present an experience report of an actual coding tournament, including setup, rules, and results.

2 BACKGROUND

Why make programming a spectator sport? We are interested in increasing the participation of underrepresented groups in computing education and in computing careers. We propose this approach

CompEd '19, May 17–19, 2019, Chengdu, Sichuan, China

ACM ISBN 978-1-4503-6259-7/19/05…$15.00
https://doi.org/10.1145/3300115.3309505

based on the observation that many young people are attracted by spectator sports to become more active in the sport disciplines of their role models themselves.

Lardinoit [16] explored how watching sporting events on television might influence subsequent desires, and found that 16% of French viewers of the Olympic Games reported that they felt more inspired to get active after watching the competitions. Relatedly, watching a sport has long been associated with encouraging children to participate in the sport themselves (in the popular press), although we are not aware of any rigorous study supporting this approach [19]. Meredith [19] lists *watching* as one of nine possible methods for achieving this outcome.

There is a precedent for making intellectual activities into a spectator sport. In 2014, the Millionaire Chess Open debuted in Planet Hollywood and was broadcast on television (TV) with the organizers expressing their desire to turn "chess into a spectator sport" that replicates the "atmosphere of the World Series of Poker" [27]. High profile chess games are already a TV spectacle, with the World Chess Championship Organization stating that televised chess matches are "expected to attract a global online and TV audience of more than 1 billion fans" [15, 27].

Similarly, Go has extremely high viewership numbers, particularly in East Asian countries [1]. In a match between DeepMind's AlphaGo Artificial Intelligence program and the world-class player Lee Sedol, Chinese-language coverage reached up to 60 million viewers [21], while English-language coverage reached 100 thousand viewers [32]. Moreover, Hikaru-no-Go, a Japanese animation series about Go (from the early 2000s), dramatically increased the popularity of Go worldwide, particularly among children [28, 29].

Computer gaming as a spectator sport, or eSport, has gained tremendous attention in recent years, even appearing in the Asian Games (organized by the Olympic Council of Asia). Tournaments for popular games such as Starcraft and League of Legends, both fast-paced real-time strategy games, can draw thousands of spectators at large arenas. In these competitions, the sports consist of using a complex real-time graphics game program. Our approach is to make programming itself a spectator sport.

Finally, we subscribe to the theory of situated learning [17, 20], which states that people learn through observation and interacting with others. We provide participants with an authentic experience [13] in the spectator event – creating, editing, and testing real code to satisfy specific requirements.

3 TOURNAMENT ENVIRONMENT

3.1 Hackathon Alternative? – A Viability Study

We created a survey to get a generalized view of people's opinions about hackathons and viability of our proposed coding tournament. First, we provided an objective description of each type of event, with information about the style and format of how the different types of events are run. Next, we asked a series of 5-point Likert scale questions [25] to determine how agreeable (with 1 being least agreeable, and 5 being most agreeable) participants would be to attend different types of competitive events as a spectator, and also how comfortable they would be in competing in these events with others watching them. Finally, we asked if they had ever attended a hackathon, if they have a computing-related job, and about their age and gender.

We report the range and median of our responses (since we collected ordinal data). In addition, since our data is not normally distributed, we use nonparametric Chi-Squared tests (which are commonly used to analyze ordinal data [9]) with $\alpha = 0.05$ confidence to compare participants' responses based on their demographic categories. We report all of our (Pearson) Chi-Squared statistics in Table 1.

We surveyed 200 people on Amazon Mechanical Turk (MTurk), limiting our participants to people living in the USA, and between the ages of 18-35 years old (the primary age range of hackathon participants [2]). We chose to use MTurk because we wanted to get responses from a wide range of people who might be better representative of the general public (as opposed to using a mailing list at our STEM-focused university, where more than half the student population are enrolled in engineering or computing-related fields). Our goal is to understand what types of computing events people would be willing to participate in as a spectator or competitor.

The task took a mean time of 3.12 minutes, and we paid each participant US$1.00 to complete the task. Our sample included 107 females and 93 males, was ethnically diverse (79 White/Caucasian, 46 Hispanic/Latino, 33 Black/African-American, 28 Asian/Pacific-Islander, and 14 multi-ethnic or other), and 87 people reported they had a computing-related job. Also, 19 people reported that they had been to a hackathon in the past (5 females and 14 males; all with computing-related jobs).

Table 1 shows the summary of all the statistics, including the median and range, calculated for each of our questions. For each question, we compared across five categories (gender, age, ethnicity, computing-job, and hackathon experience (H-Exp)) to see if there were any differences in responses by measure.

When asked if they would be willing to attend a 48-hr hackathon, a 12-hr hackathon, or a 3-hr coding tournament as a *spectator*, our participants had a median response of 2, 3, and 3, respectively. There was no statistically significant difference detected by gender, age, ethnicity, or job for either of the hackathons. However, we found a significant difference by participants' job and H-Exp for 3-hr tournaments. Those with computing-related jobs and H-Exp were more likely to report they would attend a 3-hr coding tournament as a spectator compared to those in non-computing-related jobs and people without H-Exp.

To the question whether they would be willing to attend a 48-hr hackathon, a 12-hr hackathon, or a 3-hr coding tournament as a *competitor*, our participants had a median response of 1, 2, and 3, respectively. There were no statistically significant differences detected by gender, age or ethnicity in participants' responses. However, we found a significant difference by participants' H-Exp (and trending by participants' job for hackathons). Those with H-Exp were more likely to report that they would attend a 48-hr hackathon, 12-hr hackathon, or a 3-hr coding tournament as a *competitor* compared to those without H-Exp.

Lastly, we attempted to establish baseline measurements by asking questions about chess and poker. When asked if they would attend a *chess tournament* or a *poker tournament* as either a spectator or competitor, our participants had a median response of 1 for all questions. There was no statistically significant difference detected by gender, age, ethnicity, job, or H-Exp for those questions.

In summary, we found that our participants were generally not enthusiastic about attending any of the proposed types of competitive events as a spectator or competitor (the highest median value was 3, which indicates they neither agree nor disagree that they would attend an event as a competitor or spectator). This was not completely unexpected, as the general public might not be interested in these types of events overall.

However, we did find that people who had computing-related jobs or previous experience with hackathons were more likely to report that they would attend hackathons (as a spectator or competitor) and coding tournaments (as a spectator and as a competitor). This is also not unexpected, as those who are more familiar with computing might be more open to attending computing-related events. Moreover, our data indicates that people are more willing to attend 48-hr/12-hr hackathons and coding tournaments as spectators (medians were 2, 3, and 3, respectively) than spectating a chess or poker tournament (medians were 1 and 1, respectively).

While we did not compare events directly against each other, these results (medians and statistical results) are good indicators that a coding tournament can be at least as attractive as hackathons (which do attract high numbers of attendees [2]) for the general public, those with computing-jobs, and/or past hackathon experience. Additionally, the shorter time-investment required by coding tournaments compared to hackathons may be attractive to more people. Finally, we were concerned that minorities and females may not be attracted to these competitive events. However, we did not find any differences by gender, ethnicity, or age for any of our questions, indicating that members of these groups might be willing to watch, or even possibly compete in these types of events.

We acknowledge the limitation of our sample, as our MTurk workers may not be representative of the general public. Nevertheless, MTurk workers are generally tech-savvy (as evidenced by the high number of our participants who self-reported that they have computing-related jobs and had attended hackathons), and may be a good representation of the people that are aware of these types of events and might consider attending. For example, we found that those who attended hackathons in the past were more likely to say that they would attend other hackathons and coding tournaments. We conclude from our survey results that there is sufficient interest to explore the viability of coding tournaments further.

I would attend a…	Statistic		p-val
48hr hackathon as a *spectator*	Gender:	$\chi^2(4,N=200)=0.757$	.946
	Age:	$\chi^2(8,N=200)=11.008$	.201
Median: 2 (disagree), Range 1-5	Race:	$\chi^2(16,N=200)=18.340$	.304
	Job:	$\chi^2(4,N=200)=4.240$	.375
	H-Exp:	$\chi^2(4,N=200)=3.223$	.521
12hr hackathon as a *spectator*	Gender:	$\chi^2(4,N=200)=3.182$	.528
	Age:	$\chi^2(8,N=200)=5.522$	.701
Median: 3 (neutral), Range 1-5	Race:	$\chi^2(16,N=200)=15.956$	.456
	Job:	$\chi^2(4,N=200)=5.255$	.262
	H-Exp:	$\chi^2(4,N=200)=4.863$	.302
3hr coding tournament as a *spectator*	Gender:	$\chi^2(4,N=200)=1.780$	.776
	Age:	$\chi^2(8,N=200)=14.267$	.075
	Race:	$\chi^2(16,N=200)=14.456$	.565
Median: 3 (neutral), Range 1-5	Job:	$\chi^2(4,N=200)=10.466$	***.048****
	H-Exp:	$\chi^2(4,N=200)=11.262$	***.043****
chess tournament as a *spectator*	Gender:	$\chi^2(4,N=200)=3.065$	.547
	Age:	$\chi^2(8,N=200)=7.242$	.511
Median: 1 (s.disagree), Range 1-5	Race:	$\chi^2(16,N=200)=7.020$	.973
	Job:	$\chi^2(4,N=200)=7.082$	.132
	H-Exp:	$\chi^2(4,N=200)=4.130$	.389
poker tournament as a *spectator*	Gender:	$\chi^2(4,N=200)=2.345$	.673
	Age:	$\chi^2(8,N=200)=7.492$	.485
Median: 1 (s.disagree), Range 1-5	Race:	$\chi^2(16,N=200)=17.319$	.365
	Job:	$\chi^2(4,N=200)=0.648$	.958
	H-Exp:	$\chi^2(4,N=200)=1.886$	.757
48hr hackathon as a *competitor*	Gender:	$\chi^2(4,N=200)=1.548$	.818
	Age:	$\chi^2(6,N=200)=5.403$	.493
Median: 1 (disagree), Range 1-5	Race:	$\chi^2(12,N=200)=17.554$	.130
	Job:	$\chi^2(4,N=200)=9.312$	***.054***
	H-Exp:	$\chi^2(4,N=200)=10.625$	***.047****
12hr hackathon as a *competitor*	Gender:	$\chi^2(4,N=200)=4.715$	.318
	Age:	$\chi^2(8,N=200)=6.570$	.584
Median: 2 (disagree), Range 1-5	Race:	$\chi^2(16,N=200)=9.997$	.867
	Job:	$\chi^2(4,N=200)=9.351$	***.053***
	H-Exp:	$\chi^2(4,N=200)=17.127$	***.032****
3hr coding tournament as a *competitor*	Gender:	$\chi^2(4,N=200)=4.504$	.342
	Age:	$\chi^2(8,N=200)=6.911$	.546
	Race:	$\chi^2(16,N=200)=6.766$	.978
Median: 3 (neutral), Range 1-5	Job:	$\chi^2(4,N=200)=10.234$	***.037****
	H-Exp:	$\chi^2(4,N=200)=10.419$	***.034****
chess tournament as a *competitor*	Gender:	$\chi^2(4,N=200)=3.090$	.543
	Age:	$\chi^2(8,N=200)=7.081$	.528
Median: 1 (s.disagree), Range 1-5	Race:	$\chi^2(16,N=200)=22.849$	.118
	Job:	$\chi^2(4,N=200)=3.603$	.463
	H-Exp:	$\chi^2(4,N=200)=1.070$	.900
poker tournament as a *competitor*	Gender:	$\chi^2(4,N=200)=3.275$	.513
	Age:	$\chi^2(8,N=200)=7.870$	.446
Median: 1 (s.disagree), Range 1-5	Race:	$\chi^2(16,N=200)=20.450$	.201
	Job:	$\chi^2(4,N=200)=1.942$	.746
	H-Exp:	$\chi^2(4,N=200)=1.610$	.807

Table 1: Survey results for participants.

3.2 Participants and Audience

Now that we had some evidence that people would attend a coding tournament, we brainstormed how to organize the activity and what the participants' experience should be. We realized that not every kind of programming activity can be adapted into a competitive sport, especially if audience members cannot judge the results and cannot compare the outcomes of two competitors.

However, we determined this matter is less of an issue if programming problems are limited to graphical programming tasks. Even audience members with no technical background can judge the similarity of pairs of graphical displays. We converged on a one-on-one tournament design concentrating on small, well-defined graphics coding problems that would potentially be exciting to watch for non-programmers.

3.3 Programming Environment

The strong connection between computer graphics and programming education has a long history, going back to the Logo language [22] from 1967 and its use of turtle graphics. We chose to use the *Processing* language[1] for our tournaments, which its developers describe as being designed "for learning how to code within the context of the visual arts" [10]. It is based on a simplified version

[1]The Processing language website: https://www.processing.org/

of Java, which is the major teaching language for programming in many US colleges and has a large established user base, especially among the visual arts community [10].

We considered using other popular (block) programming environments such as Scratch [24] and Alice [6]. However, we determined that using a textual language would offer more benefits to those interested in learning programming (e.g., learning the language *Processing* would enable a user to relatively quickly advance to Java) and that the animation capabilities of these other environments go beyond what is currently intended for this project. While animated graphics would provide interesting visuals for the audience, it creates challenges with respect to fair evaluation of the results of two competitors.

3.4 Setup and Procedure of a Game

We implemented the "CodeSport" tournament system in Java, requiring three computers (one for the referee, and one for each player) with internet connections. The referee's view contains a list of pre-arranged graphics programming problems, each consisting of a collection of shapes that we call *diagrams*. The referee sets a time limit on the screen, typically 5 minutes, and a required overlap percentage. A round of the game ends either at the time limit or when one of the players reaches the desired overlap percentage between her/his solution and the referee's diagram. To allow for more sophisticated shapes and fast-paced rounds, the referee may choose to provide players with parts of the solution code, requiring them to edit (i.e., debug) existing code instead of starting with a blank coding pane (which may be beneficial for less-experienced coders [18]). The referee's screen is projected on a large screen so that audience members can see the countdown timer and the players' current progress.

The players' view (see Figure 1), contains a code editing pane, the target image, the program/image the code generates, the timer, and the overlap percentage score. Each player's screen is projected on large screens so audience members can follow the player's coding progress and state of their diagrams. Many of the diagrams include repeated shapes and/or patterns, which would be most effectively solved by using arrays (e.g., to store information such as color) and loops (e.g., to repeat commands and cycle through arrays).

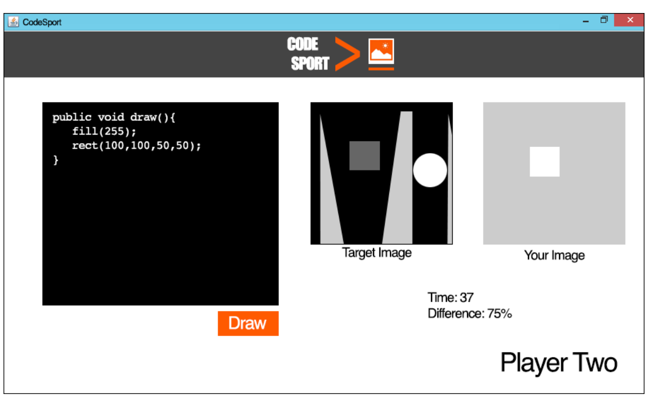

Figure 1: A player's screen showing the code pane (left), goal shapes (center), and current output (right).

We designed the tournament rules to be fast-paced, flexible, and to accommodate different time constraints and numbers of competitors. The winner of a best-of-three rounds wins the game. Each round has a recommended time limit of 5 minutes (±1-2 minutes, depending on task difficulty), making each game approximately 15 minutes. In an elimination tournament, players compete in one-on-one elimination games, with the winners advancing in their bracket. For longer tournaments, referees can choose to provide brief intermissions to give players and the audience a brief rest and to allow for announcements.

3.5 Features

In a CodeSport tournament, the basic procedure of one ***round*** of a game is as follows:

- The referee selects a diagram and displays it on the public screen (e.g., a half circle in red, slanted by 30 degrees with a black fill, as in Figure 2). This is the target shape that the two participants must match as closely as possible by *writing a program* in the Processing language.
- The referee clicks a button to activate both players' screens with the newly selected diagram.
- The two players attempt to reproduce the target diagram by writing the necessary code on their computers. This may require the use of different types of data structures, looping, and/or conditional statements.
- Players can push a button to execute their current code and display their current solutions.
- Once they are sufficiently satisfied with their progress, players can push a button to send the referee their diagram.
- The referee's computer performs a pixel-by-pixel comparison between the target shape/color and the player's diagram and returns the current overlap percentage.
- The audience sees the correct solution and the players' solutions at the same time on the public screens.
- Players can continue to submit their solutions until the referee's computer determines a winner.
- The referee's computer decides the winner of the round when a player's solution sufficiently overlaps with the goal solution, or the round reaches the time limit. We define "sufficient" as a numerical threshold parameter with a value between 95%-100%, to be set by the referee and/or organizers before the tournament begins. The referee predetermines the time limit for each round and players can see a countdown timer at all times. For rounds that reach the time limit, the player with a higher overlap percentage at the time of the deadline is the winner.
- The referee's computer displays the winner and shows the final outputs of both players at the end of the round. It also displays the overlap percentages and the winner's code.

4 CODESPORT EXPERIENCE REPORT

We ran a CodeSport tournament at a local STEM-focused university, recruiting participants through mailing lists and word-of-mouth. We offered free food and US$120 in prize money as incentive ($50 for 1st, $30 for 2nd, and $20 each for two 3rd place winners).

We had a total of 26 spectators (6 females, 20 males) and 16 players (2 females, 14 males) compete in the tournament. These 16 players were randomly grouped into eight pairs, and competed one-on-one, in single-elimination brackets. Each pair played one game (i.e., three rounds), with the eight winners advancing to the quarter-finals where they competed for the top four spots. These four winners played in the semi-finals and the two winners of these games were the finalists. When not competing, competitors stayed to watch others compete along with the other spectators. The tournament consisted of 15 games in total. Our first-place winner was a fourth-year, Asian, female student majoring in mathematics.

To determine the effectiveness of our coding tournament, we surveyed our 16 players exploring their attitudes towards participating in different kinds of competitive events. We e-mailed our players an online questionnaire two weeks after the event to minimize any novelty bias, and make a fairer comparison to past hackathons they may have participated in. Unfortunately, we had only collected the e-mail addresses of competitors (to distribute prizes), so were unable to contact those who were just spectators. We received a total of 12 responses—our sample had 1 female and 11 males, was ethnically diverse (4 White/Caucasian, 3 Hispanic/Latino, 5 Asian/Pacific-Islander), and included those with part/full time jobs (4 did not have jobs). Our respondents were composed of undergraduate upper-classmen (3+ years in college), and they were all in STEM-related majors with some prior programming experience.

We asked a series of 5-point Likert scale questions to determine respondents' 1) preference between hackathons and coding tournaments (with 1 meaning highly preferring hackathons, 3 preferring hackathons and coding tournaments equally, and 5 highly preferring coding tournaments), and 2) agreeability to statements regarding their experience with coding tournaments (with 1 being least agreeable, and 5 being most agreeable). We report the range and median of our (ordinal) responses. We use nonparametric Chi-Squared tests (likelihood ratios) with $\alpha = 0.05$ confidence to compare participants' responses based on their demographic categories with the understanding that our sample size is small (which is why we report likelihood ratios) and that the resulting statistics may not be widely generalizable. We report all of our Chi-Squared statistics in Table 2.

4.1 Hackathons vs. Coding Tournaments

Our first set of questions was designed to learn more about the participants' preference(s) between hackathons and coding tournaments. We provided objective definitions and examples of both types of events to remind participants of the differences.

When asked which type of event they would prefer being a *spectator*, our participants were split with a median response of 3 (range 1-5). When asked which type of event they would prefer being a *competitor* in, our participants leaned towards coding tournaments with a median response of 4 (range 1-5). There were no statistically significant differences detected by gender or ethnicity.

When asked which type of event motivated them to be a *competitor in the future*, our participants had a slight preference for coding tournaments with a median response of 3.5 (range 1-5). When asked which event type motivated them to *learn more about programming*, our participants leaned towards coding tournaments with a median response of 4 (range 1-5). There were no statistically significant differences detected by gender or ethnicity for any these measures.

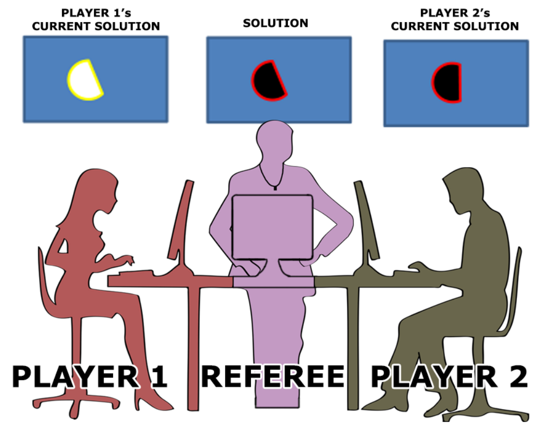

Figure 2: Stage setup with two players and a referee at their stations. The screens behind the referee display the goal shape and the players' current solutions.

4.2 Enjoyment of the Codesport Tournament

The next set of questions was designed to evaluate participants' experience with the CodeSport tournament, and whether they would consider attending in the future.

When asked about being a *spectator* for the coding tournament, participants indicated they enjoyed it with a median response of 4 (range 3-5). Although not quite statistically significant, Asians/Pacific Islanders and Hispanic/Latino players were more likely to report they enjoyed being a *spectator* compared to their Caucasian/White counterparts (see Table 2). Additionally, participants agreed they would consider attending a coding tournament as a *spectator* in the future, with a median response of 4 (range 3-5). There were no statistically significant differences detected by gender or ethnicity for these measures.

Hackathon vs Coding Tournament	Statistic		p.val
I would prefer to be a *spectator* at Hackathons ←→ C.Tournaments	Gender:	$\chi^2(2,N=12)=2.385$	.665
	Race:	$\chi^2(4,N=12)=11.998$	.151
I would prefer to be a *competitor* at Hackathons ←→ C.Tournaments	Gender:	$\chi^2(2,N=12)=3.065$	.547
	Race:	$\chi^2(4,N=12)=7.133$	.522
_____ motivates me to consider being a *competitor* in the future.	Gender:	$\chi^2(2,N=12)=6.884$	.142
	Race:	$\chi^2(4,N=12)=7.133$	.522
_____ motivates me to learn more about computer programming.	Gender:	$\chi^2(2,N=12)=3.065$	.547
	Race:	$\chi^2(4,N=12)=9.905$	.272
I enjoyed being a *spectator* for the coding tournament.	Gender:	$\chi^2(2,N=12)=1.142$	.564
	Race:	$\chi^2(4,N=12)=9.168$	***.057***
I enjoyed being a *competitor* in the coding tournament.	Gender:	$\chi^2(2,N=12)=0.856$	.652
	Race:	$\chi^2(4,N=12)=7.50$	.112
I would consider *spectating* at a coding tournament in the future.	Gender:	$\chi^2(2,N=12)=2.385$	.303
	Race:	$\chi^2(4,N=12)=3.175$	.529
I would consider *competing* at a coding tournament in the future.	Gender:	$\chi^2(2,N=12)=1.880$	.598
	Race:	$\chi^2(4,N=12)=12.401$	***.054***

Table 2: Post survey results for event participants.

To the question about being a *competitor* in the coding tournament, participants reported that they highly enjoyed the tournament, with a median response of 5 (range 3-5). Moreover, participants agreed they would consider attending another coding tournament as a *competitor* in the future, with a median response of 4 (range 2-5). Although not quite statistically significant, Asians/Pacific Islanders and especially Hispanic/Latino players were more likely to report they would consider being a future *competitor* compared to their Caucasian/White counterparts (see Table 2). There were no statistically significant differences detected by gender or ethnicity for any of the other measures.

5 DISCUSSION

5.1 Lessons Learned

We found that participants reported that they preferred coding tournaments over hackathons. Participants consistently reported that the CodeSport tournament was enjoyable, that they wanted to learn more about programming, and that they would consider attending future events as competitors or spectators. Although our post-tournament sample size was small, we found that our Asian/Pacific Islander and Hispanic participants' responses trended towards significance in reporting that they enjoyed being spectators (while not competing) and that they would also consider being competitors in future tournaments, compared to their Caucasian/White counterparts. Although our results are not conclusive, we saw some evidence that coding tournaments may be able to attract spectators who might later consider learning more about programming (and a subset of these people may possibly consider becoming competitors themselves). These are promising outcomes and something we will continue to investigate with more tournaments.

5.1.1 What Worked and Did Not Work. Overall, the CodeSport application worked well and the tournament succeeded in engaging the people who came out to participate in the tournament. However, this is also an issue, as we only collected survey data from the competitors, who are likely the most representative of the types of people who already have an interest in computing and attend events such as hackathons, and may be different from the people who came just to spectate. For future events, we will survey non-competitors, and we will try to appeal to a wider audience in our advertisements, focusing on recruiting people who are not in computing or closely related STEM fields.

5.1.2 Adoption by Others. The CodeSport software is compatible across many operating systems and is free to download and use. We would like to collaborate with others who are interested in using, adapting, and evaluating the system to further test the viability of coding tournaments as a way to attract a more diverse group of participants into computing.

5.1.3 Limitations and Future Work. There are several limitations to our experience report that limit its generalizability. First, our initial survey was conducted on Mechanical Turk, so our participants may not be representative of the general public. However, people did attend our coding tournament and indicated that they enjoyed the tournament and would attend again.

We also realize that the prizes and free food are factors that may have incentivized people to spectate and/or to compete in the tournament. However, these types of incentives are also common in hackathons and related events, so we assume the effects of these were minimal, and therefore a reasonable comparison.

Next, our post-tournament questionnaire had a small sample size and only represented those who participated as competitors, which limits how we can interpret the results. Moreover, the participants were all from the same university with some prior programming experience. Although our winner was a female, the tournament mostly consisted of college-aged males, which reflects the gender demographics of the university, and typical makeup of hackathon participants [2]. It would have been better for a larger, more diverse audience to witness the winner's achievements at the tournament.

We will attempt to address these issues by running more events and getting more feedback from our participants (especially from spectators). We will specifically try to recruit more female and/or minority participants, and include more collaborative elements (as opposed to competitive elements) to be more gender-inclusive and appeal to a wider audience [3–5]. We also plan to host middle/high school students to participate in CodeSport tournaments to learn how these types of events might appeal to and engage these younger students (and possibly contrast with older participants).

6 CONCLUSION

Educators and researchers have tried many different approaches to increase participation in computing education and in the computing professions [12]. Recently, hackathons have become popular and touted as an effective mechanism in broadening participation in computing-related activities. However, evidence suggests that certain people might feel uncomfortable or have other constraints that prevent them from participating in hackathons.

In this paper, we presented an alternative method for broadening participation in computing. We aimed to increase the public's interest in programming by making it a spectator sport, since there is evidence that a percentage of sport spectators will eventually take up that activity [16].

First, we conducted a survey that confirmed that people would attend and participate in coding tournaments. Next, we created the CodeSport system and tournament rules. We ran a tournament with 16 participants and received feedback from 12 of them. Participants reported that they highly enjoyed the coding tournament, that they preferred coding tournaments over hackathons, and that they would continue to attend future coding tournament events. Likewise, participants agreed that the event motivated them to learn more about programming and take part as competitors in the future. Given these results and feedback, we are encouraged about the viability of coding tournaments as an alternative to hackathons that may be able to attract additional people into computing through spectating and possibly competing.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under grant DRL-1837489. Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Steven Borowiec and Tracey Lien. 2016. AlphaGo beats human Go champ in milestone for artificial intelligence. *Los Angeles Times* 12 (2016).
[2] Gerard Briscoe. 2014. Digital innovation: The hackathon phenomenon. (2014).
[3] Margaret M Burnett, Laura Beckwith, Susan Wiedenbeck, Scott D Fleming, Jill Cao, Thomas H Park, Valentina Grigoreanu, and Kyle Rector. 2011. Gender pluralism in problem-solving software. *Interacting With Computers* 23, 5 (2011), 450–460.
[4] Margaret M Burnett, Elizabeth F Churchill, and Michael J Lee. 2015. SIG: Gender-Inclusive Software: What We Know About Building It. In *ACM Conference Extended Abstracts on Human Factors in Computing Systems.* ACM, 857–860.
[5] Anne Campbell. 2004. Female competition: Causes, constraints, content, and contexts. *Journal of Sex Research* 41, 1 (2004), 16–26.
[6] Tebring Daly. 2013. *Influence of alice 3: reducing the hurdles to success in a cs1 programming course.* University of North Texas.
[7] Adrienne Decker, Kurt Eiselt, and Kimberly Voll. 2015. Understanding and improving the culture of hackathons: Think global hack local. In *Frontiers in Education Conference (FIE).* IEEE, 1–8.
[8] Allan Fowler, Foaad Khosmood, Ali Arya, and Gorm Lai. 2013. The global game jam for teaching and learning. In *Conference on Computing and Information Technology Research and Education New Zealand.* 28–34.
[9] J Frost. 2016. Best Way to Analyze Likert Item Data: Two Sample T-Test versus Mann-Whitney. (2016).
[10] B Fry and C Reas. 2017. A short introduction to the Processing software and projects from the community. (2017).
[11] Walter Gantz. 1981. An exploration of viewing motives and behaviors associated with television sports. *Journal of Broadcasting & Electronic Media* 25, 3 (1981), 263–275.
[12] James Geller. 2016. Building, Recruiting, And Inclusion for Diversity. (2016). Retrieved March 7, 2017 from https://web.njit.edu/~geller/braid/Main.html
[13] Mark Guzdial. 2015. Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics* 8, 6 (2015), 1–165.
[14] Phil Johnson. 2014. Why hackathons are a boys' club and how to change that. (2014). Retrieved March 26, 2017 from https://www.itworld.com/article/2700151/networking/why-hackathons-are-a-boys--club-and-how-to-change-that.html
[15] Tovin Lapan. 2014. Chess as a spectator sport? Organizers of big-money tourney say yes. (2014). Retrieved March 26, 2017 from http://lasvegassun.com/news/2014/oct/15/chess-spectator-sport-organizers-big-money-tourney/
[16] Thierry Lardinoit. 2014. Watching the world cup: can TV encourage physical activity? (2014). Retrieved March 26, 2017 from http://knowledge.essec.edu/en/sustainability/watching-world-cup-can-tv-encourage-physical-activ.html
[17] Jean Lave and Etienne Wenger. 1991. *Situated learning: Legitimate peripheral participation.* Cambridge University Press.
[18] Michael J Lee, Faezeh Bahmani, Irwin Kwan, Jilian LaFerte, Polina Charters, Amber Horvath, Fanny Luor, Jill Cao, Catherine Law, Michael Beswetherick, et al. 2014. Principles of a debugging-first puzzle game for computing education. In *Visual Languages and Human-Centric Computing (VL/HCC).* IEEE, 57–64.
[19] Janis Merideth. 2014. 9 fun ways to motivate your child in sports. (2014). Retrieved March 26, 2017 from https://blogs.usafootball.com/blog/693/9-fun-ways-to-motivate-your-child-in-sports
[20] Sharan B Merriam, Rosemary S Caffarella, and Lisa M Baumgartner. 2012. *Learning in adulthood: A comprehensive guide.* John Wiley & Sons.
[21] Cade Metz. 2016. The sadness and beauty of watching Google's AI Play go. *Wired* (2016).
[22] Seymour Papert. 1980. *Mindstorms: Children, computers, and powerful ideas.* Basic Books, Inc.
[23] Jon A Preston, Jeff Chastine, Casey O'Donnell, Tony Tseng, and Blair MacIntyre. 2012. Game jams: Community, motivations, and learning among jammers. *International Journal of Game-Based Learning (IJGBL)* 2, 3 (2012), 51–70.
[24] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (2009), 60–67.
[25] Melanie A Revilla, Willem E Saris, and Jon A Krosnick. 2014. Choosing the number of categories in agree–disagree scales. *Sociological Methods & Research* 43, 1 (2014), 73–97.
[26] Gabriela T Richard, Yasmin B Kafai, Barrie Adleberg, and Orkan Telhan. 2015. StitchFest: Diversifying a College Hackathon to broaden participation and perceptions in computing. In *ACM Technical Symposium on Computer Science Education.* ACM, 114–119.
[27] Oliver Roeder. 2016. The World Chess Championship comes to New York City. (2016). Retrieved March 8, 2017 from https://fivethirtyeight.com/features/the-world-chess-championship-comes-to-new-york-city/
[28] Charles Scanlon. 2002. Young Japanese go for Go. (2002). Retrieved March 8, 2017 from http://news.bbc.co.uk/2/hi/asia-pacific/2164532.stm
[29] Yoko Shimatsuka. 2005. Do Not Pass Go. (2005). Retrieved March 15, 2017 from http://www.asiaweek.com/asiaweek/magazine/nations/0,8782,132162,00.html
[30] Shuba Swaminathan. 2014. Why don't more women go to hackathons? (2014). Retrieved March 26, 2017 from https://www.quora.com/Women-in-Technology-1/Why-dont-more-women-go-to-hackathons
[31] Jeremy Warner and Philip J Guo. 2017. Hack. edu: Examining how college hackathons are perceived by student attendees and non-attendees. In *ACM Conference on International Computing Education Research.* ACM, 254–262.
[32] Frank Wunderlich-Pfeiffer. 2016. Künstliche Intelligenz: Alpha Go spielt wie eine Göttin. (2016). Retrieved March 26, 2017 from https://www.golem.de/news/kuenstliche-intelligenz-alpha-go-spielt-wie-eine-goettin-1603-119646.html
[33] Jorge Luis Zapico, Daniel Pargman, Hannes Ebner, and Elina Eriksson. 2013. Hacking sustainability: Broadening participation through green hackathons. In *International Symposium on End-User Development.*