What Emotions Do Novices Experience during Their First Computer Programming Learning Session?

Nigel Bosch[1], Sidney D'Mello[1,2], and Caitlin Mills[2]

[1] Departments of Computer Science
[2] Psychology, University of Notre Dame, Notre Dame, IN 46556
{pbosch1,sdmello,cmills4}@nd.edu

Abstract. We conducted a study to track the emotions, their behavioral correlates, and relationship with performance when novice programmers learned the basics of computer programming in the Python language. Twenty-nine participants without prior programming experience completed the study, which consisted of a 25 minute scaffolding phase (with explanations and hints) and a 15 minute fadeout phase (no explanations or hints) with a computerized learning environment. Emotional states were tracked via retrospective self-reports in which learners viewed videos of their faces and computer screens recorded during the learning session and made judgments about their emotions at approximately 100 points. The results indicated that flow/engaged (23%), confusion (22%), frustration (14%), and boredom (12%) were the major emotions students experienced, while curiosity, happiness, anxiety, surprise, anger, disgust, fear, and sadness were comparatively rare. The emotions varied as a function of instructional scaffolds and were systematically linked to different student behaviors (idling, constructing code, running code). Boredom, flow/engaged, and confusion were also correlated with performance outcomes. Implications of our findings for affect-sensitive learning interventions are discussed.

1 Introduction

Computer science (CS) remains a difficult degree to complete and has some of the highest attrition rates in undergraduate universities in the U.S. [1]. There has been some research aimed at identifying the factors that contribute to the eventual success or failure of students in computer programming classes. Some of this research has focused on individual differences like mathematical ability, programming aptitude, and psychological traits of non-cognitive factors like temperament and motivation [2–5]. Many of these factors have proven to be somewhat influential in predicting a student's decision to enroll in a computer programming course, as well as their eventual success in such courses, but these trait-based measures are very coarse grained and assume fixed dispositions instead of malleable factors.

Taking a somewhat different approach, the present paper focuses on the emotions that students experience during their first encounter with computer programming. It is expected that flow/engagement is the ideal affective state in which students tend to be most capable of acquiring meaningful information through the learning process [6, 7].

K. Yacef et al. (Eds.): AIED 2013, LNAI 7926, pp. 11–20, 2013.

However, other emotions interact with flow/engagement and augment or detract from learning. For example confusion and frustration are expected to arise quickly when the results of a program do not match expectations (confusion) or the student has no idea how to proceed and gets stuck at a logical impasse (frustration). Persistent failure is associated with frustration [8] and lower self-efficacy, which can lead to boredom and disengagement [9], and ultimately attrition [10]. Therefore, our working hypothesis is that emotional factors play an instrumental role in the process of learning to program and can influence both immediate (failing an exam) and long-term outcomes (dropping out of a CS course).

There has been some research that has investigated the emotions that students experience while learning programming, as well as the effect of those emotions on eventual success in a CS class [11–13]. For example, [12] used two human observers to code student affect (boredom, confusion, delight, surprise, frustration, flow, or the neutral state) during 50-minute lab sessions. They found that confusion, boredom, and on-task conversation (i.e. asking for help) were negative significant predictors of performance on a midterm exam.

More recently, [14] collected several data sources while students conversed with a human tutor about the exercises they were completing via a computer-mediated interface. They found that frustration reported by students correlated (r = .53) with confusion reported by the tutor. Additionally, tutor reports of student confusion and frustration were correlated (r = .59), and confusion was negatively correlated with posttest scores (r = -.38).

These studies have provided some important insights into the emotions that arise when students learn to program and the influence of these emotions on performance. The long-term goal of this research is to develop advanced learning environments that detect and respond to student emotions. However, much more basic research on the emotions themselves is needed before such an affect-sensitive learning environment can be successfully engineered. As an initial step in this direction, the present study systematically tracks student emotions during computer programming. It builds upon and extends previous research in this area in the following ways. First, we delve more deeply into the emotions experienced by novice programmers by tracking emotion at a fine-grained level (every 20 seconds) during a 40 minute programming session. Second, we focus on tracking emotions during students' *first* programming experience. This was accomplished by carefully screening participants to remove those with prior programming experience and those who are majoring in computer science. Third, our focus is one-on-one human-computer programming experiences without interference, distractions, or social pressures that may become factors when teachers or peers are involved in the learning process. Our emphasis was on the following three questions regarding the emotions of novice programming students: (1) which emotions are most prevalent overall and at various phases in the session, (2) how are student behaviors linked to their emotions, and (3) what is the relationship between emotion and performance?

2 Methods

2.1 Participants

Participants were undergraduate students selected from the Psychology Subject Pool at a private Midwest university in the U.S. 35 participants completed the study, but 6 were removed from consideration due to self-reported prior experience with computer programming, thereby resulting in a sample of 29 novices. We chose to eliminate students with prior experience so that the sample would be representative of novices, who may or may not eventually become programmers.

2.2 Learning Environment

The computerized learning environment consisted of four main components: an instructional area with texts and diagrams, a coding area with syntax highlighting, a hint display area, and an output console area. Participants were able to test their code via "Run" and "Stop" buttons. They used the "Submit" button to move to the next exercise, which executed their code non-interactively, using predefined inputs to determine code correctness. Participants were then given non-elaborated feedback about whether or not their submission was correct, and if correct they would automatically proceed to the next exercise. Hints were available via a "Show Hint" button. The possible score for each exercise was set to be the number of hints for that exercise plus one. Using a hint resulted in a deduction of one point from the exercise and the cumulative score was always displayed to the participants. Hints were made available on a variable time delay ranging from 45 to 90 seconds relative to the start of the exercise or the previous hint request. This delay was used so that participants would be encouraged to think about exercises instead of simply using hints to solve them quickly. Additionally, hints were only available for selected exercises as discussed below.

2.3 Procedure

Participants were individually tested in a two-hour session. The study consisted of three main phases as discussed below. A webcam built into the bezel of the monitor recorded the face of participants, while screen capture software recorded videos of the learning environment. The learning environment kept logs of the participants' interactions, including actions like key presses, button presses, and code snapshots.

Phase 1: Scaffolding Phase (25 minutes). The goal of the scaffolding phase was to provide foundational knowledge that could be applied in the fadeout phase. The scaffolding phase consisted of a set of 18 programming exercises. Each exercise had a problem statement, an explanatory text, and a set of hints. Participants needed to write working Python code to solve the problem in each exercise. Hints ranged from further instructional explanation of the key concept(s) in an exercise, code examples illustrating the concept(s), up to complete solutions for an exercise (bottom-out hint).

The exercises were predominately math-based geometry problems with numeric inputs. This topic was chosen because it is often used in introductory programming courses. Complexity and difficulty of exercises increased throughout the scaffolding phase. This was accomplished by introducing one new concept or incrementally adding to previous concepts. Explanations were precise but not exhaustive enough for participants to solve the exercises without thinking of some possible solutions, experimenting with code, or resorting to using hints when they became stuck.

One example of an exercise participants would encounter during the experiment is as follows: "Suppose you want to calculate the mileage you are getting in your car easily. Create a program to assist in this, first by prompting for *Miles driven:* and then *Gallons of gas used:* Store each of these values in a variable and print out the resulting miles per gallon." This exercise represents an incremental step from reading user input and storing it as a variable (previous exercise) to reading two different inputs into different variables (current exercise).

Participants could complete as many exercises as possible in the 20 minute time limit for the scaffolding phase before being automatically directed to the fadeout phase. On average, participants completed 16 exercises ($SD = 3.40$).

Phase 2: Fadeout (15 minutes). The fadeout phase consisted of two exercises that integrated the individual concepts covered in the scaffolding phase. The exercises in this phase were considerably more difficult compared to the scaffolding phase. No hints or explanation were available during the fadeout phase to encourage unscaffolded problem solving. The first fadeout exercise was a debugging exercise, in which participants were given code containing a variety of errors and were asked to correct the code. Five minutes were allocated for this debugging task. The second component of the fadeout phase consisted of a difficult programming exercise requiring participants to produce eleven lines of code. It also required the use of an output formatting technique that the participants were not familiar with, thereby ensuring every participant would encounter at least one logical impasse during this phase. Ten minutes were allocated for this exercise, but no student completed the exercise in that time.

Phase 3: Retrospective Affect Judgment. The retrospective affect judgment phase commenced immediately after the programming session. It involved the participant providing judgments of their emotions while viewing synchronized videos of their face and screen recorded during the session. Participants provided judgments on 13 emotions, which were mostly selected from Pekrun's taxonomy of academic emotions [15]. These included basic emotions (anger, disgust, fear, sadness, surprise, happiness), learning-centered emotions (anxiety, boredom, frustration, flow/engaged, curiosity, confusion/uncertainty) and neutral (no apparent feeling).

Emotion ratings were made at 100 points over the course of viewing the videos. The judgment points were roughly chosen to correspond with interaction events such as key presses, running of code, or displaying a new exercise. Rating points were pseudo-randomly selected with a minimum of 20 seconds between points to alleviate annoyance from making judgments in quick succession. At each rating point participants were required to select an emotion as the primary emotion they were experiencing at the time, and were also given the choice of reporting a secondary emotion.

It is important to mention three points pertaining to the affect judgment methodology. This procedure was adopted because it affords monitoring participants' affective states at multiple points, with minimal task interference, and without participants knowing that these states are being monitored while they complete the learning task. Second, this retrospective affect-judgment method has been previously validated [16], and analyses comparing these offline affect judgments with online measures including self-reports and observations by judges have produced similar distributions of emotions [17, 18]. Third, the offline affect annotations obtained via this protocol correlate with online recordings of facial activity and body movements in expected directions [19]. Although no method is without its limitations, the present method appears to be a viable approach to track emotions at a relatively fine-grained temporal resolution.

2.4 Assessing Performance

The participants' cumulative score (see above) was used as a measure of performance in the scaffolding phase. The highest possible score was 67, while the lowest possible score was a 0. Scores for the fadeout phase of the study were calculated differently because there were no hints. Instead, we considered the number of lines of code in a participant's solution that corresponded semantically to lines in a "correct" solution. The correct solution was very specific in the debugging task since participants were given code with predetermined errors. Thus, we were able to use a text processing script to remove formatting differences and determine the number of lines correctly debugged, which was used as the score (maximum of 9). For the coding portion of the fadeout phase, two trained human judges compared lines from participants' code against a correct solution to determine the score (maximum of 11). The human judges independently scored every solution and resolved any differences.

3 Results and Discussion

Which Emotions Are Most Prevalent Overall and across Different Phases? A total of 3,035 affect judgments were collected from the 29 participants. Only 589 of the judgments included a secondary affect rating, and five of the participants never reported a secondary emotion at any point. Because of the paucity of secondary emotion reports, we will not consider them any further in these results.

The analyses proceeded by computing proportion scores for each participant's primary emotion reports. The distribution of emotion proportions violated assumptions of normality, so nonparametric tests are used for all analyses. Table 1 presents mean proportions of emotion reports overall and across the two phases of the study.

The results indicated that flow/engaged, confusion/uncertainty, frustration and boredom (henceforth referred to as frequent emotions) plus neutral accounted for approximately 86% of all affect judgments, while the other eight emotions (curiosity, happiness, anxiety, surprise, anger, disgust, fear, and sadness) only accounted for 14% of the emotion reports. Moreover, Wilcoxon signed rank tests (with a Bonferroni correction of .00125 to account for multiple tests) indicated that the four frequent

emotions and neutral occured at significantly ($p < .05$ unless specified otherwise) higher rates than the eight less frequent emotions. This finding is in line with previous research suggesting that boredom, engagement/flow, confusion, and frustration are the emotions that routinely occur during learning with technology [20]. Hence, the subsequent analyses will focus on these four states as well as neutral.

We compared the emotions reported during the two phases of the study (scaffolding and fadeout). Five Wilcoxon signed rank tests, one for each emotion (plus neutral), revealed that there were significant differences for frustration and neutral. There was also a marginally significant difference for boredom. Results indicated there was more neutral reported in the scaffolding phase ($M = .187$, $SD = .187$) compared to the fadeout phase ($M = .097$, $SD = .178$), ($Z = -3.01$, $p = .003$). A different pattern was revealed for frustration in that there was less frustration reported in the scaffolding phase ($M = .109$, $SD = .085$) than the fadeout phase ($M = .184$, $SD = .152$), ($Z = -2.56$, $p = .010$). Similarly, there was less boredom reported in the scaffolding phase ($M = .104$, $SD = .131$) compared to the fadeout phase ($M = .146$, $SD = .210$), ($Z = -1.71$, $p = .088$). These findings are particularly interesting because of the differences in the two phases. The scaffolding phase gave students hints and explanations, while the fadeout phase did not provide any assistance. This might have caused more frustration in the fadeout phase since there was no easy way to resolve any difficulties encountered, though other factors such as increased problem difficulty and time within the session may also be influential here.

Table 1. Proportion of emotions made in retrospective affect judgment

Emotion	Overall	Scaffolding	Fadeout
Flow/Engaged	.231	.233	.229
Confusion/Uncertainty	.217	.207	.235
Frustration	.139	.109	.184
Boredom	.118	.104	.147
Curiosity	.059	.073	.034
Happiness	.030	.042	.011
Anxiety	.022	.013	.038
Surprise	.014	.019	.004
Anger	.009	.004	.018
Disgust	.006	.008	.003
Fear	.000	.001	.000
Sadness	.000	.001	.000
Neutral	.153	.187	.097

How Are Student Behaviors Linked to Their Emotions? To investigate this question, we grouped the different student behaviors into three broad categories: *idling*, *constructing,* and *running*. When participants were entering code into the learning environment interface, they were *constructing*. When executing code either via a Run

or Submit interaction event, they were *running* code, and they were *idling* when otherwise not interacting with the interface.

We computed proportional scores for each emotion and neutral with respect to each of these three behaviors (see Table 2 for mean proportions of emotions for these behaviors). We then computed five separate Friedman tests for each emotion and neutral in order to test for differences in emotions based on the three types of student behavior. Tests for differences in flow/engagement, frustration, and boredom were significant, $p < .01$. There was also a trend in differences for confusion, $\chi 2(2, N = 29) = 4.42$, $p = .110$. Post-hoc comparisons in the form of Wilcoxon signed rank tests with a Bonferroni adjustment ($\alpha = .016$) were conducted in order to further probe these differences. The results indicated that flow/engagement was reported at higher rates when students were *constructing*, followed by *running*, and *idling* (constructing > running > idling). There was significantly more boredom when students were *idling* compared to *running* (idling > running). Frustration was greater when students were *running* compared to when students were *constructing* or *idling*, which were statistically equivalent (running > constructing = idling). Finally, confusion was greater when students were *idling* compared to *constructing*, while both were similar to *running* (idling > constructing).

These patterns were quite revealing about the types of emotions that occurred based on the behavior exhibited. Students experienced more engagement but also frustration when they were engaging in behaviors that require some activity (e.g., running and constructing). Idling might be indicative of two different emotions, namely boredom or confusion. On one hand, students might stop interacting to idle because they are disengaged. On the other, idling might indicate confusion that requires some processing before moving forward. A finer-grained analysis of behavior is needed to resolve these two alternatives.

Table 2. Means and standard deviations (in parentheses) for the proportion of emotions and neutral for each type of student behavior

Emotion	Constructing	Idling	Running
Boredom	.117 (.162)	.156 (.162)	.087 (.135)
Confusion	.176 (.127)	.236 (.134)	.241 (.144)
Flow/Engaged	.303 (.245)	.220 (.181)	.151 (.145)
Frustration	.119 (.100)	.124 (.110)	.193 (.124)
Neutral	.189 (.214)	.150 (.165)	.126 (.138)

What Is the Relationship between Emotion and Performance? On average, students scored 52.1 ($SD = 4.24$) out of the maximum scaffolding score of 67 (77.6%). Scores were considerably lower for the more difficult fadeout debugging ($M = 4.24$, $SD = 2.64$; 47.1% out of maximum 9), and fadeout coding ($M = 5.66$, $SD = 3.64$; 51.5% out of a maximum of 11) portions of the study. We correlated these scores with the proportion of emotions reported at corresponding portions of the study and the resultant Spearman correlation matrix is presented in Table 3. It should be noted that although we tested the significance of the correlational coefficients, our small sample size of 29 participants

does not yield sufficient statistical power to detect small (*rho* ≈ .1) and medium sized effects (*rho* ≈ .3). Hence, in addition to discussing significant effects we also consider non-significant correlations of .2 or higher to be meaningful because these might be significant with a larger sample.

Table 3. Correlations between emotions and performance

Emotion	Scaffolding	Fadeout: Debugging	Fadeout: Programming
Boredom	.239	*-.341	**-.459
Flow/Engaged	-.061	.254	**.512
Confusion/Uncertainty	**-.407	-.001	-.207
Frustration	-.031	.041	-.026
Neutral	.188	-.087	-.036

Note. $**p < .05$; $*p < .10$.

The results were illuminating in a number of respects. Consistent with our expectations, boredom was negatively correlated with performance during both parts of the fadeout phase. However, boredom was positively correlated with performance during the scaffolding phase, which was contrary to our expectations. This might be attributed to students finding the exercises in the scaffolding phase to be less challenging, presumably due to the presence of hints and explanations. Flow/engagement was not correlated with performance during the scaffolding phase, but was a positive predictor of performance in both the debugging and programming parts of the fadeout phase, which is what we might expect.

Confusion/uncertainty had a large negative effect on performance during the scaffolding phase, suggesting that much of the confusion went unresolved. Confusion was not correlated with performance in the debugging portion of the fadeout phase, but had a smaller negative correlation with performance during the programming part of the fadeout phase. Finally, we were surprised to discover that frustration was not correlated with performance during both the scaffolding and fadeout phases, a finding (or lack thereof) that warrants further analysis.

4 General Discussion

We performed a fine-grained analysis of the emotional states of novice computer programming students with an eye for applying any insights gleaned towards the development of computerized interventions that respond to emotion in addition to cognition. We found that flow/engaged, confusion, frustration and boredom represented the majority of emotion self-reports, thereby suggesting that an affect-sensitive intervention should focus on these states. We also found that the emotions varied as a function of instructional scaffolds and were systematically linked to different student behaviors (idling, constructing code, running code), a finding which would pave the way for developing automated interactional- and contextual-based methods to track these emotions. Finally, our results revealed that the emotions were not merely incidental to

the learning process; they also correlated with performance in expected and surprising ways. In general, but noting exceptions discussed above, performance was negatively correlated with boredom and confusion, positively correlated with flow/engaged and not correlated with frustration.

There are some limitations with the present study that need to be addressed in the future. First, self-reports are biased by the honesty of the participants, so future studies should combine additional method in addition to or in lieu of self-reports. Possible methods include trained observers, physiological sensors, and peers that may be able to pick up on more nuanced indicators of affective states. Second, the sample size was also quite small, which limited the statistical power required to detect smaller effects. Third, the participants were sampled from a single university, which might not be reflective of the body of novice computer programmers as a whole. Fourth, the course-grained nature of some of the logs made it difficult to disambiguate when students read explanations from other idling activities. This can be resolved by redesigning the interface or by using an eye tracker to determine what part of the interface students are focusing on while not interacting.

Future work will focus on collecting additional data to alleviate the limitations discussed above. We will also use log data (e.g. keystrokes, syntax errors, hint usage) and video recordings to build models that detect novice programmer emotions, using established computer vision and machine learning techniques [21]. The long-term goal is to use these detectors to trigger automated interventions that are informed by affect. It is our hope that an affect-sensitive learning environment for novice computer programmers equipped with intelligent handling of emotions might contribute to a more technical workforce to handle the demands of the age of Big Data.

Acknowledgment. This research was supported by the National Science Foundation (NSF) (ITR 0325428, HCC 0834847, DRL 1235958). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

References

1. Haungs, M., Clark, C., Clements, J., Janzen, D.: Improving first-year success and retention through interest-based CS0 courses. In: Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, pp. 589–594. ACM, New York (2012)
2. Alspaugh, C.A.: Identification of Some Components of Computer Programming Aptitude. Journal for Research in Mathematics Education 3, 89–98 (1972)
3. Blignaut, P., Naude, A.: The influence of temperament style on a student's choice of and performance in a computer programming course. Computers in Human Behavior 24, 1010–1020 (2008)
4. Law, K.M.Y., Lee, V.C.S., Yu, Y.T.: Learning motivation in e-learning facilitated computer programming courses. Computers & Education 55, 218–228 (2010)
5. Shute, V.J., Kyllonen, P.C.: Modeling Individual Differences in Programming Skill Acquisition. Technical report no. AFHRL-TP-90-76, Air Force Human Resources Laboratory, Brooks AFB, TX (1990)

6. Csikszentmihalyi, M.: Flow: The psychology of optimal experience. Harper and Row, New York (1990)
7. Lee, D.M.C., Rodrigo, M.M.T., Baker, R.S.J.d., Sugay, J.O., Coronel, A.: Exploring the Relationship between Novice Programmer Confusion and Achievement. In: D'Mello, S., Graesser, A., Schuller, B., Martin, J.-C. (eds.) ACII 2011, Part I. LNCS, vol. 6974, pp. 175–184. Springer, Heidelberg (2011)
8. Burleson, W., Picard, R.W.: Affective agents: Sustaining motivation to learn through failure and a state of stuck. In: Social and Emotional Intelligence in Learning Environments Workshop In Conjunction with the 7th International Conference on Intelligent Tutoring Systems, Maceio-Alagoas, Brasil (2004)
9. D'Mello, S., Graesser, A.: Dynamics of affective states during complex learning. Learning and Instruction 22, 145–157 (2012)
10. Larson, R.W., Richards, M.H.: Boredom in the middle school years: Blaming schools versus blaming students. American Journal of Education 99, 418–443 (1991)
11. Khan, I.A., Hierons, R.M., Brinkman, W.P.: Mood independent programming. In: Proceedings of the 14th European Conference on Cognitive Ergonomics: Invent! Explore!, London, United Kingdom, pp. 28–31 (2007)
12. Rodrigo, M.M.T., Baker, R.S.J.d.: Coarse-grained detection of student frustration in an introductory programming course. In: Proceedings of the Fifth International Workshop on Computing Education Research, pp. 75–80. ACM, New York (2009)
13. Rodrigo, M.M.T., Baker, R.S.J.d., Jadud, M.C., Amarra, A.C.M., Dy, T., Espejo-Lahoz, M.B.V., Lim, S.A.L., Pascua, S.A.M.S., Sugay, J.O., Tabanao, E.S.: Affective and behavioral predictors of novice programmer achievement. SIGCSE Bulletin 41, 156–160 (2009)
14. Grafsgaard, J.F., Fulton, R.M., Boyer, K.E., Wiebe, E.N., Lester, J.C.: Multimodal analysis of the implicit affective channel in computer-mediated textual communication. In: Proceedings of the 14th ACM International Conference on Multimodal Interaction, pp. 145–152. ACM, New York (2012)
15. Pekrun, R., Stephens, E.J.: Academic emotions. In: Harris, K.R., Graham, S., Urdan, T., Graham, S., Royer, J.M., Zeidner, M. (eds.) APA Educational Psychology Handbook. Individual differences and cultural and contextual factors, vol. 2, pp. 3–31. American Psychological Association, Washington, DC (2012)
16. Rosenberg, E.L., Ekman, P.: Coherence between expressive and experiential systems in emotion. Cognition & Emotion 8, 201–229 (1994)
17. Craig, S., D'Mello, S., Witherspoon, A., Graesser, A.: Emote aloud during learning with AutoTutor: Applying the Facial Action Coding System to cognitive–affective states during learning. Cognition & Emotion 22, 777–788 (2008)
18. Craig, S., Graesser, A., Sullins, J., Gholson, B.: Affect and learning: An exploratory look into the role of affect in learning with AutoTutor. Journal of Educational Media 29, 241–250 (2004)
19. D'Mello, S.K., Graesser, A.: Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. User Modeling and User-Adapted Interaction 20, 147–187 (2010)
20. D'Mello, S.: A selective meta-analysis on the relative incidence of discrete affective states during learning with technology (in review)
21. Blikstein, P.: Using learning analytics to assess students' behavior in open-ended programming tasks. In: Proceedings of the 1st International Conference on Learning Analytics and Knowledge, pp. 110–116. ACM, New York (2011)