Response to Reviewers

We appreciate the informed feedback from the reviewers. We have grouped together the feedback from the reviewers when they seem to be addressing the same issue. We have then explained for each group the changes we made in the paper. We feel that the reviewers comments have helped us to significantly clarify and strengthen the presentation of work.

Changes

(1) Reviewer Comments

"*Reviewer 3: Define the overall theme of the evaluation. What are the research questions??*
"*Reviewer 3: 3. Your title emphasizes the evaluation as a central component of the paper. If this is the case, I expect the evaluation to be better structured around core research questions or hypotheses. Making clear the core contributions of BlockPy you are trying to evaluate in the introduction would also help (as in my first suggestion). As is, the evaluation section feels a bit like combination of separate results without a central theme."*
"Reviewer 3: 5: Additionally, the evaluation section would be stronger if motivated by clear research questions or hypotheses. Even if you analysis wasn't that structured to begin with, organizing the findings to be a bit more motivated would be helpful to the reader."
"Reviewer 1: Too much of the paper is spend describing and justifying the design and tool little is spent on its evaluation."
"Reviewer 1: From the description of the study and the data collection, it sounds like there is a wealth of data that can be explored to show has various features designed into the environment were successful and places where novices still struggled. While this exists in some places, a more systematic and comprehensive evaluation would go a long way to making this a very strong paper, resulting in a greater contribution to the literature. Further, more carefully linking the analysis to the design features will make the paper more useful as it will help other make more informed design decisions. Currently, much of the evaluation talks about students transition from PyBlock to Spyder, while that is interesting, that is not what the paper is actually about, instead focusing on what happened while learners used PyBlock. A good example is the data you present about student using (or not using) the pseudocode feature of the environment. More analysis making those links would strengthen the paper."
"Reviewer 2: First, the analysis could be much stronger. The methodolgy was not clearly laid out and the results only appeared where they supported the narrative. The survey questions should be enumerated and results for each survey question should be shown. The questions should be tied to specific research questions and should be discussed more formally in terms of those research questions. It isn't necessary to do hypothesis testing but there should be a more clear vision for the questions and the analysis. "

(1) Revision

A major recurring feedback from the reviewers was the that the evaluation tended to be scattered in some places. Therefore, a number of substantial changes were made to better organize the work. Section 1 (Contributions) was edited to to highlight major contributions of the work, explicit research questions are identified, and better explain the major story. Section 5 was heavily edited to better reflect the core contributions of the paper and answer the research questions - each subsection of section 5 addresses a different research question.

(2) Reviewer Comments

"Reviewer 2: but before accepting it the authors need to be much more clear about the audience (novices, non-major students, continuing education outside of formal educational context, etc)"
"Reviewer 3: 4. I'm not sure who the target audience of this paper is..."

(2) Revision

Multiple reviewers mentioned that it was unclear who the audience was, both for the paper and BlockPy. The introduction was edited to make the audience of both more clear - we are aiming to speak to both CS educators and developers, and BlockPy is developed for all introductory computing students, and in particular non-computing majors.

(3) Reviewer Comments

"2.1: Authenticity seems to be a core motivation for BlockPy, and something you repeatedly claim separates it from existing environments. I think you would do well to define it directly in this section using education literature. That way the reader can judge whether you meet this goal. You seem to sometime use the word to mean "useful" or "practical" (e.g. Data Science is useful in any career path), but I think that's not quite the same as being perceived as authentic."
"Reviewer 2: Secondly, there is a disconnect between the stated motivation and the actual solution. Guzdial work suggests that coding needs to be not only relevant and creative, but that it also needs to be perceived as useful and authentic. Guzdial et. al also highlight the importance of longer-term goals. In my reading, it is not clear how PyBlocks accomplishes any of these goals. By providing scaffolding the experience is less authentic, by having all students use datasets that are not designed for their specific use-cases it is not personally relevant, and as the authors suggest the context (data-science) is in line with the goals for all students. This context (data science) may even be daunting for students who are less inclined toward math and science and have the opposite effect of the intended goal; shutting them off the the CS discipline. This is the opposite goal of the recent broadening computing initiative. "
"Reviewer 2: about the evaluation (what are the research questions and what does ALL your data say about them), and about how the context might be appropriate for the audience (from my reading, it appears that all students regardless of their major use the same datasets and it is hard to believe those datasets are equally relevant for all students). I think having different majors use different data sources might be one potential way to clear up the context issue."

(3) Revision

Section 2 was heavily rewritten to better explain our argument for authenticity within BlockPy. The environment itself is not meant to be authentic, but the primary context that it supports (Data Science) provides a sense of authenticity and usefulness to the learners. The environment provides numerous scaffolds in specific support of this context.

Subsection 2.3 was moved to become Subsection 2.1 to explain the scaffolds first, and then 2.2 and 2.3 (formerly 2.1) were combined to become the new 2.2. This change better reflects the order of contributions of the environment: first the pedagogical scaffolds of the environment, then the data science context. Although both are major contributions, the data science context has been previously heavily explored in several cited prior works by the authors; we did not want to make this work redundant to those, and so seek to reduce the space spent evaluating the data science context. These works have been included as citations in the final paragraph of Subsection 2.2 along with some descriptions.

(4) Reviewer Comments
"Reviewer 1: In your discussion of Situated Learning should cite Lave & Wagner who created the theory"
"Reviewer 3: 2.1: I appreciate your references to theories of motivation (e.g. the MUSIC Model), but it is not clear if all of them directly influenced your design of BlockPy and how."

(4) Revision
In the new Subsection 2.2, the justification for the inspiration from the MUSIC model and Situated Learning Theory was improved, and a new citation was included for the relevant Lave and Wenger paper.

(5) Reviewer Comments
"Reviewer 3: 2.3: "not only are learners more likely to be successful when working with Python" - Compared to what language? This seems like a strong claim that should have a reference."

(5) Revision
The implicit comparison to other languages was dropped from 2.3, instead relying on the citation for the previous sentence to back up the claim that students using Python are likely to succeed.

(6) Reviewer Comments
"Reviewer 3: 2.3: "learners can negatively perceive block interfaces as being childish or unsuitable for professionals" - I would recommend citing:
[1] D. Weintrop and U. Wilensky, “To Block or not to Block, That is the Question: Students’ Perceptions of Blocks-based Programming,” in Proceedings of the International Conference on Interaction Design and Children, 2015, pp. 199–208."

(**6) Revision**
We have incorporated the relevant citation into the work; this was a very interesting paper, and we appreciate how it connects to our work.

(7) Reviewer Comments
"While the paper is well-written, sometimes it can come across as more opinionated than objective. For example, claims about Scratch bearing no resemblance to intro programming languages or Greenfoot having an inauthentic context could easily be debated (and I imagine

their creators would). It can come off as if you are trying to minimize these environments (maybe not actually your aim), which is not necessary to motivate the need for BlockPy - they can both play different, important roles.

2.3: "Scratch/Snap! language, which bears little-to-no relation to the languages seen in introductory courses" - A bit of a nit here, but many introductory course _do_ use block-based languages. Consider rewording."

(7) Revision
The language in the new Subsection 2.1 has been changed to be less aggressive towards other environments. Arguments against contexts authenticity no longer focus on Greenfoot or PencilCode; these arguments have been moved to Subsection 2.2 and discussed with more emphasis on the relationship of authenticity with communities of practice to better highlight our point.

(8) Reviewer Comments
"Reviewer 3: 4.3: This section could be cut/combined if you're short on space. I recognize these are important design choices, but the number of readers who will need to how to solve this exact problem is small."
(8) Revision
4.3 was removed based on reviewer 3's suggestion.

(9) Reviewer Comments
"Reviewer 3: 4.4: I know Kelly Rivers has spent a lot of time grappling with this exact issue (in Python) and has some sophisticated techniques for parsing/repairing invalid Python code. She talks about a bit in:
[2] K. Rivers and K. R. Koedinger, “Data-Driven Hint Generation in Vast Solution Spaces: a Self-Improving Python Programming Tutor,” Int. J. Artif. Intell. Educ., vol. 16, no. 1, 2015."
(9) Revision
A reference to and some explanation of Rivers' paper is now included in 4.3 (formerly 4.4) to highlight the connection to existing work.

(10) Reviewer Comments
"Reviewer 1: I would revisit the title, I’m not sure “Open-Access” is the most salient feature of PyBlcoks"
(10) Revision
The title was changed by recommendation of Reviewer 1, to eliminate the phrase "open-access". In further consideration, we changed the title to highlight the salient aspects of the work (blocks and data science context integration).

(11) Reviewer Comments
"Reviewer 3: 2.1: I appreciate your references to theories of motivation (e.g. the MUSIC Model), but it is not clear if all of them directly influenced your design of BlockPy and how."

"Reviewer 3: 1. It seems to me that you emphasize 3 key features of BlockPy in the introduction: Block/Text Interface, Authenticity and Scaffolding/Feedback. You do a good job of explaining the first element, but I think the other two are not fully covered. You contrast the authenticity of Data Science with the games/art focus of other environments. However, you never define "authenticity" as you see it, justify why BlockPy has more of it, or demonstrate this to be the case in your evaluation. I would appreciate a little more literature on the subject and more structuring of the paper around this idea. As for feedback, your explanation of how BlockPy's feedback works is quite short, and you say yourself in the evaluation that it is inadequate. Additionally, there is a wealth of literature on the subject, which is largely omitted here. Perhaps emphasize scaffolding more as future work than as a key feature of BlockPy that sets it apart from the rest. (Or give it more space and focus more of the evaluation on it.)"

(11) Revision
Considerable time and attention was spent in rewriting section 2 to more clearly explain our use of the word "authenticity", with respect to the learning context. We also attempted to more clearly discuss how BlockPy was influenced by theories of motivation and situated learning theory.

(12) Reviewer Comments
Reviewer 3: 3, Guided Feedback: The "Guided Feedback" section seems far too short and vague. What kinds of feedback does it provide. How? When? Based on what theory? I know this is an overview paper, and I'm sure you have more detailed work elsewhere or in the works, but scaffolding seems to be a key feature and I was excited to read more about it.
(12) Revision
The Guided Feedback section was improved with the addition of 1) an example of a specific kind of feedback that could be added, and 2) a list of common types of feedback that could be given within the environment.

(13) Reviewer Comments
"Reviewer 3: 3: If you need to cut something, I think some of the technical details are not as critical here, but that might be my personal bias towards the education side of things."
"Reviewer 3: 3: Again, I acknowledge that this is an overview paper, but most sections here feel underdeveloped. One option would be to cut some (mentioning them in passing at the end) and flesh out others. My suggestions for cutting would be "Parsons Problems", "Dangling Blocks" and "Free and Open Source." These could easily be summarized quickly."
"Reviewer 1: Likewise, the extended discussion of features of the environment can be trimmed as some are more novel and new than others – focus on the novel stuff and spend less time on features that have already been implemented and discussed by other environments."
(13) Revision
Under suggestion by reviewer 1 and 3, a number of the more esoteric technical details in chapter 3 were deleted - in particular, the sections on "Parsons Problems", "Dangling Blocks" and "Free and Open Source" were removed.

(14) Reviewer Comments

"Reviewer 3: Fig 5: There are 5 colors. Does this mean the students only used BlockPy for 5 days before transitioning? Is that normal? Other curricula (e.g. Dann et al.'s Alice curriculum) take much longer. This is worth addressing directly."

(14) Revision

In the second paragraph of Section 5, we added explanation about the length of time that BlockPy was used in our curriculum, and cited contrasting experience.

(15) Reviewer Comments

"Reviewer 3: Fig 5: Completion rates are quite poor on the last day. Why? This definitely needs to be addressed."

(15) Revision

A sentence was added in the second paragraph of section 5.2 to better explain the lower completion rates on the last day.

(16) Reviewer Comments

"Reviewer 3: 5: I would have appreciated a paragraph on the methodology you used when parsing survey responses. You sprinkle findings throughout the next sections, but I'm curious how you can to them."

"Reviewer 3: 5: I don't see much mention of the survey questions after 5.1 and 5.2. Were there any non-free-response questions, and if so why don't you report them?"

(16) Revision

Section 5's introduction was extended to more clearly explain the survey and qualitative coding methodology, and the specific text of the survey questions was given. A new table (Table 2) was added and explained to show the specific results of the qualitative analysis of the survey.

(17) Reviewer Comments

"Reviewer 3: 5.4: I would caution against using edit distance (string or tree) as a measure of productivity, especially in a block language. These languages lend themselves to a tinkering approach, in which students make many edits without necessarily being productive. Picking up large groups of blocks and moving them also artificially inflates a metric like that. I would suggest cutting this section or reworking the angle of analysis. Even if you find a better measure of productivity, the work students do in block and text modes likely do not represent equivalent, comparable tasks. So comparing the productivity in each isn't fair. For more on how students use the two modes differently, see:"

[3] D. Weintrop and N. Holbert, “From Blocks to Text and Back : Programming Patterns in a Dual-modality Environment,” in Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education - SIGCSE ’17, 2017, pp. 633–638.

"Reviewer 1: Some of the figure are unclear and need more description in the text with “edit distance” in figure 7 being a good example – what is the distance being measured between? Consecutive runs? Time intervals?"

(17) Revision

Reviewer 3 made an excellent point against using edit distance, which upon further reflection we agree with this critique. We have removed the figure and paragraph comparing the relative edit distances. We hope to revisit this data in a future work with a more nuanced comparison of the affordances of text and block-based editing.

(18) Reviewer Comments
"Reviewer 3: 5.6 [now section 5.1]*: This section might benefit from some discussion of how the transition from blocks to text is mediated in instruction. See:*
[4] W. Dann, D. Cosgrove, and D. Slater, "Mediated Transfer: Alice 3 to Java," in Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, 2012, pp. 141–146."
(18) Revision
Further explanation was put in 5.1 for how the mediation was given in the instruction. The text was slightly reorganized to better present our observations regarding the transition.

(19) Reviewer Comments
"Reviewer 3: In various places (e.g. table 2 [now Table 3]*) you include raw counts, without knowing the total number of occurrences these numbers aren't useful, consider reporting percentages there and throughout."*
(19) Revision
Table 3 (formerly table 2) was changed to report percentages instead of raw values. In addition, the table and the accompanying explanation in the text indicate the total number of programs analyzed.

(20) Reviewer Comments
"Reviewer 3: Fig 9 [now Fig 5]*: I think it might be more interesting to label each student on each assignment with one of: [0-25%, 25-50%, 50-75%, 75-100%] of time using the block interface. Then use a stacked bar chart to show the breakdown of students for each assignment. That would allow us to visually see your "deeper analysis" point that it was multiple students using text."*
"Reviewer 1: Another good example is Figure 9 [now Fig 5] *– the data used to create this figure could probably be its own paper, the way it is presented on just scratches the surface of what is happening and what is interesting about PyBlock."*
"Reviewer 2: Some results such as Figure 9 [now Fig 5] *were potentially interesting but they weren't discussed or put into context by triangulating them with survey or interview results that support the observations."*
(20) Revision
Figure 5 (formerly figure 9) was heavily revised based on the suggestions of the reviewers. Instead of only reporting the percentage of time spent in each mode (block vs. text), the image now shows the percentage of students that spent given proportions of time (in groups of <25%, 25-50%, 50-75%, and >75%) in Block mode. Although somewhat denser in terms of information, we believe that this graph tells a more interesting story. This should also help with the concerns about the depth of the analysis of the data.

(21) Reviewer Comments
"Reviewer 1: Likewise, there is a current lack of specificity that undermines that good work that was done. For example, the manuscript states that "most students were able to answer most questions" – what does this tell us? If you include the actual numbers, we might learn something about student successes/difficulties."
(21) Revision
Using the reviewer's comment, we improved the presentation of the student completion rates to give the percent of students who completed problems at the 90% and 60% levels and better clarify the observation.

(22) Reviewer Comments
"Reviewer 3: 6.4: "visual programming" - I'd recommend "block(s)-based programming" over this term, since "visual programming" has a lot of baggage associated with older languages like LabView, which doesn't really pertain to modern block-based languages."
(22) Revision
Reviewer 3's suggestion about the term visual programming was followed to replace the term "visual programming" with the more accurate "block-based programming".

(23) Reviewer Comments
"6: A major omission in this section is future work for evaluation. From a CS Ed perspective (which is the first keyword you list!), this is the primary work you should be concerned with. If you continue to develop the system without rigorous evaluation, you may end up sinking a lot of work into something that only works well in theory. You embed a number of hypotheses in the justification of BlockPy, and I would like to see a good bit of the future work address validating those hypotheses (Dual Block/Text interfaces are (more) helpful, Data Science is a (more) authentic context, BlockPy Feedback helps students perform/learn better, a BlockPy curriculum improves learning compared to a standard Python curriculum)."
(23) Revision
A new subsection (6.1) was added to describe an ongoing experimental study to better evaluate BlockPy features.

(24) Reviewer Comments
"Reviewer 3: 6: If you need room for the above, I think reducing the number of technical additions mentioned here could be helpful. At the least, they could be combined into a "New Features" section.
"Reviewer 1: Future directions for PyBlock are interesting, but a single paragraph or two listing future directions should be sufficient."
(24) Revision
We have shortened each of the sections in chapter 6 to conserve space and reduce emphasis on the future work.

(25) Reviewer Comments

"Reviewer 3: 6.1: You claim to be guided by Rivers' work, but from your description of your guidance system and proposed future directions, I don't see much resemblance. It seems like your focus is on improving the authoring interface, rather than offloading that burden using data-driven hint generation as Rivers does. It sounds like you'd like to contrast your approach with hers in the last sentence of the paragraph (increased emphasis on the role of the instructor). I think would be productive - why do you believe this approach is more fruitful for your situation. What about your improved hint-designer interface will ensure that instructor generated feedback scales well?"

(25) Revision
In the interests of space, we have removed the citation here to Kelly's work (already cited elsewhere in the paper) and Singh's to simplify the conversation about the new instructor feedback interface. Our currently envisioned interface does not use a data-driven approach, so we do not want to draw too big a comparison between these projects and our own.

(26) Reviewer Comments
"6.1: On that note, I'd recommend Price et al.'s work on supporting novices _specifically_ in novice programming environments, which seems quite relevant to this section.
[5] T. W. Price, Y. Dong, and D. Lipovac, "iSnap: Towards Intelligent Tutoring in Novice Programming Environments," in Proceedings of the ACM Technical Symposium on Computer Science Education, 2017.
(26) Revision
We have incorporated this reference into section 6.2 (formerly 6.1) to better refer to existing work in this space by Price et al.

(27) Reviewer Comments
6.4: "preliminary results gathered in our research suggest the appeal of Data Science over other contexts" - I missed this in your results, and I think it is an important argument to make if you have the data to do so.
(27) Revision
The sentence in question was clarified to refer to the research conducted and published in a different study.

Once again, we thank you for your very helpful feedback. We hope that this revision is a satisfactory response to your suggestions.

Sincerely,
Bart, Tibau, Kafura, Shaffer, and Tilevich