

“Enabling Writers” Prize Competition

Summary

All Children Reading: A Grand Challenge for Development (ACR GCD) has launched “Enabling Writers,” a \$100,000 global prize competition to incentivize the development of software solutions that allow writers to easily create and export decodable and leveled fiction and non-fiction readers in mother tongue languages to help early primary students (ie. grades one to three) learn to read.

To solve this important challenge, ACR GCD aims to attract individuals, companies and communities from all fields, but especially the software development, digital literacy and education communities. Applicants were invited to submit their software applications until October 1, 2014. Following formal evaluation, three finalists received feedback and awards of \$12,000 each. Following piloting and reviewed in three countries with ACR GCD partner programs, the highest performing software will win a grand prize of \$100,000.

Background

One of the main barriers to improving children’s reading skills is the lack of appropriate and engaging reading materials in mother tongue languages, meaning children are unable to get the reading practice they need in the language they speak and understand.

The Enabling Writers prize competition seeks to drive the creation of new software technologies—and the improvement of existing programs—that make it easier, more cost-effective and efficient to write high-quality early grade reading materials that follow tested reading instruction methodologies.

The prize competition is powered by InnoCentive, a global leader in crowdsourcing innovation problems.

Challenge

‘Enabling Writers’ aims to spur the development of software that would allow writers to use an easy step-by-step process, on a computer or mobile device, to create texts that follow tested early-grade reading instruction methodologies. The software would:

- Work for writers who know a story they want to write or a subject matter they want to present, but also provide less prepared writers with existing stories and nonfiction text that they could adapt for their audiences;
- Support the creation of both decodable and levelled readers
- Ensure writers are kept within technical boundaries appropriate for the target reader and reading level; and
- Provide directions and prompts in a common national language, but allow authors to write in both national and local languages.

The winning software will be freely available and released to the public for open source development.

Awards

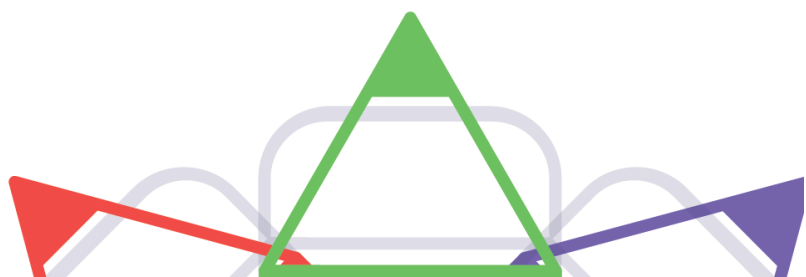
Not only would entrants contribute to improving literacy in the developing world, the three finalists received:

- A \$12,000 cash prize
- Invitations to exclusive events
- Global recognition from ACR GCD
- Expert feedback on their software from child development and digital education specialists
- The chance to win \$100,000

Judging Criteria (first pass)

The first pass will look at the following mandatory, objectively-assessable criteria.

Criteria	Score
Based on the reading level of their intended audience, can users create templates that control for letters and sounds used in the text, as well as font type, font size, word and line spacing, number of words on a page, and sentence length of the readers?	[Yes/No]
Can writers create templates with pre-established parameters for : (a) decodable readers, and (b) leveled readers?	[Yes/No]
Is the software Unicode-compliant and allow for the creation of readers in Latin, Arabic and Devanagari scripts, including diacritical, tone and accent marks as well as the characters from the International Phonetic Alphabet needed some languages?	[Yes/No]
Can the software produce custom letter and word lists from inputted texts or lists for writers' use?	[Yes/No]
Can users create readers both from scratch and by adapting pre-loaded sample texts?	[Yes/No]
At a minimum, can users add .jpeg and .png files to their readers?	[Yes/No]
Are prepress and print-ready readers able to be outputted to PDF and a common editable format in which layout is preserved when opened through free-to-use software (e.g. HTML); but if electronic readers, are they able to be outputted to at least .epub format?	[Yes/No]

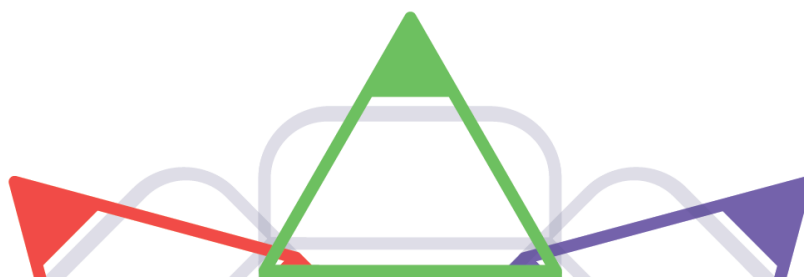


Judging Criteria (second pass)

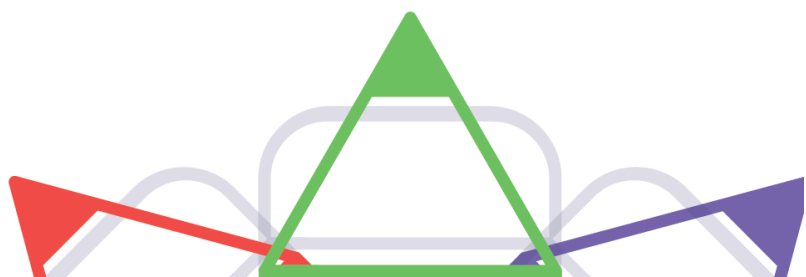
Entries that meet the criteria in the first pass will be taken forward to the second pass of judging. The second pass will focus on the subjective and/or optional criteria detailed below. Judges will rate the following criterion on a scale of 1-5, with 5 being excellent and 1 being poor. This is then multiplied by its respective weighting and at the end, the scores from all the judges are averaged and normalised.

2

Criteria	Description	Weighting
Customization options and flexibility	<ul style="list-style-type: none"> To what extent are writers able to customize their readers in regards to appearance and functionality? 	1
Availability on a wide range of devices	<ul style="list-style-type: none"> A score of 2 indicates both Windows and Mac or Linux software or both Android and iOS or Windows Mobile applications A point will be added for each additional format available up to a maximum of 3 (Windows, Mac, Linux, Android, iOS, Windows Mobile or web application) 	1
Multilingual support	<ul style="list-style-type: none"> How extensive is the selection of scripts and range of languages for prompts, menus and sample texts within the software? 	2
Accessibility	<ul style="list-style-type: none"> How extensive are the provisions for writers and children with disabilities? 	1
Ease of use	<ul style="list-style-type: none"> How helpful and appropriate are the guidance and prompts offered to the user? How user-friendly and intuitive is the interface? 	3

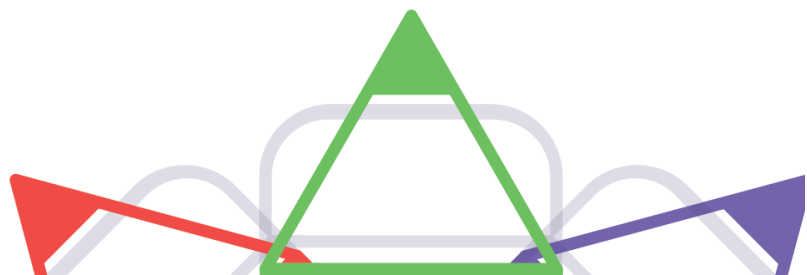


Functionality	<ul style="list-style-type: none"> • In how many languages are pre-established decodable reading levels offered to the user? • How appropriate are the decodable reading levels that have been set? • How appropriate are the resulting readers for early primary school children? • What is the extent and effectiveness of assistance offered for children when they're faced with more advanced/new words? (Users should be able to add new words and an accompanying definition to the dictionary) 	4
Reliability	<ul style="list-style-type: none"> • How reliable is the software? Were there any issues installing or running the software? (E.g. any bugs, software repeatedly crashing etc.) 	2
Aesthetics	<ul style="list-style-type: none"> • How aesthetically pleasing is the software's user interface? • How aesthetically pleasing are the resulting readers? 	1
Data sharing functionalities	<ul style="list-style-type: none"> • What is the quality of the actual or proposed procedures for capturing, collating, and making publically available useful data on how the software is used and its effectiveness in fulfilling its purpose? <p>(2 points will be awarded for the presence of the procedures and up to 3 for quality, and if hypothetical, the feasibility)</p>	1
Clear and concise accompanying manual and support	<ul style="list-style-type: none"> • How clear is the manual? Is it concise yet comprehensive? • Is it written at no more than a sixth grade level? • What is the extent and quality of troubleshooting provided for common issues? • Would the average user be able to learn to use the software with no more than 20 hours of training? 	3



Coding quality¹	<ul style="list-style-type: none">• How conducive to open source development is it?<ul style="list-style-type: none">○ How readable is the code?○ How well documented is the code?○ To what extent does it follow best practices of the community?	3
-----------------------------------	--	---

¹ Only the top ten entries will be assessed against this criterion.



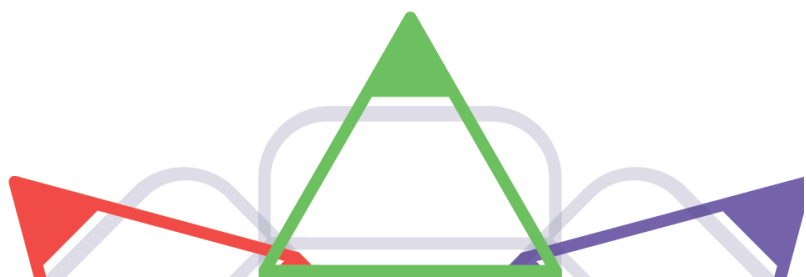
In general, we are looking for software that enables writers to more easily create readers that are designed for students learning to read. The readers should be appropriate, interesting, and provide the optimum reading and learning experience for early primary school children (i.e. grades one to three). Our target audience for the software are writers and ultimately, when readers are printed, early primary school children in developing countries.

The specific software specifications required include:

1

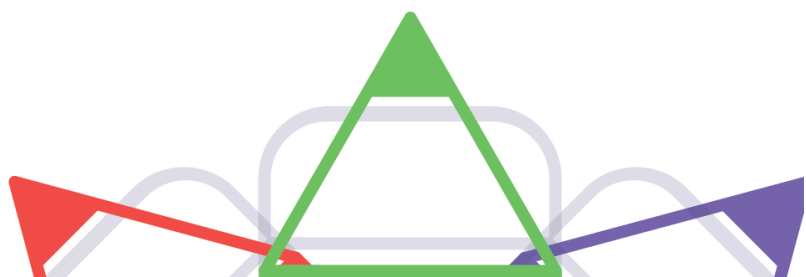
1. **A step-by-step approach to create decodable and leveled readers, including menus, prompts and guidance for the user.**
 - a. At a minimum, menus and prompts should be provided in English and **one** African, Asian or Latin American language and be modifiable to any language using a Latin, Arabic or Devanagari script.
2. **Options to create readers both from scratch and by adapting pre-loaded sample texts.**
3. **Operating on low technical requirements.**
 - a. Windows and Mac applications must have offline functionality (i.e. the software engine is not running remotely and accessed only via the web or a virtual machine). In these instances the software should be able to run on computers or laptops with a maximum of:
 - i. 4GB RAM
 - ii. 2 GB available hard drive space
 - iii. 1.5 Ghz processor
4. **Availability on a wide range of devices.**
 - a. **Both** Windows **plus** either Mac **or** Linux; or
 - b. **Both** an Android **plus** either iOS **or** Windows Mobile application.
 - c. Availability as a web application (**desired, but non-essential** functionality).
5. **A rapid user training in creating quality, and age-appropriate, localized texts, provide the resources to support this** (within a maximum of 20 hours training).
6. **Multilingual support.**
 - a. Must be Unicode-compliant and support Latin, Arabic, and Devanagari scripts and the necessary diacritical,¹ tone and accent marks, as well as any characters from the International Phonetic Alphabet needed for African, Asian, and Latin American languages written in Latin, Arabic or Devanagari scripts.
7. **An option for users to set technical boundaries for their readers:**
 - a. Pre-established settings for appropriate decodable and leveled structures; and
 - b. Customized word lists generated by the writer or from inputted lists or texts in the language.

¹ The International Phonetic Alphabet, The International Phonetic Association, [http://www.langsci.ucl.ac.uk/ipa/IPA_chart_\(C\)2005.pdf](http://www.langsci.ucl.ac.uk/ipa/IPA_chart_(C)2005.pdf). 2013.

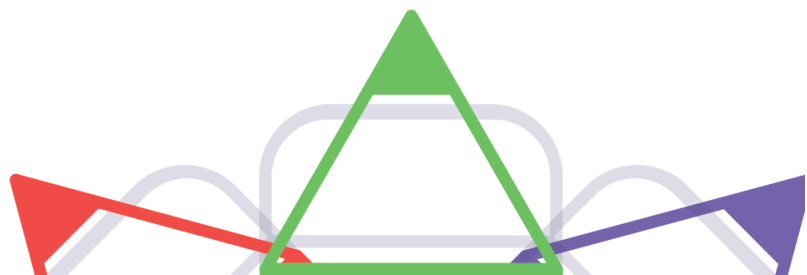


8. **Production of custom word lists from text inputted by users.**
 - a. Users should be able to filter word lists by one or more phonemes/letters.
9. **Based on the reading level of their intended audience, users can create templates that automatically set:**
 - a. Font type
 - b. Font size
 - c. Word and line spacing
 - d. Number of words on a page
 - e. Sentence length
10. **Graphical elements which users can add to their readers.**
 - a. Most common image and graphic file formats including, at a minimum, .jpeg and .png file formats.
11. **Ability to create prepress or print-ready readers that can be printed with a wide range of printing technology (from desktop to large commercial printing equipment)**
 - a. Readers should be able to be outputted to PDF **and** a common editable format in which layout is preserved when opened through free-to-use software (e.g. HTML); but if creating electronic readers, they should be outputted to at least .epub format.
12. **A clear and concise manual written in English at no more than sixth grade level** (as determined by a common [readability program](#)).
 - a. The manual should guide users through the writing and exporting process and provide troubleshooting for any common issues.
13. **Accessibility.**
 - a. Although the main focus of this prize is to create printed texts, developing electronic readers is encouraged, but not required. Any electronic readers supported by the software should include provisions for writers and children with disabilities, for instance text-to-speech functionalities and ability to connect with assistive technologies.
14. **Development conducive to open source coding.**
 - a. Software must be built using only source code you have the exclusive rights to and/or source code released under an open source compatible license.
 - i. The solver must hold all the necessary rights, licences, permissions and consents for images, content etc. contained within your software and manual.
 - b. Code should be well documented, highly readable, and follow the best practices of the development community.
15. **Ability to capture and collate data on software usage and make publically available.**

This is a desired but non-essential criterion. It will, however, be taken into consideration during the judging process.



- a. Software should capture basic meta-data on the readers being produced (language, country etc.).
- b. Software should have the capability for users to upload resulting readers to a freely accessible global repository (the global repository does not need to be created already).



Who are considered early primary school children?

Children in grades/levels one to three or children out of school aged approximately 6-10.

What are decodable texts?

Decodable texts are those that only contain letters and sounds a child or student has already learned. They support explicit reading instruction at the very beginning of reading instruction, generally kindergarten and first grade. This approach to instruction introduces children systematically to letters and sounds, according to a language specific scope and sequence.

What are leveled texts?

Leveled texts are books or stories with increasing levels of difficulty (Cunningham, et. al, 2005). They are designed to provide students with reading materials that range from very simple to more complex¹.

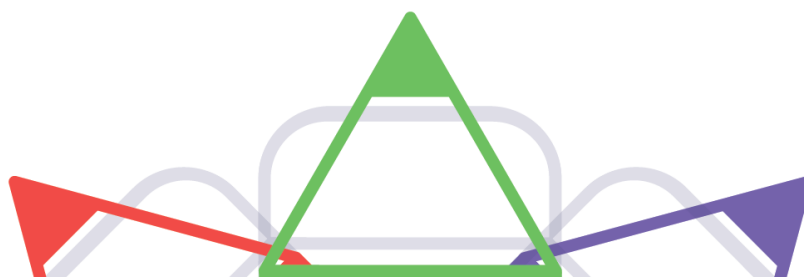
Purposes and characteristics of leveled texts:

- Leveled texts are used to build vocabulary, develop comprehension skills, learn the structure of narrative or expository text, and develop fluency skills.
- They provide young students with reading material that is at their independent reading level (defined as material with no more than five unknown or new words per page).
- They can be placed on a continuum of difficulty, from very easy to very challenging. The 'level' of a book in a leveling system can help a teacher determine whether that book would be an appropriate match for a reader.
- They are not calibrated using quantitative methods but rather by qualitative expert judgment.
- Teachers can use them to evaluate the skill levels of each individual student and to adjust the match between students and books as needed.
- They are not designed for use with only one specific lesson or program; teachers using different programs or curricula can use the same series of leveled readers.

What is a reader?

These are non-fiction and fiction texts (short books or cards) that support reading acquisition.

¹ Davidson, Marcia. "Books that Children Can Read: Decodeable Books and Book Leveling." Cambridge Education. 2013.



What format should the software be in?

Software can be in any form – it can be an application for Windows, Mac OS, Linux or mobile. However successful software needs to be accessible to the widest possible audience; so Windows software must be accompanied by either a Mac or Linux version (and vice-versa) and a mobile app for Android must be accompanied by a mobile app for iOS or Windows Mobile (and vice-versa). Web applications are very welcome, and will be acknowledged in the scoring, but are not eligible on their own (i.e. they must be submitted along with one of the above mentioned combinations). Obviously if your software is available in all of these formats, feel free to upload them all!

Remember, however, that mobile apps must still fulfil all of the software specifications.

Does the software have to be standalone or can it be an add-on?

The software can be an add-on, but it must be for a widely used application such as MSWord. The add-on must still be developed using source code you have the exclusive rights to and/or is licensed under open source; the application it is an add-on to, does not.

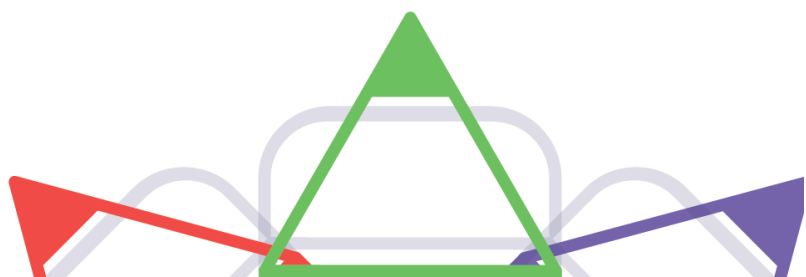
Can I write my software in any programming language?

Yes, you can use any programming language to write your software. However the code must be able to be manipulated in a freely available development environment and use a freely available compiler/interpreter. Remember that winning software will be made available for open source development, so it must be conducive to this.

What does the entry process involve?

You will be asked to answer sets of questions related to you and your software before finally uploading a copy of your software (as a single executable file as well as the source code files, necessary libraries, and associated documentation) and manual. The questions are structured as follows:

- The first set asks for some basic details on you and your team. These are for reporting purposes only and will not be used in evaluating your entry.
- Then you are asked to confirm whether your software meets the essential specifications as listed [here](#).
- The final set of questions gives you an opportunity to provide further detail on the functionalities of your software



Who will be judging my entry and have access to my software?

Entries will be seen by judges from InnoCentive, WV, USAID, the Australian Government, as well as a number of expert judges from a range of fields and industries including literacy, education and technology. Unless you are selected as one of the three finalists, your software will not be shared beyond this circle and judges are committed to maintaining the highest ethical standards when reviewing your entry.

What does Phase Two involve?

In Phase Two, each finalist's submission will be tried out in countries where ACR GCD Partners have early grade reading programs. In each country, three writers will each use all three software submissions and produce leveled and decodable readers. The three entries will be judged by two criteria:

1. The ease of use, as judged by the writers.
2. The quality of the readers produced, as judged by a local team of educators and students.

The software with the highest combined score will be awarded **\$100,000**. Following Phase 1, finalists will receive feedback from expert judges in order to assist them in further developing their software and fixing any bugs in preparation for Phase 2. Phase 2 will be funded and managed by the ACR GCD Partners.

What's the prize?

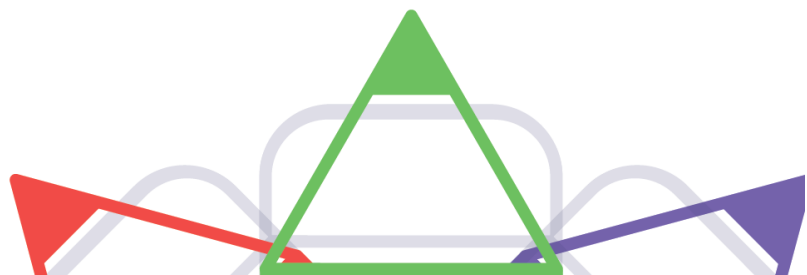
Not only will you be contributing to improving literacy in developing countries, the three finalists in Phase One will receive:

- A \$12,000 cash prize.
- Invitations to exclusive events.
- Global recognition from ACR GCD.
- Expert feedback on their software.
- Exclusive entry into Phase 2 and the chance to win \$100,000.

What language should be used for the software and manual?

The manual should be in English. Software should be Unicode-compliant and user prompts within the software must support at least Latin, Arabic, and Devanagari scripts and the necessary diacritical,² tone and accent marks, as well as any characters from the International Phonetic

² The International Phonetic Association. <http://www.langsci.ucl.ac.uk/ipa/ipachart.html>. 2013.



Alphabet needed for African, Asian, and Latin American languages written in Latin, Arabic or Devanagari scripts.

If I win, what do I have to provide in order to receive the prize?

To receive an award, finalists must provide the software to the public under an open source license, with minimum rights provided to the public equal to a GNU General Public License, whereby the finalist makes the submitted work publicly available under a license that allows the software and source code to be freely used, copied, and shared, and for any derivative works to be freely used, copied, and shared, without charge and with proper attribution. WV, USAID, and the Australian Government each receive a royalty-free, nonexclusive, and irrevocable right to reproduce, publish, or otherwise use the submitted work for Federal purposes, and to authorize others to do so. Under the Federal purposes rights granted by the finalists, USAID intends to provide the submitted work, including the source code to the public free of charge.

Finalists must provide all source code, necessary libraries, and associated documentation. Please note that software must be built using source code you have the exclusive rights to and/or released under an open source compatible license.

Which formats should my software output readers to?

Readers should be able to be outputted to PDF **and** a common editable format in which layout is preserved when opened through free-to-use software (e.g. HTML); but if creating electronic readers, they should be outputted to at least .epub format.

