Auto Multiple Choice - Feature # 27: Circle or Square or Circle within a square

Status:	Closed	**Priority:**	Normal
Author:	red sea	**Category:**	
Created:	03/23/2012	**Assignee:**	
Updated:	03/26/2013	**Due date:**	
Description:	The form of boxes, shading to answer.. Proposed addition to allow the user to choose between: -Circle -Square -Circle within a square best regards		

History

06/04/2012 09:08 am - Pieter Van den Hombergh

red sea wrote:

> The form of boxes, shading to answer..
>
> Proposed addition to allow the user to choose between:
> -Circle
> -Square
> -Circle within a square
>
> best regards

It appears to me that a circle or oval shape is filled in much easier by the students. So I think this could help.

07/16/2012 01:49 pm - Alexis Bienvenüe

- Target version set to 1.2.0

07/16/2012 11:06 pm - Alexis Bienvenüe

- Status changed from New to In Progress

Since revision r1118, you can use something like

<pre>
\AMCboxDimensions{shape=oval,width=1.8ex,height=2.5ex}
</pre>

to make boxes shaped as circles or ovals.

Please have a look.

08/28/2012 10:11 pm - red sea

thanks

i got this error messege:

<pre>
LaTeX Error: Environment tikzpicture undefined.

Undefined control sequence.
Illegal unit of measure (pt inserted).
Missing number, treated as zero.
Illegal unit of measure (pt inserted).
Illegal unit of measure (pt inserted).
Missing number, treated as zero.
Illegal unit of measure (pt inserted).
Undefined control sequence.

</pre>

08/28/2012 10:41 pm - red sea
after i added :
\usepackage{tikz}

to the head, it worked with (shape=oval) only.
when i try to used (shape=circle or squere) i got this message :

<pre>
Package xkeyval Error: value `circl' is not allowed.
</pre>

regards

08/28/2012 10:44 pm - Alexis Bienvenüe
@shape@ must be @square@ or @oval@. You can get a circle using @shape=oval@ with the same value for both @width@ and @height@.

08/28/2012 11:13 pm - red sea
ok thanks , now worked.

can i use "diamond" or "Star" or any shape in tikz?

best regards

08/28/2012 11:15 pm - Alexis Bienvenüe
> can i use "diamond" or "Star" or any shape in tikz?

No. Drawing the shape is not enough to make it work. Some work has to be done for counting the number of black pixels inside the shape. This work has only be done for squares and ovals.

08/28/2012 11:33 pm - red sea
thanks.

regards

10/30/2012 05:07 pm - Pieter Van den Hombergh
I just tested (I had not noticed you already had an implementation).

Two remarks:
The oval shape is quite nice and will help the candidates to fill out the marks much easier.
The pdf rendering with pdflatex is *much* slower. With a 30 question rendered for two students (using csvsimple) using the oval command takes 7.3 seconds real time, with the default rectangular boxes 0.601 seconds, both timed with time pdflatex source.tex. This would make it very slow indeed. My normal exams have between 40 and 80 candidates.

Observing the log output produced while pdflatex makes its run makes me guess that most of the time is used rendering the ovals. I appears to pause at the end of each answer sheet. It has to put 131 ovals per answer sheet.

Would it not be much quicker if you rendered the oval once and re-use it every next time? Both LaTeX and pdf can do define and reuse.

Of course I will have to test the scanning performance and quality and will do that asap.

In any case, it look promising.

10/30/2012 09:57 pm - Alexis Bienvenüe

> Would it not be much quicker if you rendered the oval once and re-use it every next time? Both LaTeX and pdf can do define and reuse.
I'm afraid I don't see how to implement this idea in the @automultiplechoice.sty@ package... Please send a patch if you know how to do this.

10/31/2012 02:11 pm - Pieter Van den Hombergh

Took a look into your implementation and considered the render and reuse.

It might be possible using the \newsavebox, \savebox and \usebox commmand but then you would have to draw 'box' and content differently.
I would consider such to much of a change in the inner workings of the latex sty file. That is not the way to go I think.

I also made and experiment and replaced the 4 tikz arc commands and cycle with one ellipse.
In the exam set I did the experiment on, (16 open questions, 2 open questions with in total 71 tik boxes per form) rendering an exam for 59 students, it reduced the compilation time (one pdflatex run) from 2 minute 24 seconds to 1 minute 8 seconds.

Of course some tweeking might be needed to get the ellipse in the proper proportion, but since it is somehow the closed figure we want, that might do the trick.

At the moment consider it low priority and leave the code as is.

To give you the complete results of my experiment, here you have the ellipse in latex/tikz code, with the arc parts commented out.
I made a copy of the .sty file and edited that before changing anything in your sources.

10/31/2012 02:15 pm - Pieter Van den Hombergh

Would like to upload my oval definition to share it with you.
\renewcommand the \AMC@shape@oval command in a separate .sty file, to experiment with it first.

10/31/2012 02:23 pm - Alexis Bienvenüe

> Would like to upload my oval definition to share it with you.
Sure. Please attach it (click on the "files" banner at the bottom of the page when adding notes).

10/31/2012 02:34 pm - Pieter Van den Hombergh

I just did two experiments with the oval/ellipse form type.

Filling in the form by the students is both faster and cleaner (Filling a round shape with a pen/pencil can be done much quicker)
Scanning and scoring works quite well. I tried hard to fill the shapes properly and have an MSE of 1.6 and a sensitivity of 5.7.

So from my point of view the oval is a *success* and can be used (in my next exam series).
I gladly accept the slower production speed (it only costs my patience, a nice exercise anyway) in exchange for more comfort and reliability at the filling out the form side.

10/31/2012 04:56 pm - Pieter Van den Hombergh

For the time being, I will stick to your original oval, which works best in scanning and the GUI (r 1197)

11/04/2012 03:36 pm - Pieter Van den Hombergh

- File Oval.sty added

I played a little with tikz and came up with an oval version which renders a bit quicker, although not spectacular.

I use a tikz rectangle with rounded corners with a radius of half the box with. The oval is the indiscernible from the original.
Timing (with time) on my machine reveals, for 46 students, 30 question with 262 ovals per exam
with orginal oval real 2m51.417s
with 'rectangle' version, same conditions: real 1m32.480.

I packed the new oval in a Oval.sty, in the attachment and added \usepackage{Oval} to my source.

11/06/2012 02:36 pm - Alexis Bienvenüe

Very nice: this also cut by two the processing time with ovals on my computer. I will replace the current implementation with yours shortly in the development code.

01/15/2013 01:24 pm - Pieter Van den Hombergh

Even nicer would be to have the option to use the default rectangular boxes in the questions and the oval boxes only on the separate answer form, where they have play out their benefits when filling in by pencil or pen.
This could again save a substantial off the processing time, because to all questions the choice boxes are rendered twice, once with the question and once on the separate form.

Something like *\AMCOvalOnAnserformOnly* or similar more appropriate to the current API style.
Or *\AMCQuestionBoxShape* and *\AMCAnswerformBoxShape*, both defaulting to rectangular.

01/15/2013 03:39 pm - Pieter Van den Hombergh

Maybe you might also default to rectngualr boxes in the solution rendering. No one has to fill in the solution anyway ;-))
This will again save some time in the 'update documents' process.

01/15/2013 03:40 pm - Pieter Van den Hombergh

Maybe you might also default to rectangular boxes in the solution rendering. No one has to fill in the solution anyway ;-))
This will again save some time in the 'update documents' process.

01/15/2013 03:48 pm - Pieter Van den Hombergh

I just tested (I had not noticed you already had an implementation).

Two remarks:
The oval shape is quite nice and will help the candidates to fill out the marks much easier.
The pdf rendering with pdflatex is *much* slower. With a 30 question rendered for two students (using csvsimple) using the oval command takes 7.3 seconds real time, with the default rectangular boxes 0.601 seconds, both timed with
<pre>
time pdflatex source.tex
</pre>
This would make it very slow indeed. My normal exams have between 40 and 80 candidates.

Observing the log output produced while pdflatex makes its run makes me guess that most of the time is used rendering the ovals. I appears to pause at the end of each answer sheet. It has to put 131 ovals per answer sheet.

Would it not be much quicker if you rendered the oval once and re-use it every next time? Both LaTeX and pdf can do define and reuse.

Of course I will have to test the scanning performance and quality and will do that asap.

In any case, it look promising.

01/15/2013 05:39 pm - Alexis Bienvenüe

> Would it not be much quicker if you rendered the oval once and re-use it every next time? Both LaTeX and pdf can do define and reuse.
I'm afraid I don't know how to do that, particularly if letters or other characters are to be drawn inside the shapes. Please write the LaTeX code if you can.

01/17/2013 10:19 am - Pieter Van den Hombergh

- File amcnewoval.sty added

Pieter Van den Hombergh wrote:
Alexis Bienvenüe wrote:
> > Would it not be much quicker if you rendered the oval once and re-use it every next time? Both LaTeX and pdf can do define and reuse.
> > I'm afraid I don't know how to do that, particularly if letters or other characters are to be drawn inside the shapes. Please write the LaTeX code if you can.

I did some small experimentations and using \savebox{\amcoval} and \usebox{\amcoval}, in which I saved the oval drawing code in the amcoval savebox. I did these experiments to make sure there is any (speed) gain at all. There is a substantial speed up in this case, although I have no exact numbers and the code is not yet ready for show time. There are some implications though, you may not want.

* The slowness in the current oval rendering is not so much in the rounded corners as in the execution of the tikz environment which is not (yet) optimized for such use case. This can be shown easily by implementing the standard square as a tikz rectangle by just copying the code and leaving the rounded corner feature out. This defines a tikz rectangle which should in theory be no slower then a normal latex fcolorbox, but I am sorry to say it is.
* The optimal speed up can be achieved when you use the savebox/usebox construct in such a way that you can apply it as a blob of ink. You can render this blob once (with tikz for instance) and then more or less stamp it any time you need it. This works fine, speed wise. It is however not compatible with arguments to such macros and your current shape macros take 5 arguments, not the least being _filled_ and the character inside, if to be drawn.
* If you want to use such things you might have to refactor some code (i.e. remove the parameters from the blob) and then some tweaking to get the placement of the check boxes and its content or surroundings right, the things that tikz does for you so elegantly with the node placement. It might also mean that you have to change higher level syntax or semantics. I cannot estimate your development priorities, but this would not land at the top of my list.
* For instance: such change would make the ovals incompatible with drawing the letter or digit inside the check box, which on the other hand is a bad idea for scanning discrimination between checked and unchecked. I would always place the character or digit outside the square or circle.
* This would have an impact on your design, such as using different a rendering for the check boxes in 4 variants: in the exam, on the separate answer form and both in checked and unchecked form.

I will attach a file with my experiment so you can have a look.

While typing this I did a test, which for the speeded up version does
<pre>
time pdflatex source.tex
real 0m25.046s

user 0m24.450s
sys 0m0.452s
</pre>
when called with *time pdflatex source.tex*
and for the subversion only version
<pre>
time pdflatex source.tex
real 3m0.798s
user 2m59.875s
sys 0m0.456s
</pre>
Of course a production run will need 4 pdflatex rederings in total.
Printing speed is not affected as far as I know.

Runs on a same real exam (which I cannot reveal on this forum for obvious reasons) with 30 questions, 47 candidates, 564 pages for a run in the resulting single source.pdf file and 384 check boxes per candidate (192 on the form, 192 in the exam).

01/17/2013 01:40 pm - Alexis Bienvenüe

- File ovals-savebox.sty added

Thanks.
I used your idea to write a patch that should work in all contexts: can you test it? Load this package right after @automultiplechoice@.

01/17/2013 10:46 pm - Pieter Van den Hombergh

Alexis Bienvenüe wrote:
> Thanks.
> I used your idea to write a patch that should work in all contexts: can you test it? Load this package right after @automultiplechoice@.

I will try it tomorrow at work on the same machine I did the timing on. The machine I am now at is too fast ;-)).
Even on this fast machine, running latex on the current exam, two passes for both exam and solution takes a while and is of course only able to use one core per run.

Come to think of it: On a halfway modern machine, having more then one core (this one has six, my faithful laptop, the one I timed with two), you could run the corrections and exam latex run in parallel (ssuming the latex run is error free), which would chop the default 'update documents' process in halve, because both runs would write to different files (I think). Starting two tasks in parallel in the shell is easy enough, as you probably know.

01/17/2013 11:09 pm - Pieter Van den Hombergh

Pieter Van den Hombergh wrote:
> Alexis Bienvenüe wrote:
> > Thanks.
> > I used your idea to write a patch that should work in all contexts: can you test it? Load this package right after @automultiplechoice@.
>
> I will try it tomorrow at work on the same machine I did the timing on. The machine I am now at is too fast ;-)).
> Even on this fast machine, running latex on the current exam, two passes for both exam and solution takes a while and is of course only able to use one core per run.
>
> Come to think of it: On a halfway modern machine, having more then one core (this one has six, my faithful laptop, the one I timed with two), you could run the corrections and exam latex run in parallel (ssuming the latex run is error free), which would chop the default 'update documents' process

in halve, because both runs would write to different files (I think). Starting two tasks in parallel in the shell is easy enough, as you probably know.

I could not resist: It feels a lot quicker.
On this machine (at home, 6 core amd processor with way to much memory so disk buffers are never filled), same exam-conditions as the time measurement above the result is
<pre>
time pdflatex source.tex
real 0m18.967s
user 0m18.605s
sys 0m0.276s
</pre>
which makes me assume your tweak executes in the same time as the original square boxes. (It was 25 seconds real on the laptop, see above).

I also ran the 'update documents' process in the GUI again (on my home machine), and now it is also quick again (with its 4 passes), so this is probably the way to go for this feature. It would cause me to bring down my coffee ration.

Both renderings (Question and Solution) look optically fine, did not yet test the scanning and post processing.

Good work and very quick turnaround on this one. Bravo.

01/17/2013 11:19 pm - Pieter Van den Hombergh
By the way, this morning the students score the exam I used in the timing example. That rendering scanned flawless and have my first result this way. There where almost no incompletely filled boxes, because these ovals are a lot easier to fill with a pencil (preferred) or felt-tip pen.

01/18/2013 11:56 am - Pieter Van den Hombergh
h2. It works.

As promised my test results:
<pre>First run, old style ovals timing.
real 3m0.869s
user 2m59.883s
sys 0m0.404s

Second run, savebox ovals timing
real 0m25.377s
user 0m24.738s
sys 0m0.388s

which is more then 6 times better!

</pre>

I also tested with a complete cycle: prepare, layout detection, print, scan and score and this is also Ok (I scored one exam by hand with a pencil and scanned that. Should give maximum grade of course).

I had the small issue that the corrected version did not have filled boxes, which I did not notice at home, so I must have a further look into that. I will let you know if this problem is on my side.

01/18/2013 01:23 pm - Alexis Bienvenüe
- File deleted (ovals-savebox.sty)

01/18/2013 01:24 pm - Alexis Bienvenüe

- File ovals-savebox.sty added

> I had the small issue that the corrected version did not have filled boxes
I updated the style file with a new version, but I'm not sure it addresses this problem.

01/18/2013 11:43 pm - Pieter Van den Hombergh

h2. Conclusion: Works.

* On both home workstation and laptop your first and second patch both seem to produce the same result.
* Is there anything that I should look into or pay attention to in particular?
* My observations of different behaviour between laptop and computer at home was wrong. The experiments had a different set up so it was probably not something in the first patch
* Anyway, This looks good.
* Will use it for my next exams coming week.

Thanks, as always.

01/21/2013 11:05 pm - Pieter Van den Hombergh

I do not want to open another issue, but was wondering if a complete latex run is required when you select *Update marking scale* in the marking tab.
I just assume that you want/need to re-acquire the questions weight (points) and correct/wrongs, but need not do all the regeneration for all candidates.
We do pre-association, and if you need only information en not the student information, then it might be less time consuming if you processed the material
for one candidate only, thereby gathering al information about the questions, which will be the same for all candidates.

I added it here because it relates to the speed of latex processing in some way. Not doing unneeded work is always faster, whatever algorithm you choose ;-))
One could also imagine that such run would be done in some kind of DRAFT mode, e.g. avoiding complex renderings which might save some processing time.

Maybe my assumption is wrong and do you need to recompute all xy values.

I noticed that the program makes two or three (or maybe four) latex runs when using *Update marking scale*, always using only one cpu. I could imagine that you need two runs for each processing variant, but it might be the two 'exam' runs and the two 'answer' runs could be run in parallel, as long as they use different output (and aux) files, but you could simply put those in separate subdirectories using pdflatex options such as @-output-directory=DIR@ and @-jobname name@. This could also produce some speedup, at least when you have more then one core to spend.

03/25/2013 04:54 pm - Alexis Bienvenüe

- % Done changed from 0 to 100

- Status changed from In Progress to Closed

> was wondering if a complete latex run is required when you select *Update marking scale* in the marking tab.
I'm afraid yes. All candidates may have different questions, different answers, different scoring strategies (the case of random numerical values with a @\AMCnumericChoices@ is a good example). This is not always the case but we have to take these cases into account.

> imagine that such run would be done in some kind of DRAFT mode

Yes, this is a good idea... but I don't know how to do that.

> I noticed that the program makes two or three (or maybe four) latex runs when using Update marking scale
This looks strange. It should be 2 runs at most. Can you confirm that ?

03/26/2013 09:16 am - Pieter Van den Hombergh

Alexis Bienvenüe wrote:
> > was wondering if a complete latex run is required when you select *Update marking scale* in the marking tab.
> I'm afraid yes. All candidates may have different questions, different answers, different scoring strategies (the case of random numerical values with a @\AMCnumericChoices@ is a good example). This is not always the case but we have to take these cases into account.
>
> > imagine that such run would be done in some kind of DRAFT mode
> Yes, this is a good idea... but I don't know how to do that.
>
> > I noticed that the program makes two or three (or maybe four) latex runs when using Update marking scale
> This looks strange. It should be 2 runs at most. Can you confirm that ?

I will be preparing exams for next next week, starting preparations today, so I will have a look and record my findings here.

03/26/2013 12:17 pm - Pieter Van den Hombergh

I rand an exam today and tested a bit with 'update marking' while also looking at the time stamps of the files.

I am running 1.1.1335 (svn:1346) (svn update, make deb and install, ubuntu 12.04).

From what I see, in Marking-> check 'update marking scale', the program only updates the amc-compiled.* files and no other. I think this is correct because as you explained a complete latex run is in order.
So either my earlier observation is wrong or some other files are no longer updated, in particular the DOC-catalog.pdf (nice addition by the way) and the DOC-corrige.pdf are not updated in 'updating marking scale'.

Watching the computer activity during a long run (30 questions, 40+ candidates, update marking) I saw only one pdflatex process as in one run, which means that this is also quite optimal.
I am running the program on the same machine and it appears a lot quicker. Your work must have helped, because my machine is at the same OS level as before.

03/26/2013 12:24 pm - Alexis Bienvenüe

Good. Thanks for the report!

Files

Oval.sty	455 Bytes	11/04/2012	Pieter Van den Hombergh
amcnewoval.sty	1.5 kB	01/17/2013	Pieter Van den Hombergh
ovals-savebox.sty	2.1 kB	01/18/2013	Alexis Bienvenüe