

The jeopardy package*

Robert Mařík
marik@mendelu.cz[†]

April 7, 2010

1 Introduction

Jeopardy like game – redesigned the `jj_game.cls` by D. P. Story. The package requires to load `exerquiz` and `dljslib` packages from AcroTeX bundle. The current only supported way to produce pdf file is `pdftex`. The main properties include

- In contrast to `jj_game.cls`, the `oQuestion` environment is used to allow fill-in questions.
- The `proofing` option is introduced. Use this option to typeset your answers into the document. To check the answers for `\RespBoxMath` and `\RespBoxTxt` commands remember to use transparent background for these text fields or use the browser which does not show these fields, like `xpdf`.
- Option `twoplayer` builds game for two players.
- Option `bgpicture` is used to place the picture below the buttons (`graphicx` package is used). The name of the picture is stored in the `\JeopardyPictureFile` command. Default value is `picture.jpg`. You can change it by using command like `\def\JeopardyPictureFile{blabla.jpg}`. If you use this option, you may also consider the (rather tricky) option `finetune`. This option is described with the corresponding code at the end of this file. The `bgpicture` option is automatically turned on when `picture` option is active.
- The option `picture` is introduced for another version of the game. The user gets no score but uncovers parts of a picture hidden by buttons and (after wrong answer) non-transparent fields. It can be used in the class and students can guess what is on the picture. The option adds the "Solution" button below the gameboard. If the user clicks this button, all buttons from not yet answered questions and fields from wrong answered questions are

*This document corresponds to `jeopardy v1.1a`, dated 2010/04/07.

[†]Supported by the grant 18/2006 of the Fund for Development of Czech Universities.

hidden and the picture is fully visible. After the second click on this button, a `\ChampionMsg` is shown. This can be used for example to store the name of a person on the picture.

- The user is not allowed to browse pages of the game without clicking in gameboard. We use the (not portable) `\pdfpageattr` command to achieve this behavior. This can be turned off by `allowpeeking` option.
- The `evalonblur` option may be used. This concerns filling forms for `\RespBoxTxt` and `\RespBoxMath` questions and the following problem with switching keyboard layout: When filling the text field, if user switches the keyboard layout¹ using keyboard shortcut on Linux, the (not yet finished and hence incorrect) answer is evaluated and marked as wrong. Known workarounds:
 - The user should erase the contents of the field completely, switch the keyboard, click into the field and type his/her answer.
 - The user should change keyboard using mouse and Gnome applet.
 - The author can compile the file with `evalonblur` option. The answer is evaluated after `shift+click`-ing outside the field.

This package has been tested on Texlive2005+Linux (pdf_εTeXk, Version 3.141592-1.30.4-2.2 (Web2C 7.5.5)), with `exerquiz 2006/18/06 v6.05e`.

2 Usage

To instal the style run `latex jeopardy.ins` from the directory with `jeopardy.dtx`. You get files `jeopardy.sty` and `jeopardy.cfg`. Copy both files where L^AT_EX finds them and include your customizations into `jeopardy.cfg`.

For sample file look at the files `game*.tex`. The number of questions and categories is written into `aux` file and stored in macros `\NumberOfQuestions` and `\NumberOfCategories`. From this reason, the file has to be two times `pdflatex`-ed to build the game. The `aux` file contains also definitions for the category names. These names are not used when `picture` option is active.

`\MakeGameBoard`

Macro `\MakeGameBoard` typesets the game board. This should be at the begin of the file and followed by `category` environments. The dimensions of cells are `\CellWidth` and `\CellHeight`. The user can set these dimensions after `\begin{document}` or with the `\SetGameWidth` and `\SetGameHeight` commands. Write e.g. `\SetGameHeight{10cm}`. The fields are created via `\pushButton` and `\textField` commands and the user may change the default behavior using command available in the package `eforms.sty`. The colors for wrong and right messages are set in `exerquiz` package and stored in `\correctColor` and `\wrongColor` macros. The tokens registers `\GrandPoohbahtoks`, `\Celltoks` and `\Scoretoks` are used to build the GrandPoohbah field (see below), cell fields and score field.

¹necessary, for example, to write powers on Czech keyboard

You can set it by e.g. `\GrandPoohbahtoks{\Q{0}\textColor{1 0 0 rg}}`. All these registers are empty by default. The exception is if the `picture` option is used, the `\Celltoks` is set to `\BG{0 0 0}`. If you change it, remember to use non-transparent background for cells!

<code>\AfterGameBoard</code>	Macro <code>\AfterGameBoard</code> is executed between gameboard and score field.
<code>\everyCategoryHead</code>	Tokens register <code>\everyCategoryHead</code> is used at the begin of every category head. You can set it by something like <code>\everyCategoryHead{\textrm\footnotesize}</code>
<code>\JeopardyTitle</code>	The <code>\JeopardyTitle</code> macro contains the title of the game. The title is the hyperlink anchor used to return from questions to the gameboard. You can set it by e.g. <code>\def\JeopardyTitle{The title of the game}</code> .
<code>category</code>	The <code>category</code> environment has one parameter – the name of the category. It encloses <code>question</code> environments. The category name is not used if the <code>picture</code> option is active. In this case the name is replaced by the capital letter A, B, C, ... In this case you (or your students) can refer to the fields like on chessboard.
<code>question</code>	The <code>question</code> environment contains a question and answers for multiple choice questions. The answers are introduced by <code>\Ans0</code> (wrong answers) or <code>\Ans1</code> (correct answers). The environment includes an <code>oquestion</code> environment and hence it may contain <code>\RespBoxMath</code> or <code>\RespBoxTxt</code> command.
<code>\Goal</code>	If the user finishes the game with at least <code>\Goal</code> points, a hidden banner (GrandPoohbah field) with message stored in <code>\ChampionMsg</code> command is shown. The default initial value $45 \cdot (1 + \text{numQuestions}) \cdot \text{numQuestions} \cdot \text{numCategories} \cdot \text{JeopardyScaleFactor}$ means 90 percent of the maximal point score. The banner has height <code>\GrandPoohbahHeight</code> . This macro is not used if the option <code>picture</code> is active. In this case the <code>\ChampionMsg</code> should contain the name of the picture (for example the name of the person on photograph) and it can be shown by doubleclicking "Solution" button at the bottom.
<code>\SetGameWidth</code>	Macro <code>\SetGameWidth</code> sets the <code>\GameWidth</code> register with the width of the game desk. It evaluates the width of cells as quotient of the parameter of the macro and the number of categories. Macro <code>\SetGameHeight</code> works in a similar way if the option <code>picture</code> is not active. If the option <code>picture</code> is active, the height of the game is evaluated from the dimensions of the picture. The width of the picture is scaled into <code>\GameWidth</code> and the height is established from the height of the scaled picture.

3 Implementation

Check that we use `pdflatex` and both `exerquiz` and `dljslib` packages are loaded.

```

1 (*package)
2
3 \ifx\shortquiz\undefined
4 \PackageWarning{Jeopardy}{!!!!!!!!!!!!!! Load exerquiz package !!!!!!!!!!!!!}
5 \def\finito{\@@end}
6 \expandafter\finito\fi
7
8 \ifx\dljsRegister\undefined
9 \PackageWarning{Jeopardy}{!!!!!!!!!!!!!! Load dljslib package !!!!!!!!!!!!!}

```

```

10 \def\finito{\@@end}
11 \expandafter\finito\fi
12
13 \if\eq@drivernum1%% pdftex
14 \typeout{*** using pdftex option ***}
15 \def\JeopardyNoPeekCommand{
16   \edef\ThisPage{/AA<</O<</S/JavaScript/JS(kontrola(
17     \the\QuestCount,\the\CatCount);)>> >>}
18   \global\pdfpageattr=\expandafter{\ThisPage}%
19 }
20 \else
21 \def\JeopardyNoPeekCommand{}
22 \typeout{!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!}
23 \typeout{!!!!!!!! The package has been tested with pdflatex only.}
24 \typeout{!!!!!!!! Use pdflatex drivers in eforms.sty and exerquiz.sty.}
25 \typeout{!!!!!!!! On your own risk you can use other drivers
26   with *allowpeeking* option.}
27 \PackageWarning{Jeopardy}{No pdftex drivers for eforms and exerquiz}
28 \fi
29
30 \RequirePackage{multido}
31
32 \DeclareOption{czech}{
33 \def\formessage{za}
34 \def\Scoremessage{Body}
35 \def\fairmsg{Nesidit! Odpovídate na jinou otázku!}
36 \def\who@answers@msg#1{Odpovídá hráč #1}
37 \def\RightWrongMessages{
38 aCM = new Array\
39 ( "Spravne! Jen tak dal!",
40   "Vyborne! Gratuluji!",
41   "Dobre! Pokracujte ve spravne nacate praci!",
42   "Ano, ano, ano, ano!",
43   "Spravne! Zkuste i tezsi otazky",
44   "Spravna odpoved!",
45   "Presne tak! Ziskavate dalsi body"
46 );
47 aEM = new Array\
48 ( "To neni spravne! Zkuste jinou otazku.",
49   "Chyba. Nehadejte!",
50   "Prosim! Prosim! Premyslejte nez odpovite!",
51   "Dalsi chyba! Hadanim se daleko nedostanete!",
52   "Ale ne! Kdepak! Body dolu.",
53   "Ne, zkuste jinou otazku.",
54   "Zase spatne! Bohuzel, vase konto se opet snizuje!"
55 );}
56 \def\NotAllowedMessage{"Nesmite menit odpovedi!"}
57 \def\RightMessage{Spravne}
58 \def\WrongMessage{Spatne}
59 \def\JeopardyTitle{Tituln\'{i} strana}

```

```

60 \def\ChampionMsg{Bravo! Mistr nad mistry!}
61 }
62
63 \def\formessage{for}
64 \def\Scoremessage{Score}
65 \def\NotAllowedMessage{"Changing your answer is not allowed!"}
66 \def\RightMessage{Right}
67 \def\WrongMessage{Wrong}
68 \def\fairmsg{Play Fair! Answer the question you have clicked!}
69 \def\who@answers@msg#1{Player #1}
70 \def\RightWrongMessages{
71 aCM = new Array
72 ( "That's Right! Way to go, Kiddo!",
73   "Wonderfully Done! Congrats!",
74   "Very Good! Keep up the good work!",
75   "Good, good, good, good!",
76   "Gee, you're smart! Try a harder one!",
77   "You're pretty sharp! Continuez, s'il vous pla\string\356t!",
78   "You're the tops! But wait...try another."
79 );
80 aEM = new Array
81 ( "That's wrong! Sorry. Try Again.",
82   "Error again. Don't guess, I'm watching!",
83   "Please! Please! Work it out first!",
84   "An error has insinuated itself into your calculations. Recalibrate your thinking!",
85   "Bah! Humbug! You're haunted by Errors!",
86   "Try again. This time with order and method.",
87   "Wrong again! Use your little gray cells!"
88 );
89 }
90 \def\JeopardyTitle{Jeopardy Game}
91 \def\ChampionMsg{You are a Champion!}
92
93 \newif\ifJeopardyNoPeeking
94 \JeopardyNoPeekingtrue
95 \DeclareOption{allowpeeking}{\JeopardyNoPeekingfalse}
96
97 \newif\ifEvalOnBlur
98 \EvalOnBlurfalse
99 \DeclareOption{evalonblur}{\EvalOnBlurtrue}
100
101 \newif\ifdouble \doublefalse
102 \def\JeopardyScaleFactor{1}
103 \DeclareOption{double}{\doubletrue\def\JeopardyScaleFactor{2}}
104
105 \newif\ifoneplayer \oneplayertrue
106 \DeclareOption{twoplayers}{\oneplayerfalse}
107
108 \newif\if@jeopardyFineTune \@jeopardyFineTunefalse
109 \DeclareOption{finetune}{\@jeopardyFineTunetrue}

```

```

110
111 \DeclareOption{proofing}{\eq@proofingtrue \JeopardyNoPeekingfalse}
112 \definecolor{webgreen}{rgb}{0,.5,0}
113
114 \newif\if@jeopardyBgPicture
115 \@jeopardyBgPicturefalse
116 \DeclareOption{bgpicture}{\@jeopardyBgPicturetrue \everyPushButton{\BC{}}}
117
118 \newif\if@jeopardyPicture
119 \@jeopardyPicturefalse
120 \def\Picture{0}
121 \DeclareOption{picture}{\def\Picture{1}\def\formessage#1.{}%
122   \@jeopardyPicturetrue
123   \@jeopardyBgPicturetrue
124   \Celltoks{\BG{0 0 0}}
125   \everyPushButton{\BC{}}}
126 }
127
128
129 \newtoks\GrandPoohbahtoks
130 \newtoks\Scoretoks
131 \newtoks\Celltoks
132 \newtoks\Radiotoks
133 \newtoks\Playertoks
134
135
136 \ProcessOptions
    We load graphicx package for inserting bitmaps.
137 \if@jeopardyBgPicture
138 \RequirePackage{graphicx}
139 \fi
140
141 \sqTurnOffAlerts
142
143 \newcount \CatCount
144 \newcount \QuestCount
145 \newcount \NumberOfCategories
146 \newcount \NumberOfQuestions
147 \newcount \QuestCountA \QuestCountA=-1
148 \newcount \AnswerCount
149 \newcount \OQuestionCount
150 \newcount \temporarycount
151 \newcounter{tempcounter}
152 \everyRespBoxMath{\BG{1 1 1}}
153 \everyRespBoxTxt{\BG{1 1 1}}
154
155 \newlength \GrandPoohbahHeight
156 \GrandPoohbahHeight = 10pt
157

```

Macros for setting game width and height.

```
158 \newlength\GameWidth
159 \newlength\GameHeight
160 \newlength\CellWidth
161 \newlength\CellHeight
162 \newlength\ScoreCellHeight
163 \def\SetGameWidth#1{\GameWidth=#1\relax
164   \CellWidth=\GameWidth
165   \ifnum\NumberOfCategories=0
166     \divide\CellWidth by 10
167   \else
168     \divide\CellWidth by \NumberOfCategories
169   \fi
170 }
171 \def\SetGameHeight#1{\GameHeight=#1\relax
172   \CellHeight=\GameHeight
173   \ifnum\NumberOfCategories=0
174     \divide\CellHeight by 10
175   \else
176     \divide\CellHeight by \NumberOfQuestions
177   \fi
178   \ScoreCellHeight=\CellHeight
179 }
```

We do some basic initializations for dimensions of gameboard.

```
180 \AtBeginDocument{
181   \SetGameWidth{0.8\linewidth}
182   \SetGameHeight{0.6\textheight}
183 }
```

Macros to typeset the picture. The default name of the picture is `picture.jpg`. You can redefine the command `\insertJeopardyPicture` to replace the picture e.g. by a text written in T_EX. To do this, replace the `\includegraphics` command by something else, e.g. `minipage` environment.

```
184 \newbox\JeopardyPictureBox
185 \def\JeopardyPictureFile{picture.jpg}
186 \def\insertJeopardyPicture{%
187   \setbox\JeopardyPictureBox=%
188   \hbox{\includegraphics [width=\GameWidth]{\JeopardyPictureFile}}%
189   \SetGameHeight{1.02\ht\JeopardyPictureBox}%
190   \dimen0=\CellHeight
191   \dimen1=0.5\dimen0
192   \hbox to 0 pt{\vbox to 0 pt{\kern -\dimen1\box\JeopardyPictureBox\vss}\hss}%
193 }
194
```

We define `category` and `question` environments. We read the names of the categories from the `aux` file and store these names in macros. These names are not used when `picture` option is active. In this case we convert the number of category into capital letter.

```

195 \if@jeopardyPicture
196   \def\CategoryNameDef#1#2{%
197     \setcounter{tempcounter}{#1}%
198     \expandafter\xdef\csname CategoryName#1\endcsname{%
199       \Alph{tempcounter}\strut}}%
200 \else
201   \def\CategoryNameDef#1#2{%
202     \expandafter\gdef\csname CategoryName#1\endcsname{#2}}%
203 \fi
204

```

The category environment simply increases the counter, writes the name into aux file and checks the number of questions in the last category environment.

```

205 \newenvironment{category}[1]{%
206   \global\advance \CatCount by 1
207   \global\QuestCount=0
208   \immediate\write\@auxout{\string\CategoryNameDef{\the\CatCount}{#1}}%
209 }
210 {%
211   \ifnum \QuestCountA=-1 \global\QuestCountA=\QuestCount
212   \else
213     \ifnum\QuestCountA=\QuestCount
214     \else
215       \PackageWarning{Jeopardy}
216       {Bad number of questions in Category \the\CatCount.}
217     \fi
218   \fi
219 }

```

`question` environment increases the counter, places the action to check that the page is opened from the gameboard, writes the title of the question and defines macro `\Ans` which typesets buttons with correct and incorrect answers. It also starts the `oQuestion` environment and hence the commands `\RespBoxMath` and `\RespBoxTxt` from `exerquiz.sty` can be used. At the end of the environment we close the `oQuestion` environment and start new page.

```

220 \newenvironment{question}{%
221   \global\advance \QuestCount by 1\relax
222   \AnswerCount=0
223   \ifJeopardyNoPeeking
224     \JeopardyNoPeekCommand
225   \fi
226   \begin{center}%
227     \hypertarget{\the\QuestCount\the\CatCount}%
228     {\csname CategoryName\the\CatCount\endcsname} %
229     \formessage{} \the\QuestCount00.}%
230   \end{center}%
231 \def\Ans##1{\par
232 \ifnum\AnswerCount=0 \vskip\medskipamount\fi
233 \advance\AnswerCount by 1%
234 \setcounter{tempcounter}{\the\AnswerCount}%

```



```

235 \leavevmode\hbox to 0 pt{%
236 \hss
237 \ifeq@proofing \def\temp{##1}\if\temp1$\color{webgreen}\bullet$ \fi\fi
238 \pushButton[\CA{\alph{tempcounter}}%
239 \A{\JS{useranswer(##1,\the\QuestCount,\the\CatCount);}}]%
240 {{Answer\the\AnswerCount-\the\QuestCount-\the\CatCount}}%
241 {1.3em}{1.3em}%
242 \hskip1em}%
243 \leftskip2em}%
244 \begin{oQuestion}{Q\the\QuestCount\the\CatCount}%
245 }{\par\end{oQuestion}\newpage}
246

```

At the end of document we write the number of questions and categories into aux file.

```

247 \AtEndDocument{
248 \immediate\write\@auxout{\string\global\string\NumberOfCategories=\the\CatCount}
249 \immediate\write\@auxout{\string\global\string\NumberOfQuestions=\the\QuestCount}
250 }

```

Here we redefine some internal macros of `exerquiz.sty`.

```

251 \ifEvalOnBlur
252 \def\@@RespBoxMathActions
253 {%
254 \AA{
255 \AAFormat{\eqFormatArg}%
256 \AAOnFocus{\JS{var retn = null;}}%
257 \AAOnBlur{\JS{retn = \processJSfunc(\rbArgs,\compareJSfunc);
258 if (event.shift){
259 if (retn==true) {useranswer(1,\the\QuestCount,\the\CatCount); }
260 else {if (retn==false) {useranswer(0,\the\QuestCount,\the\CatCount);}
261 else {app.alert("Probably a syntax error somewhere.");}}}
262 else
263 {app.alert("Exit the field with Shift+Click.");
264 this.getField("obj.\oField.\thequestionno").setFocus();
265 }}
266 }
267 }
268 }
269 \def\@@RespBoxTxtActions
270 {%
271 \AA{\AAOnBlur{\JS{retn = ProcRespTxt(\rbTxtAlt);
272 if (event.shift){
273 if (retn==true) {useranswer(1,\the\QuestCount,\the\CatCount); }
274 else {if (retn==false) {useranswer(0,\the\QuestCount,\the\CatCount);}
275 else {app.alert("Probably a syntax error somewhere.");}}}
276 else
277 {app.alert("Exit the field with Shift+Click.");
278 this.getField("obj.\oField.\thequestionno").setFocus();
279 }} }%
280 \AAOnFocus{\JS{var retn = null;}}

```

```

281 }
282 }
283 \def\jsRespBox[#1]#2{%
284 \def\compareJSfunc{#1}\def\processJSfunc{#2}%
285 \ifeq@proofing\makebox[0pt][l]{\space\math@correctAnswer}\fi
286 \eq@TextField % send to the driver-dependent macro
287 \ \pushButton{button}{10bp}{10bp}}%
288 \def\eq@@RespBoxTxt{%
289 \ifeq@proofing\makebox[0pt][l]{\space\txt@correctAnswer}\fi
290 \eq@TextField
291 \ \pushButton{button}{10bp}{10bp}}%
292 \else
293 \def\@@RespBoxMathActions
294 {%
295 \AA{\AAKeystroke{if(event.willCommit) {
296     retn = \processJSfunc(\rbArgs,\compareJSfunc);
297     if (retn==true) {useranswer(1,\the\QuestCount,\the\CatCount); }
298     else {if (retn==false) {useranswer(0,\the\QuestCount,\the\CatCount);}
299         else {app.alert("Probably a syntax error somewhere.");}}}
300     }%
301     \AAFormat{\eqFormatArg}%
302     \AAOnFocus{\JS{var retn = null;}}%
303 }%
304 }%
305 \def\@@RespBoxTxtActions
306 {%
307 \AA{\AAKeystroke{if(event.willCommit) {
308     retn = ProcRespTxt(\rbTxtAlt);
309     if (retn==true) {useranswer(1,\the\QuestCount,\the\CatCount); }
310     else {if (retn==false) {useranswer(0,\the\QuestCount,\the\CatCount);}
311         else {app.alert("Probably a syntax error somewhere.");}}}
312     }%
313     \AAOnFocus{\JS{var retn = null;}}
314 }
315 }
316 \fi %%% \ifEvalOnBlur

```

Macro for typesetting the page with title (hyperref anchor), hidden GrandPoohbah field and gameboard.

```

317 \newtoks\everyCategoryHead
318 \everyCategoryHead{%
319 \def\MakeGameBoard{%
320 \begin{center}%
321 \hypertarget{GameBoard}{\JeopardyTitle}%
322
323 \edef\temp{\noexpand\textField[\noexpand\Ff{\noexpand\FfReadOnly}%
324 \noexpand\F{\noexpand\FHidden}%
325 \noexpand\Q{1}%
326 \noexpand\textSize{0}%
327 \noexpand\V{\ChampionMsg}\the\GrandPoohbahtoks%

```

```

328     ]{GrandPoohbah}{\textwidth}{\the\GrandPoohbahHeight}}%
329 \temp
330
331 \leavevmode
332 \ifnum\NumberOfQuestions>0%
333 \ifnum\NumberOfCategories>0%
334 \multido{\i=1+1}{\NumberOfCategories}{\leavevmode
335     \vtop{\hsize=\CellWidth \centering
336         \the\everyCategoryHead\csname CategoryName\i\endcsname }}\%
337 \leavevmode
338 \if@jeopardyBgPicture\insertJeopardyPicture\fi
339 \Multido{\i=1+1}{\NumberOfQuestions}%
340 {\edef\j{\i}%
341     \if@jeopardyPicture\hbox to 0 pt{\hss$\j$ } \fi
342     {\Multido{\i=1+1}{\NumberOfCategories}{%
343         \edef\temp{\noexpand\textField[%
344             \noexpand\Ff{\noexpand\FfReadOnly}%
345             \noexpand\S{S}%
346             \noexpand\Q{1}%
347             \the\Celltoks]{Field\j\i}{\CellWidth}{\CellHeight}}%
348         \hbox to 0 pt{\temp\hss}%
349         \if@jeopardyPicture
350         \pushButton[\S{B}%
351             \A{\JS{kliknuti(\j,\i);}}]{Button\j\i}{\CellWidth}{\CellHeight}%
352         \else
353         \pushButton[\S{B}\CA{\j00}%
354             \A{\JS{kliknuti(\j,\i);}}]{Button\j\i}{\CellWidth}{\CellHeight}%
355         \fi
356         }\\leavevmode}%
357     }%
358     \fi
359     \fi
360     \par
361     \AfterGameBoard
362     \if@jeopardyPicture
363     \smallskip\leavevmode
364     \hfill\pushButton[\A{\JS{hideall();}}\CA{Solution}]{Solution}{1.5cm}{12bp}
365     \else
366     \@makeScoreField
367     \fi
368 \end{center}%
369 \newpage
370 }%
371
372 \def\@makeScoreField
373 {
374     \ifoneplayer
375     \dimen0=\NumberOfCategories\CellWidth
376     \edef\temp{\noexpand\textField[\noexpand\V{}]\noexpand\textSize{0}
377         \noexpand\Q{1}

```

```

378 \noexpand\Ff{\noexpand\FfReadOnly}
379 \the\Scoretoks
380 ]{Total.1}{\dimen0}{\ScoreCellHeight}}%
381 \temp
382 \else%two players
383 \dimen0=\NumberOfCategories\CellWidth
384 \dimen1=0.5\dimen0
385 \edef\temp{\noexpand\textField[\noexpand\V{\noexpand\textSize{0}
386 \noexpand\Q{1}
387 \noexpand\Ff{\noexpand\FfReadOnly}
388 \the\Scoretoks
389 ]{Total.1}{\dimen1}{\ScoreCellHeight}}%
390 \temp
391 \edef\temp{\noexpand\textField[\noexpand\V{\noexpand\textSize{0}
392 \noexpand\Q{1}
393 \noexpand\Ff{\noexpand\FfReadOnly}
394 \the\Scoretoks
395 ]{Total.2}{\dimen1}{\ScoreCellHeight}}%
396 \temp\
397 \hbox to \hsize{\hss\hbox to \dimen0
398 {\hss
399 \radioButton[\V{1}\Ff{\FfNoToggleToOff}\A{JS{player=1;this.getField("whoanswers").value=
400 \hss
401 \hbox % to 0 pt
402 {\hss
403 \edef\temp%
404 \noexpand\textField[\noexpand\Q{1}\noexpand\Ff{\noexpand\FfReadOnly}\noexpand\V{\V{
405 {whoanswers}{0.6\dimen0}{10pt}}\temp\hss
406 }
407 \hss
408 \radioButton[\Ff{\FfNoToggleToOff}\A{JS{player=2;this.getField("whoanswers").value=
409 \hss}\hss
410 }%
411 \fi
412 }
413
414 \let\AfterGameBoard\relax
415 \edef\Goal{45*(1+numQuestions)*numQuestions*numCategories*\JeopardyScaleFactor}

```

Basic Java functionality is rewritten from `jj_game.cls`. We start with initial declarations.

```

416
417 \def\update@who@answers{
418 if (player==1) this.getField("whoanswers").value= "\who@answers@msg{A}";
419 else this.getField("whoanswers").value= "\who@answers@msg{B}";
420 }
421
422 \edef\JeopardyUpdatescore
423 {
424 \ifoneplayer

```

```

425 this.getField("Total.1").value="\Scoremessage: "+TotalScoreA;
426 if (TotalScoreA>0)
427 this.getField("Total.1").textColor=\correctColor;
428 else this.getField("Total.1").textColor=\wrongColor;
429 \else
430 this.getField("Total.1").value="\Scoremessage: "+TotalScoreA;
431 if (TotalScoreA>0)
432 this.getField("Total.1").textColor=\correctColor;
433 else this.getField("Total.1").textColor=\wrongColor;
434 this.getField("Total.2").value="\Scoremessage: "+TotalScoreB;
435 if (TotalScoreB>0)
436 this.getField("Total.2").textColor=\correctColor;
437 else this.getField("Total.2").textColor=\wrongColor;
438 if (player==1) {player=2; this.getField("radio").value=2;}
439 else {player=1; this.getField("radio").value=1};
440 \update@who@answers
441 \fi
442 }
443
444
445
446 \begin{insDLJS}[dljslibRmb]{dljsliRmb}{Rmb}
447 numQuestions = \the\NumberOfQuestions;
448 numCategories = \the\NumberOfCategories;
449 var player=1;
450 RowClick=0;
451 ColClick=0;
452 SaveRowClick=0;
453 SaveColClick=0;
454 totalQuestions = numQuestions * numCategories;
455 var QAed = "";
456 for (var i=0; i< totalQuestions; i++)QAed += "0";
457 nEM = 0;
458 nCM = 0;
459 var submitted = 0;
460 TotalScoreA = 0;
461 TotalScoreB = 0;
462 NumQAed = 0;
463 Goal = \Goal;
464 \RightWrongMessages
465 var Picture=\Picture;
466 var JeopardyPass=0;

```

These functions check that the page has been opened from the Gameboard and save the row and column number of the button which has been clicked. The OpenAction on the page is turned off after answering all questions.

```

467
468 function kontrola(row,col)
469 {
470   if (NumQAed<totalQuestions)

```

```

471 {
472   var p = numQuestions*(col-1)+row-1;
473   if (row!=RowClick) gotoNamedDest("GameBoard");
474   if (col!=ColClick) gotoNamedDest("GameBoard");
475   RowClick=0;
476   ColClick=0;
477 }
478 };
479
480 function kliknuti(row,col)
481 {RowClick=row;
482 ColClick=col;
483 SaveColClick=col;
484 SaveRowClick=row;
485 gotoNamedDest(row+""+col);
486 };
487

```

This function notifies the user. If the option `picture` is active, it also and hides the field on Gameboard in the case of correct answer.

```

488 function useranswer(ans,rowans,colans)
489 {
490   if ((rowans!=SaveRowClick) || (colans!=SaveColClick))
491   {
492     app.alert("\fairmsg");
493     gotoNamedDest("GameBoard");
494     return null;
495   }
496   row= SaveRowClick;
497   col= SaveColClick;
498   this.getField("Button"+row+col).hidden=true;
499   var p = numQuestions*(col-1)+row-1;
500   if (QAed.charAt(p)==0)
501   {
502     QAed = QAed.substring(0,p)+1+QAed.substring(p+1,totalQuestions);
503     if (ans==1)
504     {
505       if (Picture==1)
506       {
507         this.getField("Field"+row+col).hidden=true;
508       }
509       else
510       {
511         this.getField("Field"+row+col).textColor=\correctColor;
512         this.getField("Field"+row+col).value="\RightMessage";
513       }
514       if (player==1)
515         TotalScoreA += \JeopardyScaleFactor*row*100;
516       else
517         TotalScoreB += \JeopardyScaleFactor*row*100;

```

```

518     app.alert(aCM[nCM],3);
519     if (++nCM > 6) nCM=0;
520 }
521 else
522 {
523     if (Picture!=1)
524     {
525         this.getField("Field"+row+col).textColor=\wrongColor;
526         this.getField("Field"+row+col).value="\WrongMessage";
527     }
528     if (player==1)
529         TotalScoreA -= \JeopardyScaleFactor*row*100;
530     else
531         TotalScoreB -= \JeopardyScaleFactor*row*100;
532     app.alert(aEM[nEM],3);
533     if (++nEM > 6) nEM=0;
534 }
535 if ((++NumQAed>=totalQuestions) && ((TotalScoreA+TotalScoreB)>=Goal))
536 {
537     if (Picture!=1) this.getField("GrandPoohbah").hidden=false;
538 }
539 }
540 else
541 app.alert(\NotAllowedMessage);
542 gotoNamedDest("GameBoard");
543 if (Picture!=1)
544 {
545     \JeopardyUpdatescore
546 }
547 };

```

This function hides in the first call all fields and buttons on the gameboard and after the second call shows the \ChampionMsg. In addition, it sets NumQAed=totalQuestions and the user is allowed to go through the pages of the document.

```

548 function hideall(){
549     NumQAed=totalQuestions;
550     var i=0;
551     var j=1;
552     if (JeopardyPass==0)
553     {
554         JeopardyPass=1;
555         for (var i=0; i< numQuestions; i++)
556         {
557             for (var j=0; j< numCategories; j++)
558             {
559                 row=i+1; col=j+1;
560                 this.getField("Field"+row+col).hidden=true;
561                 this.getField("Button"+row+col).hidden=true;
562             }

```

```

563   }
564 }
565 else
566   this.getField("GrandPoohbah").hidden=false;
567 }

```

We redefine function used in `exerquiz` package. We will turn off notification of the correct and incorrect answers by changing the strokecolor of the text fields.

```

568 function notifyField(success, flag, fieldname) {
569   return success;
570 }
571 \end{insDLJS}

```

The fields in Acrobat have bigger dimension than their dimensions visible by \TeX . From this reason they overlap a bit. The user may try to remove this overlap, which is not nice together with picture, using option `finetune`. This option tries to resolve this problem by using smaller buttons than the `\CellWidth` and `\CellHeight`. The user has to set manually the dimension `\AdditionalShift` which is used to shift the image vertically to adjust correct position. (Sorry for the inconvenience.) I know that this solution is not clean enough but ... I did my best. At least for this verion of `jeopardy` style. Look at the files `game1a.tex` and `game4.tex`. You can also try to remove the option `finetune` from these files and compaile again to note the difference.

```

572 \newlength\AdditionalShift
573 \AdditionalShift=0pt
574
575 \if@jeopardyFineTune
576 \newlength \finetunelength
577 \newlength \finetuneCellWidth
578 \newlength \finetuneCellHeight
579 \finetunelength=1pt
580
581 \def\insertJeopardyPicture{%
582   \dimen1=\finetunelength
583   \multiply\dimen1 by \NumberOfCategories
584   \advance\dimen1 by \GameWidth
585   \setbox\JeopardyPictureBox=%
586   \hbox{\includegraphics[width=\dimen1]{\JeopardyPictureFile}}%
587   \dimen1=\finetunelength
588   \multiply\dimen1 by \NumberOfQuestions
589   \dimen2=\ht\JeopardyPictureBox
590   \advance\dimen2 by -1.9\dimen1
591   \SetGameHeight{\dimen2}%
592   \dimen0=\CellHeight
593   \divide\dimen0 by 2
594   \hbox to 0 pt{\vbox to 0 pt{\kern -\dimen0\kern-\AdditionalShift
595     \box\JeopardyPictureBox\vss}\hss}%
596 }
597
598

```



```

599
600 \def\MakeGameBoard{%
601   \begin{center}%
602     \fboxsep=\finetunelength
603     \fboxrule=0pt
604     \hypertarget{GameBoard}{\JeopardyTitle}%
605
606     \edef\temp{\noexpand\textField[\noexpand\Ff{\noexpand\FfReadOnly}%
607       \noexpand\F{\noexpand\FHidden}%
608       \noexpand\Q{1}%
609       \noexpand\textSize{0}%
610       \noexpand\V{\ChampionMsg}\the\GrandPoohbahtoks%
611       ]{\GrandPoohbah}{\textwidth}{\the\GrandPoohbahHeight}}%
612     \temp
613
614     \leavevmode
615     \ifnum\NumberOfQuestions>0%
616     \ifnum\NumberOfCategories>0%
617     \multido{\i=1+1}{\NumberOfCategories}{\leavevmode
618       \vtop{\hsize=\CellWidth \centering
619         \the\everyCategoryHead\csname CategoryName\i\endcsname }}\%
620     \leavevmode
621     \if@jeopardyBgPicture\insertJeopardyPicture\fi
622     \finetuneCellWidth=\CellWidth
623     \advance\finetuneCellWidth by -\finetunelength
624     \finetuneCellHeight=\CellHeight
625     \vtop{%
626       \Multido{\i=1+1}{\NumberOfQuestions}%
627       {\edef\j{\i}%
628         \hbox{%
629           \if@jeopardyPicture\hbox to 0 pt{\hss$\j$ }\fi
630           \Multido{\i=1+1}{\NumberOfCategories}{%
631             \edef\temp{\noexpand\textField[%
632               \noexpand\Ff{\noexpand\FfReadOnly}%
633               \noexpand\S{S}%
634               \noexpand\Q{1}%
635               \the\Celltoks]{Field\j\i}{\finetuneCellWidth}{\finetuneCellHeight}}%
636             \fbox{\hbox to 0 pt{\temp\hss}%
637               \if@jeopardyPicture
638                 \pushButton[\S{B}%
639                   \A{\JS{kliknuti(\j,\i);}}]{Button\j\i}{\finetuneCellWidth}{\finetuneCellHeight}
640                 \else
641                 \pushButton[\S{B}\CA{\j00}%
642                   \A{\JS{kliknuti(\j,\i);}}]{Button\j\i}{\finetuneCellWidth}{\finetuneCellHeight}
643                 \fi%
644               }}\kern-1pt
645             }%
646     \fi %%% \ifnum\NumberOfQuestions>0%
647     \fi %%% \ifnum\NumberOfCategories>0%
648     \par

```

```

649 \AfterGameBoard
650 \if@jeopardyPicture
651 \smallskip\leavevmode
652 \hfill\pushButton[\A{\JS{hideall()}}\CA{Solution}]{Solution}{1.5cm}{12bp}
653 \else%not picture
654 \@makeScoreField
655 \fi
656 \end{center}%
657 \newpage
658 }%
659 \fi
660

```

The user may write his/her customizations into jeopardy.cfg.

```

661 \IfFileExists{jeopardy.cfg}{%
662 \typeout{-----}
663 \typeout{jeopardy: Configuration from the file jeopardy.cfg.}
664 \typeout{-----}
665 \input jeopardy.cfg}{}
666 \}

```

We define shorter navigation panel if pdfscreen.sty package is used.

```

667 \*cfg
668 \ifx\NavigationPanel\undefined\else
669 \def\NavigationPanel{\normalsfcodes%
670 \href{\@urlid}{\addButton{\buttonwidth}{\tiny\@Panelhomepagename}}\ \ \pfill
671 \hyperlink{GameBoard}{\addButton{\buttonwidth}{\tiny GameBoard}}\ \ \pfill
672 \Acrobatmenu{FullScreen}{\addButton{\buttonwidth}{\tiny\@Panelfullscreenname}}\ \ \pfill
673 \Acrobatmenu{Close}{\addButton{\buttonwidth}{\tiny\@Panelclosename}}\ \ \pfill
674 \Acrobatmenu{Quit}{\addButton{\buttonwidth}{\tiny\@Panelquitname}}\ \
675 }
676 \fi
677 \def\PDFSCR@Warning#1{}
678 \}

```