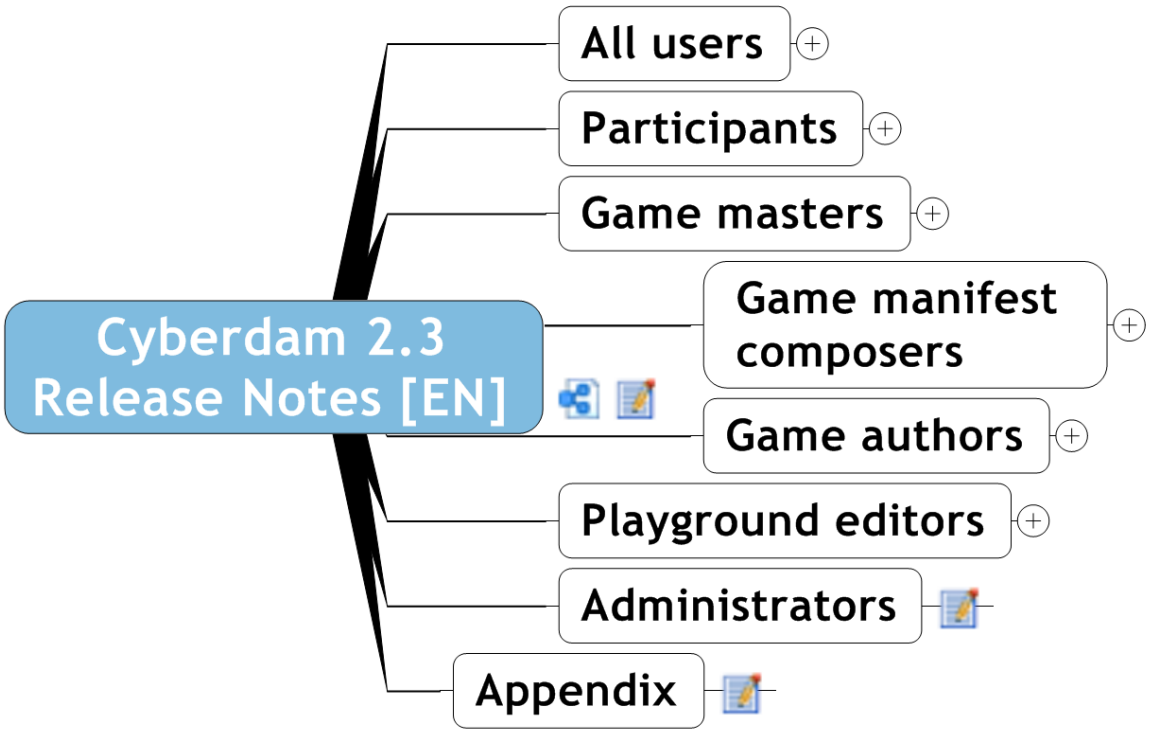
Stichting RechtenOnline

Project Leren in een Virtuele Wereld

Cyberdam 2.3 Release Notes [EN]



Draft 8 March 2010 - Pieter van der Hijden

Projectleiding:

Diny Peters
Pieter van der Hijden

Projectpartners:

E-Merge
Hogeschool Rotterdam
Hogeschool Utrecht
Strathclyde University Glasgow
TU-Delft
Universiteit Leiden
Stoas Learning
IJsfontein

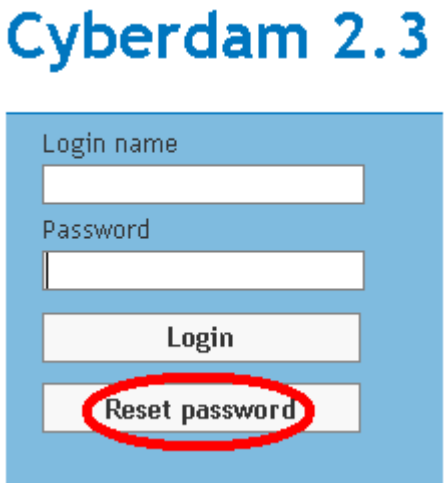
1 All users

1.1 Playgrounds for all

All users, even those who did not log in, can open a playground window that gives access to all public playgrounds (map, directory and background).

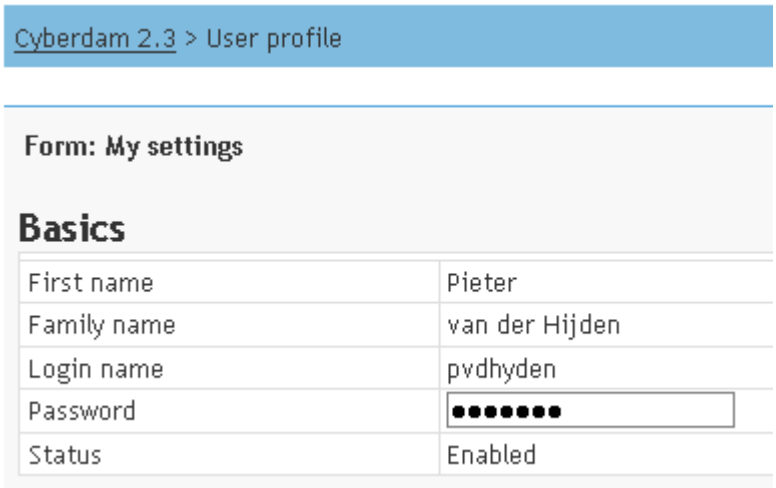
1.2 Changeable passwords

1.2.1 Reset your password



At the login page, users who forgot their password, can ask for a new one (to be sent by e-mail). Just fill in your login name and click on the Reset Password button.

1.2.2 Change your password



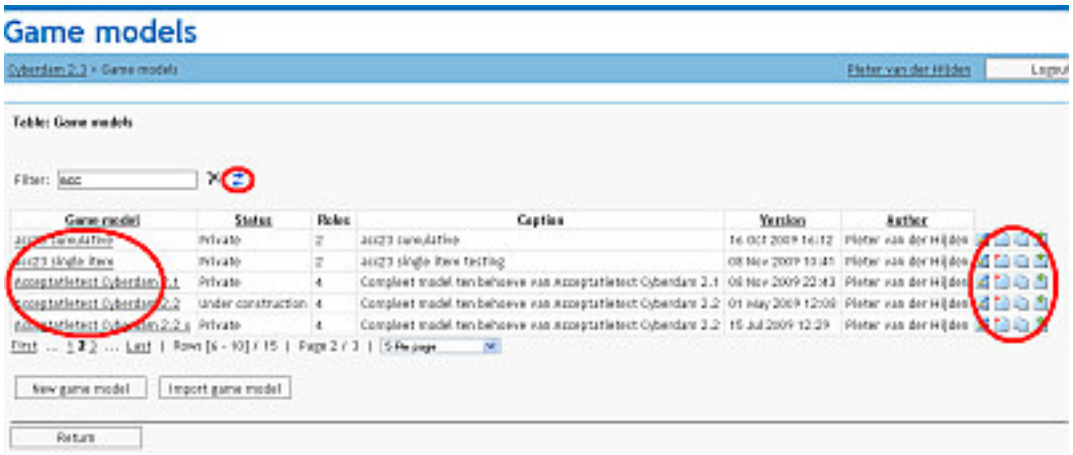
After logging in, users can change their passwords on their personal profile page; click on your name at the top right part of a page. Do not forget to save your change.

1.3 Improved user interface

1.3.1 Front page

The front page before login and the front page after login has a more consistent lay-out.

1.3.2 Tables

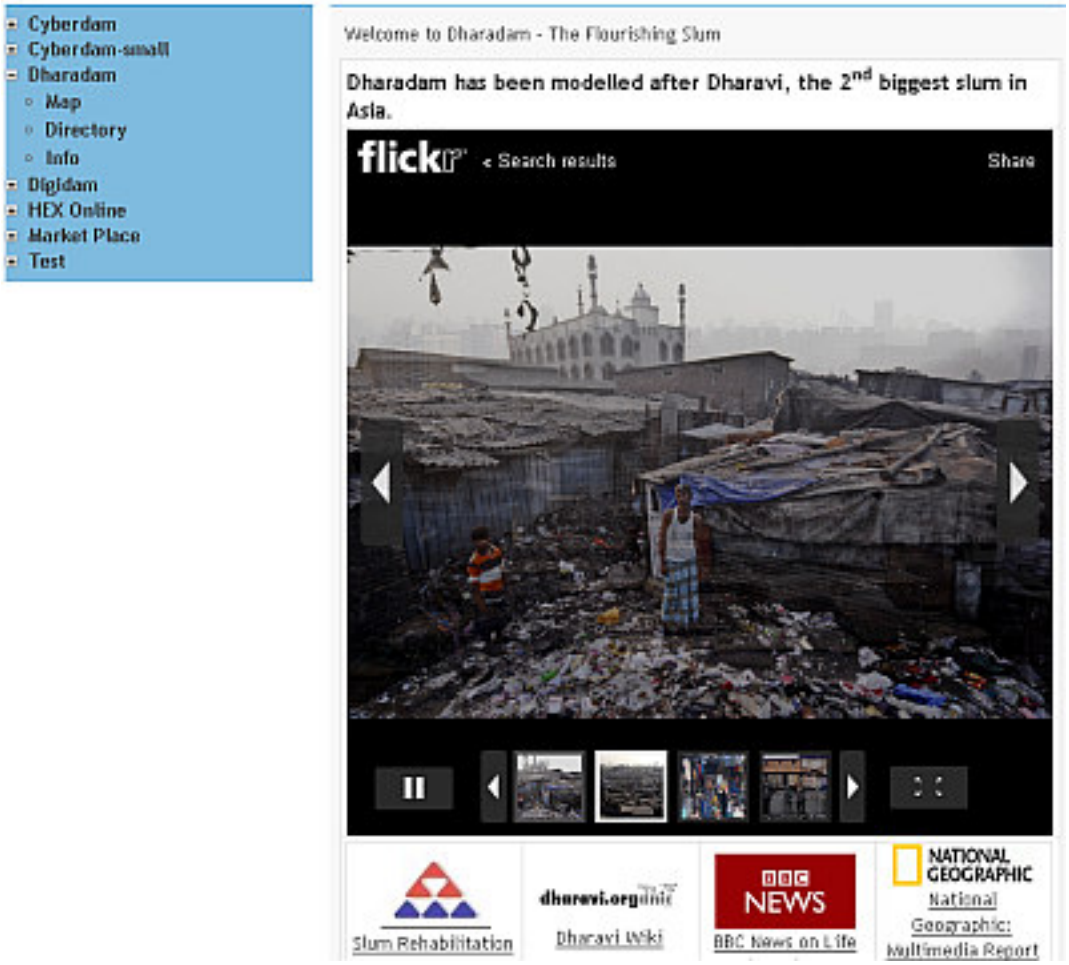


The usability and lay-out of all tables have been improved:

- The large button to refresh a table has been replaced with a small icon.
- The first field in a row is clickable now.
- Hyperlinks at the end of a row have been replaced by icons.

1.3.3 Multimedia

Dharadam Info



Already present from version 2.1, but apparently not discovered by users yet, are Cyberdam's multimedia options. Look for text entryfields that have a series of text formatting buttons at their top. Users may use these entryfields not only for well formatted texts, but also for entering multimedia or a mix of both.

How does it work:

- pictures - For pictures a special button is provided. Click the button and fill in the popup window with the pictures characteristics.
- slideshow - For slideshows, you need to find or create a slideshow on Internet and to embed some code in your entryfield to display it. The procedure to upload your own pictures and have them displayed as a slideshow in your entryfield is as follows:
    1. Store your pictures on a site like Flickr,
    2. Give them a common (and unique) keyword,
    3. Search for this keyword,
    4. Go to the corresponding slideshow,
    5. Copy the code to embed the slideshow in your entryfield,
    6. Go to the entryfield in Cyberdam where you want to embed the slideshow,
    7. Press the Code button,
    8. Paste the copied code,
    9. Press the code button again,

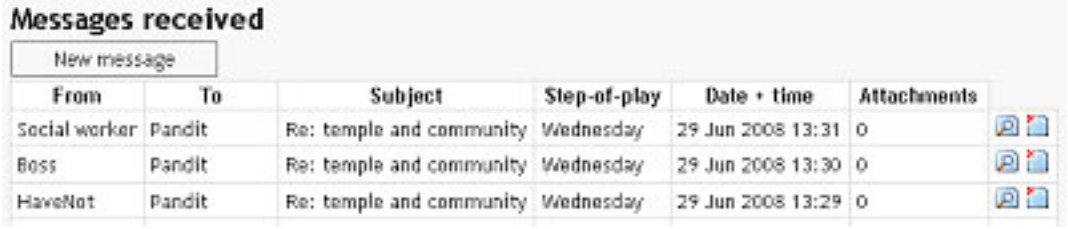
10. If OK, the save the new "text".

- videoclip :
    1. Store your videoclip on a site like YouTube,
    2. Go to the videoclip,
    3. Copy the code to embed the video in your entryfield
    4. Go to the entryfield in Cyberdam where you want to embed the videoclip,
    5. Press the Code button,
    6. Paste the copied code,
    7. Press the code button again,
    8. If OK, the save the new "text".

2 Participants

2.1 Improved user interface

2.1.1 Messages



Lists of received, sent and removed messages are cleaned up. In earlier version they listed the role name, playground name and object name for each addressee; now they list the role name only.

2.1.2 Multimedia

Note, that you can use multimedia (pictures, sound, video) wherever you can enter text, e.g. you can create messages that use multimedia.

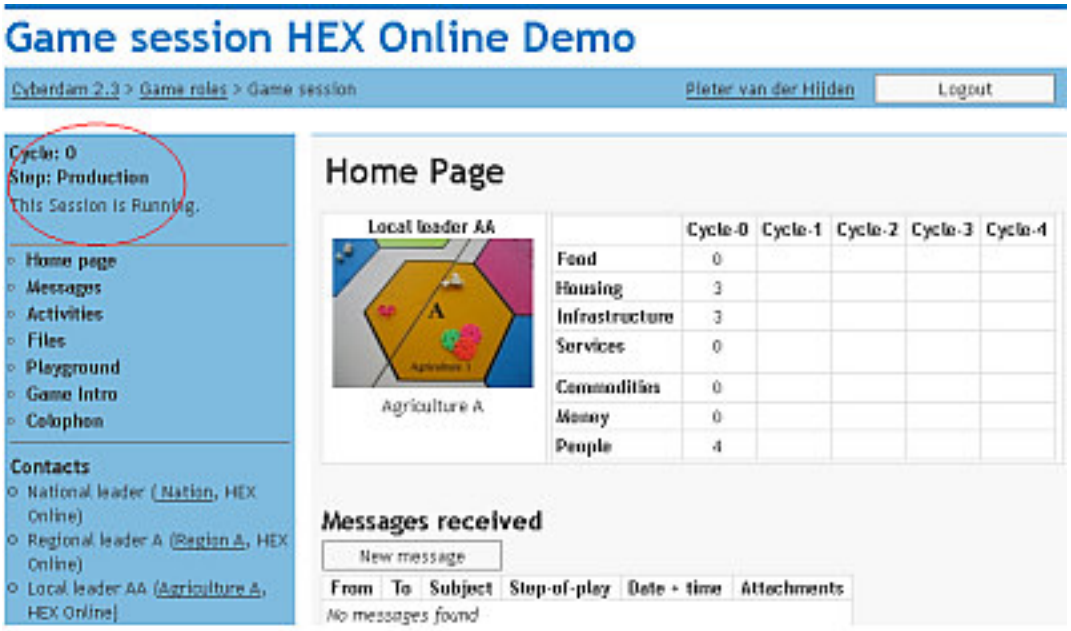
2.2 Activity types

As a participant to a game session you may be offered five different types of activities:

- input - to provide input to the system, e.g. a decision form (earlier called "form activity")
- output - to get output from the system, e.g. an event (new, see below)
- upload - to upload a file to your online directory (earlier called "file upload activity")
- message - to send a message to another role
- step - to change the step-of-play (earlier called "progress activity")

2.3 More dynamic session home page

2.3.1 Status display



On the game session pages, the status display (1) no longer has a fixed format. The game model author can specify a different format (and/or go back to the well- known default). It may include variables and therefore have a dynamic content.

2.3.2 Heads-up display (HUD)

On the game session home page, at the top of the right side, a so- called heads-up display (2) has been added. The game author can specify its content. It may include variables and therefore have a dynamic content.

2.4 System dynamics added

Until now, once you entered a certain step-of-play the set of activities offered to you was fixed. Now, the composition of the set of activities may vary during the step-of-play. In fact they may be switched on (enabled) and off (disabled) by the system.

In the case you are completing an activity that in the mean time has been switched off (disabled), upon completion, the system will tell you that the activity was no longer available.

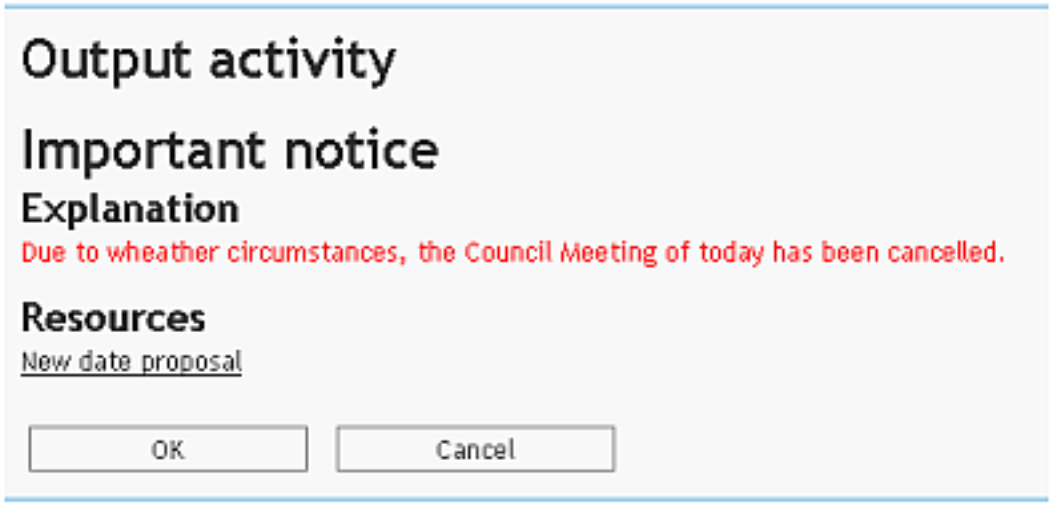
2.5 New activity types

2.5.1 Input activity

It was already there in Cyberdam 2.2, for completeness we repeat it here: the input activity. When you open it, it will display a form you may fill in.

(((example of an input activity)))

2.5.2 Output activity



A new activity type has been added: the output activity. When you open it, it will display some instructions and/or variable information to you. You only can Confirm or Cancel the receipt of this information.

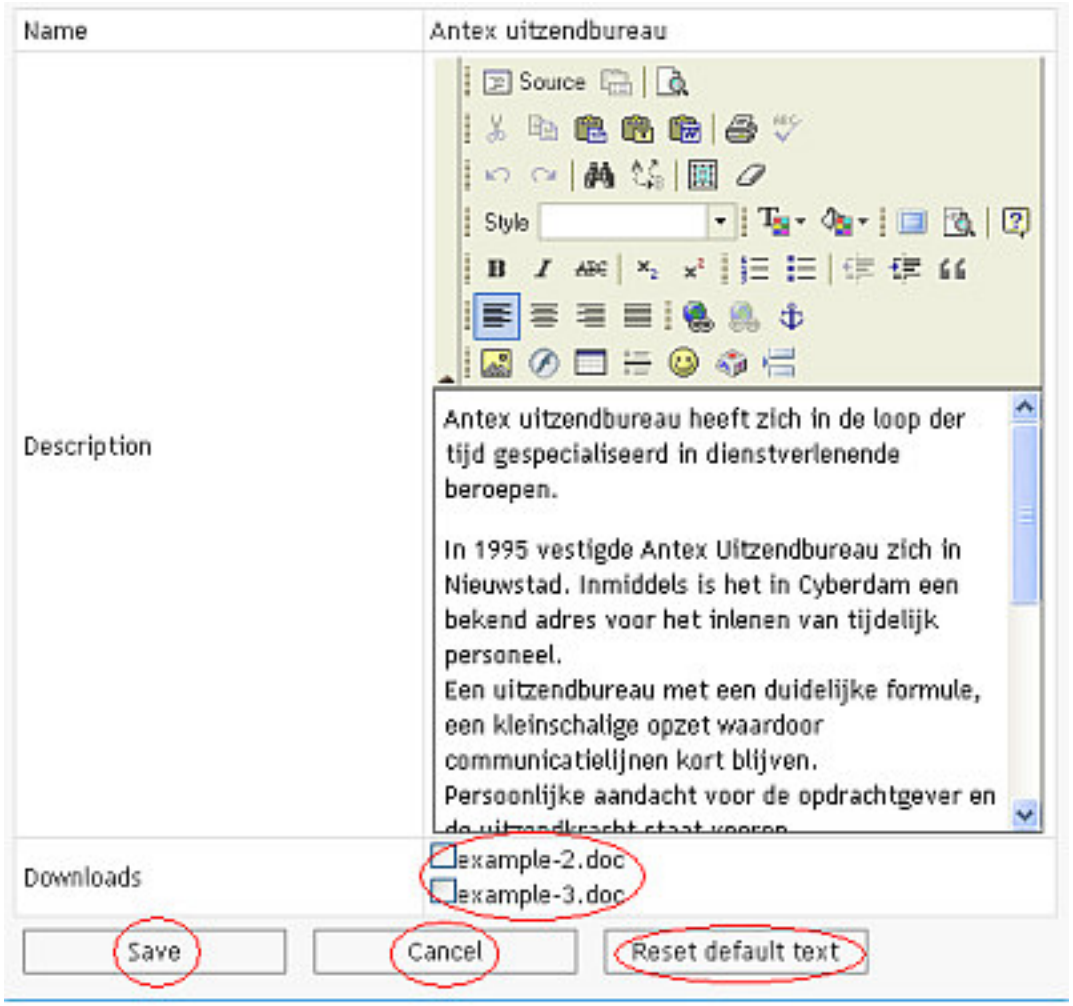
2.6 More efficient saving of activity attachments

Activity instructions may be accompanied by resources (attachments). However, once the activity was completed, these resources were no longer accessible. Now, the activity page contains an "Show history" button to list the completed activities and to give you access to the accompanying resources (see below).

2.7 Changeable playground object page

Within the context of a game session, you may change the description of the playground object related to your role and add files to it:

- When you visit the playground object page related to your role, the page will display an edit button. Clicking it opens the description field for editing and a files list for selecting and deselecting files.

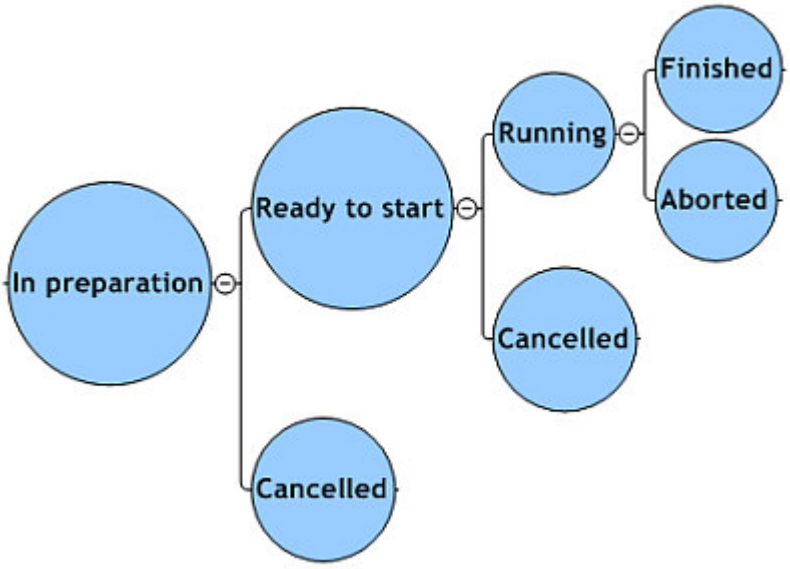


- Initially the description field is filled with the default rich text entered by the playground editor. You can change this text (e.g. adding multimedia) and click either save, cancel or reset-to- default. When more roles are linked to the same playground object they all can edit this text.
- The file list contains the names of the files you have uploaded during this session. You may select/deselect the files in this list, thus making them visible to all the participants of this session.

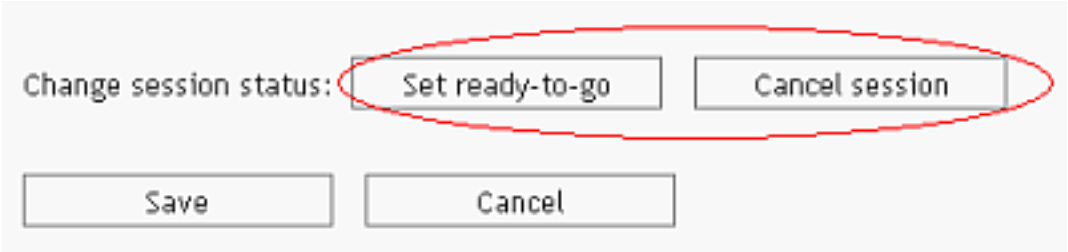
- When you visit a playground object page not related to your own role, the page displays the (modified) description and eventually some file names; these files can be opened.
- When somebody visits the playground object page out of the context of a game session, the playground object page shows the default text (as entered during playground editing) and no hyperlinks to any files, i.ethe same as in earlier versions of Cyberdam.

3 Game masters

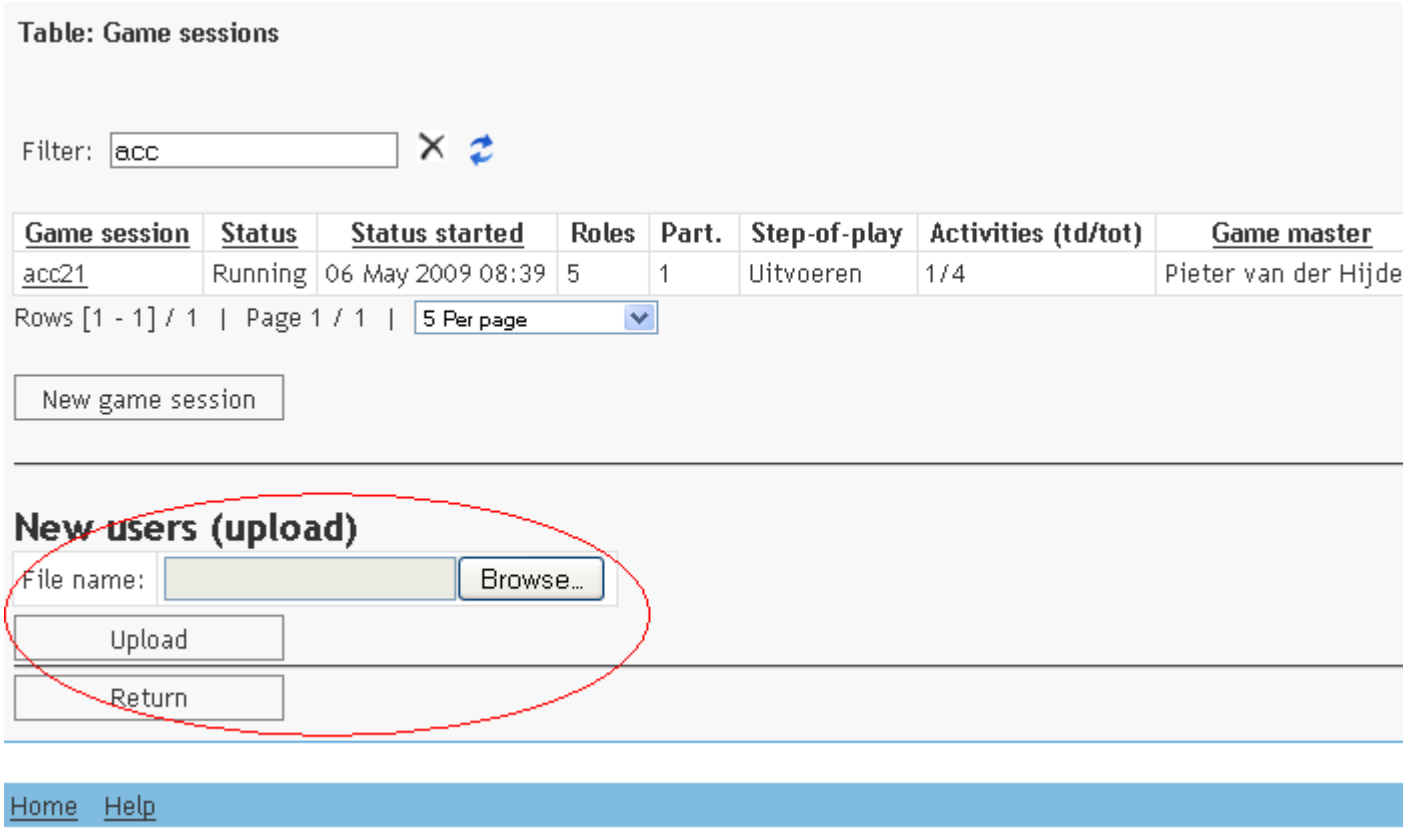
3.1 Improved user interface



The game session take-off via subsequent stages (preparation, ready-to-start, running) can be handled from one screen. In earlier versions you had to save the new status and re-open the session various times.



3.2 Efficient user registration

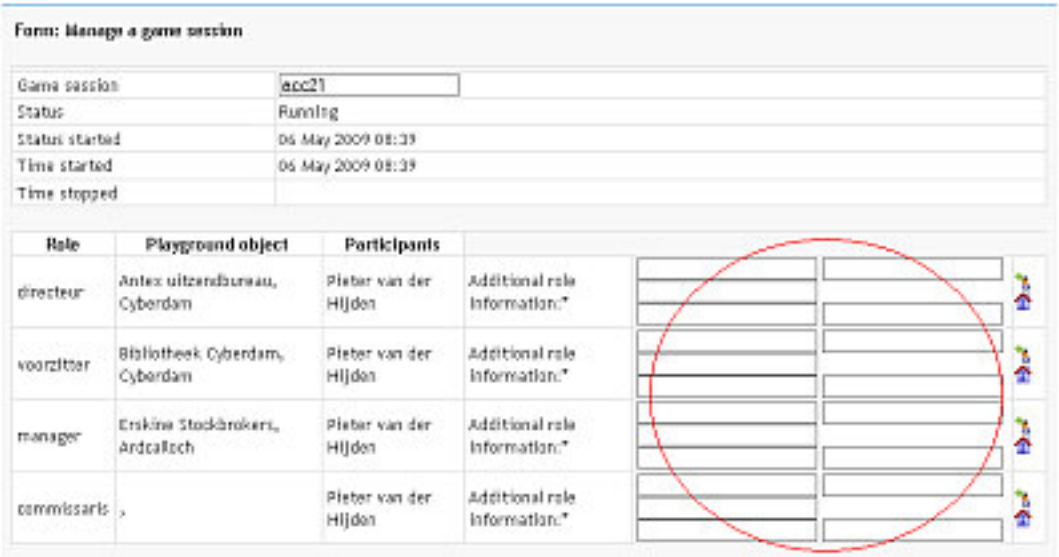


A game master now can upload a batch of users.

The procedure is as follows:

- Collect the user data: login-name, first name, last name and e-mail address.
- Prepare these data as a CSV file (a Comma Separated Value text file). Each line in the file should contain the data for one user only: login-name, first- name, last-name, e-mail-address. You can prepare such a file manually, e.g. by using the MS Notepad program, or more- or-less automatically, e.g. via MS- Excel that has the option to save your data as CSV file.
- Go to the Game Sessions page.
- Upload the file. Please, note eventual error messages the system produces.
- The system creates a new user group with your login name + date+time as group name. It creates user accounts as specified in the uploaded file while making each of them a member of the new user group.
- The system sends a welcome message with login name and password to each new user.

3.3 More integration with collaborative tools



To facilitate linking the Cyberdam sessions to external services like chat rooms, the game master can specify up to five parameters for each role (and eventual change them during the session). In fact through the session control page, the game master has direct access to five role dependent variables. Their values might be used in instructions to activities, in the status display and in the heads- up display. There they could be used as a hyperlink to the external services.

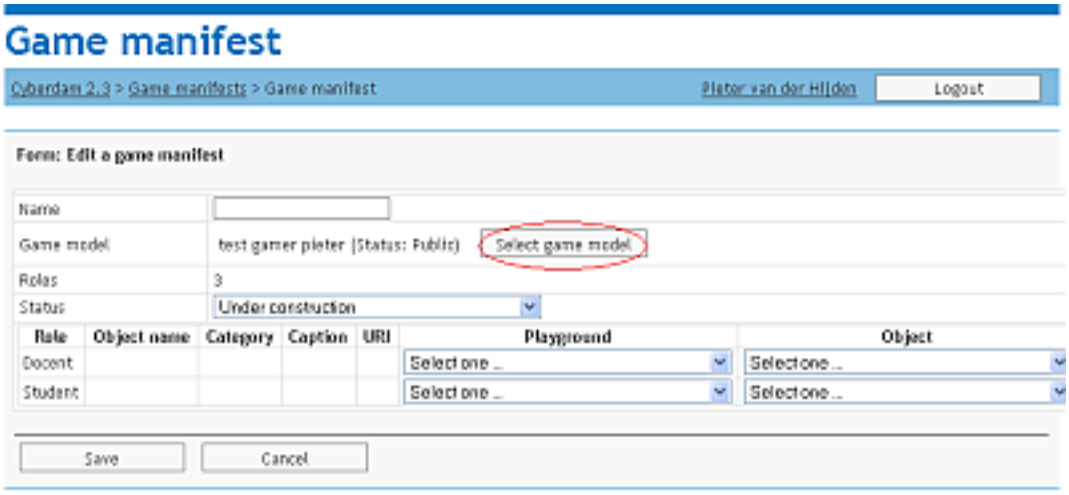
Although intended for linking to external services, the mechanism in fact is more powerful. It offers the game master five switches for each role that may have an impact on the game session.

Examples of alternative use:

- a variable indicates the experience level of a role, depending on its value, the system offers beginners, intermediate or advanced activities,
- a variable to set some initial value, like the initial budget of each role,
- a variable to record a certain score manually.

4 Game manifest composers

4.1 Improved user interface



- Using the Apply button after a change is no longer necessary. The Apply button has been removed therefore.
- A new button Change Game model enables you to use a different model for your manifest.

5 Game authors

5.1 Improved user interface

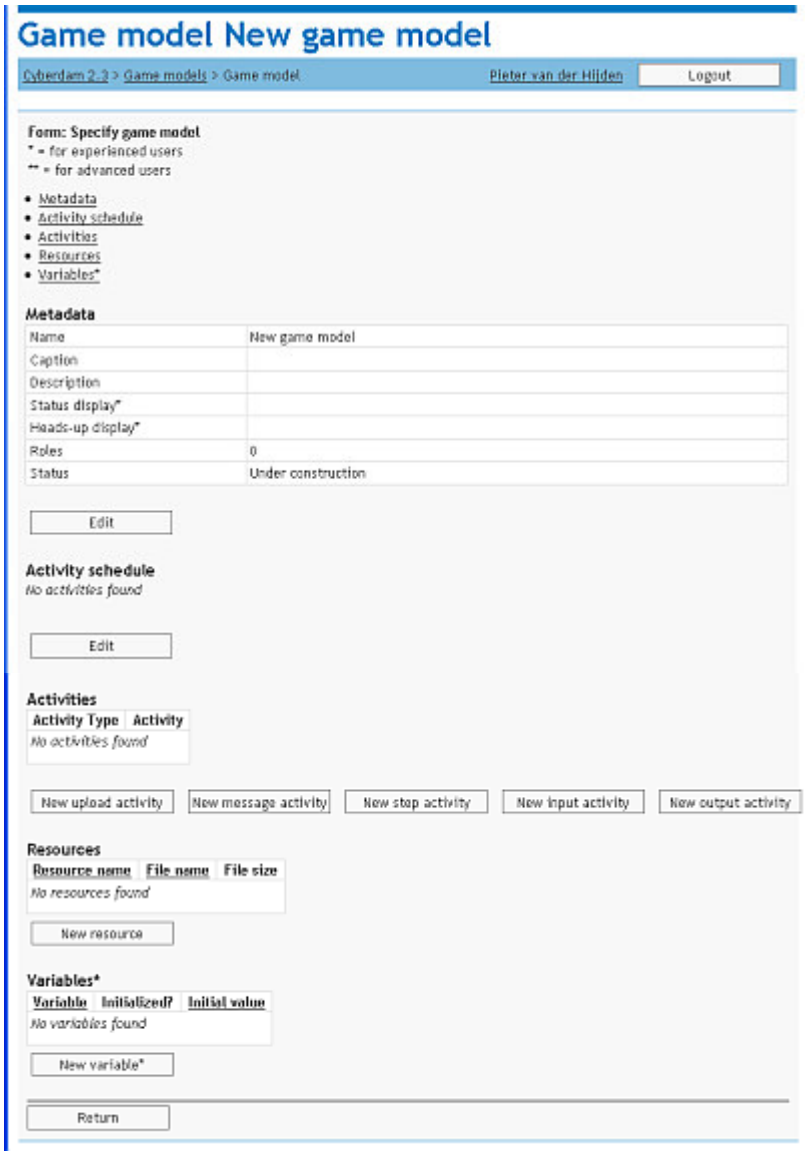
Note, that you can use multimedia (pictures, sound, video) wherever you can enter text, e.g. you can create instructions to activities and default texts for messages that use multimedia. The same is true for the description of the game model itself.

5.2 Three levels of expertise

- From now we distinguish three levels of expertise for game authors:
  - regular - roughly spoken an author familiar with roles, steps and activities,
  - experienced - for authors who use variables to store values that might change over time (on your screen, entry fields for experienced authors are marked by "*"),
  - advanced - for authors who use scripts and conditions (on your screen, entry fields are marked by "**").
- The rest of this chapter will be subdivided according to the three levels of expertise.

5.3 Expertise level: Regular

5.3.1 Game model overview

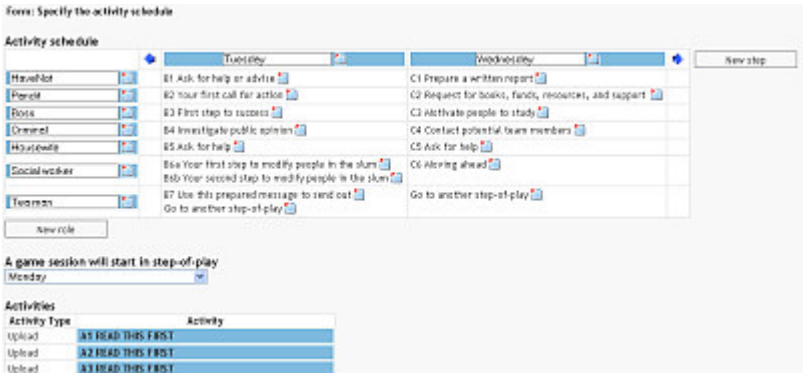


Building a game model is no longer a wizard-like process. From now, when you open a game model for editing, you will see a summary of the model metadata, activity schedule, activities, resources and variables (for experienced users). Via edit buttons you have access to various detail pages for each of these elements.

You use the activity schedule to define roles and steps-of- play.

You use this game model overview to add variables to roles (experienced users) and to add scripts (advanced users) to steps.

5.3.2 Activity schedule



The activity matrix has columns for each step-of-play and rows for each role. The cells may be filled with one or more activities. When you start a new game model the matrix is empty; it has no roles and no steps-of-play yet.

You use the activity schedule to define roles, to rename them, to change their order and to delete them eventually. The system displays all roles all the time as rows of the matrix.

You use the activity schedule also to define steps-of-play, to rename them, to change their order and to delete them eventually. The system displays the steps-of-play as columns of the matrix, only a few at a time. Scrolling the columns is possible.

Once you have defined some activities (on the game model overview form), the system lists them in alphabetical order right below the activity matrix. You may drag these activities and drop them into the cells of the activity matrix. During a game session they will be offered to that role during the indicated step-of-play.

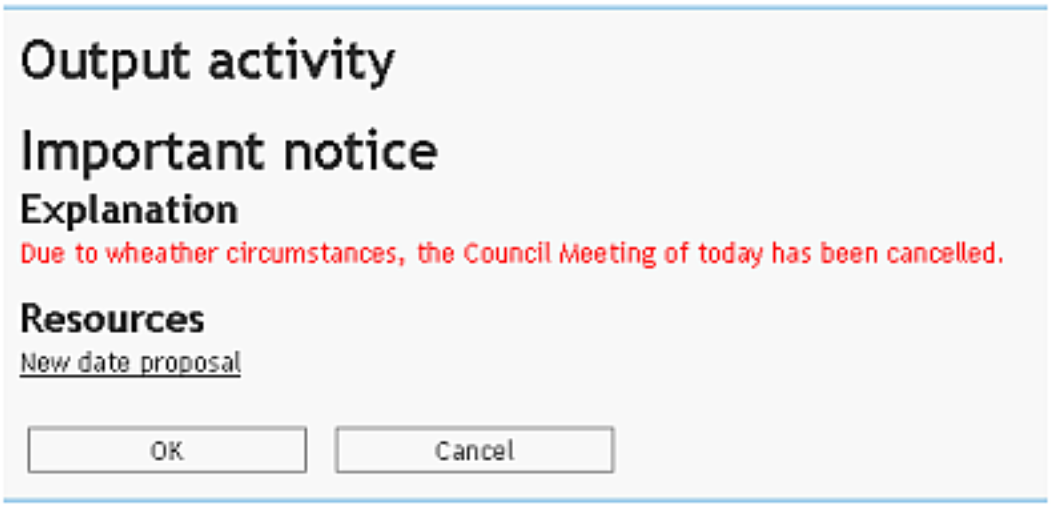
5.3.3 Activities

As a game author, you may offer five different types of activities to your game session participants:

- upload - to upload a file to your online directory (earlier called "file upload" activity)
- message - to send a message to another role
- step - to change the step-of-play (earlier called "progress activity")
- input - to provide input to the system, e.g. a decision form (earlier called "form activity").
- output - to get output from the system, e.g. an event (new, see below).

The input activity requires expertise level "Experienced".

5.3.4 Output activity



The picture shows how a participant will see the outlook activity.

A new activity type is added, the output activity type. It displays some system output (e.g. events) to the participant, i.e. a rich text, multimedia may be included. It has all the common characteristics of the other activity types. The instruction is simply followed by OK or Cancel. Pressing OK, implies the activity is considered to be completed.

((screenshot: how to enter a new output activity))

5.4 Expertise level: Experienced

Please, read the section on "Expertise level: Regular" first.

The functions belonging to the Experienced level have been available since Cyberdam 2.2. As they laid the ground work for Advanced functions introduced in Cyberdam 2.3, we include their description here for easy reference.

5.4.1 Variables

A variable is a named piece of memory where a value can be stored.

Variables give game sessions a memory for scores, amount of capital, number of inhabitants, pollution level, annual production and whatever "stock" a game author wants to record (from moment to moment) and to display to the participants when appropriate.

A game author may declare variables at game model level and at game role level. The game author can give a variable an initial value, i.e. a piece of text (a number, a sentence, some HTML-code):

- Names of variables start with a letter, an underscore or a dollar sign. Their maximum length is 255 characters. Uppercase or lowercase makes a difference. Dots are allowed in predefined variables only.
- The value of a variable is either undefined or a string of characters. A string value may contain HTML code. Note that the correctness of eventual HTML code will not be checked by the system.

Note-1: For different roles the same names of variables may be used. Example: all roles may have a variable called level and a variable called score. For each role they may contain different values. These values only will be used on screens intended for that role.

Note-2: The system has a set of predefined variables. also the system administrator can define such "global" variables. More details can be found in the appendix.

(((Screenshot of filled in model variables: food, housing, infrastructure, services, commodities, money, people.)))

(((Screenshot with filled in role variables: altname, score)))

5.4.2 Changing values

As a game author you give the variables an initial value. During the game session these values can change in two ways:

- The participant completes an input activity: During an input activity, the participant fills in a form. In fact the fields on the form refer to certain variables. The values entered will become the new values of these variables.
- The system runs a script: At the start of a game session, at the start of a new step-of-play and at the end of an activity the system may run a script that can change the value of variables as well. See Expertise level "Advanced".

5.4.3 Displaying values

You can use variables to give certain texts in your game a variable character. In fact each instruction to an activity can be dynamic in nature. Enter the name of a variable in that text (surrounded by "[%" and "%]") and at run time the value of that variable will be displayed to the

participants. This could imply the substitution of a single word or number. However, a value also maybe a piece of HTML-code, thus offering quite powerful options for dynamic texts and multimedia.

((Screenshot: instruction text with variable))

((Screenshot: instruction text with value))

Also the status display (see below) and the heads-up display (see below) can be made dynamic in the same way.

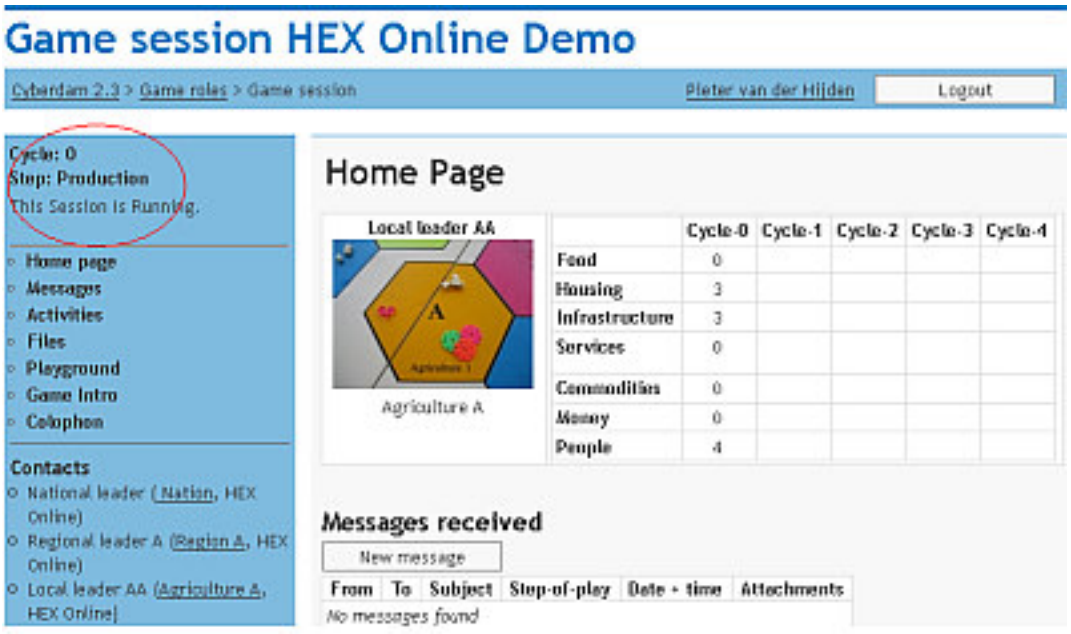
Example:

- Suppose the game author defined the variable budget at game level and gives it the value 10000.
- In an explanation field of an activity, the game author enters the text: "The actual budget is [%budget%] USD."
- During the game session, the participant will see the text: "The actual budget is 10000 USD." If for whatever reason, the value of budget has changed in the mean time, the participant will see the new value.

Another example shows the power of using the same variable name for each of the roles.

- Suppose the game author defined the variable manpower at role level for each of the roles and gives them each a different initial value, e.g. 100, 200, 300.
- In an explanation field of an activity, the game author enters the text: "Your manpower level at this moment is [%manpower%]."
- During the game session, the participant will see the text: "Your manpower level at this moment is ...." The dots contain the number that is appropriate for the role to which the activity was assigned, e.g. 100, 200, 300.

5.4.4 Status display



The picture above shows the session home page as seen by a participant.

On the game session pages, the status display no longer has a fixed format. The game model author can specify a different format (and/or go back to the well- known default). It may include variables and therefore have a dynamic content. Especially in this case some of the predefined variables might be welcome (see appendix).

You can change (or reset) the status display via the game model metadata form.

((screenshot on how to change the status display))

5.4.5 Heads-up display (HUD)

The picture shows the session home page as seen by a participant.

On the game session home page, at the top of the right side, a so- called heads-up display has been added. The game author can specify its content. It may include variables and therefore have a dynamic content.

In the picture, the heads-up display contains a image that is unique for every role. This was possible by the use of variables:

- Create a variable for every role, e.g. Code.
- Give simple alternative names for these roles as values to these variables, e.g. role Local Leader Agriculture A has as Code simply "AA".
- Prepare a set of images, all with the same prefix, followed by the Code; e.g. the image for code "AA" is FigureAA.jpg.
- Upload the images to the same web location.
- Insert one of these images into the heads-up display, e.g. FigureAA.jpg.
- In the HTML code of the heads-up display, replace the "AA" in the file name with "[%Code%]", like Figure[%Code%].jpg. During the game session for each role a different image will be displayed then.

You can change (or reset) the heads-up display via the game model metadata form.

((screenshot on how to change the heads-up display))

5.4.6 Input activity

It was already there in Cyberdam 2.2, for completeness we describe it here again: the input activity. This activity displays a form that the participant has to fill in. In fact, the participant changes the values of certain variables by this form.

(((screenshot: how to create an input activity)))

By allocating an input activity to a teacher role, the teacher may be given the opportunity to enter scores, warning messages, etc. to be displayed to the participants via the "[%...%]" mechanism explained in the previous paragraph.

5.5 Expertise level: Advanced

Please, read the section on "Expertise level: Experienced" first.

The functions belonging to the Advanced level are new in Cyberdam 2.3. In short:

You can add system dynamics to your game models. If you do not need these new features, you can simply ignore them. System dynamics implies:

- activities are conditional, they have the status enabled (default) or disabled,
- activities may have an end-of-activity script built-in (default is no script). A script is a set of calculations that can change the values of variables and move the session to another step-of-play. Upon completion of an activity, the system executes that script.
- the game model as such may have a begin-of-session script (default is no script).
- a step-of-play may have a begin-of-step script (default is no script).

5.5.1 Variables

Variables serve various purposes:

- As seen before, variables give game sessions a memory for whatever "stock" a game author wants to record (from moment to moment) and to display to the participants when appropriate.
- For advanced game authors variables make dynamic feedback in game sessions possible (as their values can be fed to a calculation script and the outcomes of the calculation can be fed back as values of variables again);
- Also variables make dynamic game session progress possible and make game content variable (e.g. certain values may trigger a move of the game-session to another step-of-play, or change a condition to show or hide an activity).

The value of a variable is either undefined or a string of characters. These values are persistent during a game session.

Scripts can read the values of variables and store them in their own internal script variables. The latter exist only during the execution of the script. Any results have to be written to the "regular" variables before the script ends therefore.

5.5.2 Scripts

A script is a set of calculations that can change the values of user defined variables and move the session to another step- of- play. In fact we use server side Javascript; see Javascript, the definitive guide; D. Flanagan; O'Reilly Media Inc., 2006.

A game model may contain:

- A single begin-of-session script; this script is run at the start of a session; the game author enters this script in the game model metadata section.
- A begin-of-step script for each step-of-play; this script is run at the start of that step-of-play; the game author enters such a script via Edit Step-Of-Play.
- A condition script for each activity; this script is run at various moments (see next paragraph), it returns either true or false which implies the activity is enabled or not. The game author enters the condition on the activity specification page.
- An end-of-activity script for each activity; this script is run when the activity has been completed (to be more precise: when an open activity is terminated: not by Cancel, while still Enabled and the current step-of-play = original step- of- play). The game author enters this script on the activity specification page.

((screenshot on how to define a script at game model level, step-of-play level, activity level))

((screen shot on how to see the results of the script))

A script:

- may define script variables (they will only "exist" during the execution of the script); these variables may have a numerical, a character string or a Boolean (true/false) value; examples result=100, level="beginner", condition=true;
- may assign a value to a variable; this value may be the outcome of an expression; example result=result+10;
- may refer to the values of all other types of variables; examples variables.score=result (the model variable called score receives as new value the value of script variable result);
- can execute calculations (with numbers, strings and Booleans), including conditions, selections, loops and functions (self created or from a standard library, e.g. with string functions and a random number generator) and change the values of user defined variables;
- can change the current step-of-the-play, e.g. to move to step-of-play "XXX" use the script function: cyberdam.setNextStepOfPlay ("XXX").
- see the appendix on how to refer to predefined variables.

5.5.3 Conditions

Activities become conditional, i.e. whether an activity is offered to the participant not only depends on its presence in the appropriate cell of the activity schedule, but also on the outcome of a script (true implies enabled, false implies disabled; default is no condition = no script = activity enabled). This condition is tested, i.e. this script is run at various moments:

- When the session home page or the activities page are generated: if the condition script returns false, the activity will not be on the activity list at all.
- When the activity is opened (as the enabled-status of the activity could have changed in the mean time): if the condition script returns false (or original step-of- play is no longer current step-of- play), the system generates an error message ("activity no longer available") and the page is refreshed (now no longer listing that activity).
- When an open activity is terminated (not by Cancel): if the condition script returns false (or the original step-of- play is no longer current step-of-play, or activity has been completed by another participant already), the system generates an error message ("activity no longer available") and returns to the activities page (now no longer listing that activity).

6 Playground editors

6.1 Improved user interface

Once you open a playground in the list of playgrounds, you will see its general data (metadata) and the list of objects on a single page.

Note, that you can use multimedia (pictures, sound, video) wherever you can enter text, e.g. you can include multimedia in the description of a playground and in the descriptions of its objects.

((screenshot))

6.2 Second default playgrond map

A second default playground map has been added to the application, i.e. Digidam, based on the same legacy Flash module as the Cyberdam playground map.

((screenshot))

6.3 Simple generic playground map mechanism

You can create your own map based on a single image file (JPG). It is a simplified version of the Cyberdam (and Digidam) legacy maps; it does not offer panning and zooming and separate symbols for each object category. However, it does have a directory of objects which become clickable icones on the map, complete with the little popup window that brings you back to the object page.

You are free to choose the size of your map, although 500*500 pixels gives the best fit with the pop-up playground window. The top left corner has coordinates 0,0. The unity used is pixel.

((screenshot Dharadam))

7 Administrators

Note, that you can use multimedia (pictures, sound, video) wherever you can enter text, e.g. you can include multimedia in all system texts.

8 Appendix

For game authors with expertise level "Advanced".

Predefined variables and how to use them in scripts.

Concept	Variable name	How to use in scripts
Actual date and time	system.datetime	variables.system.datetime
Dialogue text (language dependant) for certain multi-language KEY	system.message.*	cyberdam.getmessage("*")
Variable defined by system administrator	example	variables.example
Name of the game model	model.name	variables.model.name
Caption (long title) of the game model	model.caption	variables.model.caption
Name of the role	role.name	variables.role.name
Name of the game session	system.session.name	variables.system.session.name
Actual session status, e.g. Running	system.session.status	variables.system.session.status
Name of the actial step- of-play	system.stepOfPlay.name	variables.system.stepOfPlay.name
Name of the playground for this role	system.playground.name	variables.system.playground.name
Name of the playground object for this role	system.playgroundObject.name	variables.system.playgroundObject.name
Game master parameter for this role	system.role.value1…5	variables.system.role.value1