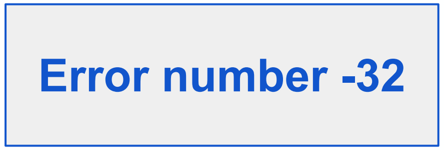
Eye Spy Heuristics Bingo

Human Computer Interaction

Learning Activity

Eye Spy Heuristic Bingo - Activity Instructions



Error number -32

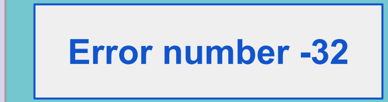
Learning outcome: Learners will be able to develop an awareness for design heuristics.

This activity is intended for use after a lesson on developing an understanding for Jakob Nielsen's Ten Usability Heuristics.

1. Share an interface example with students like the error message to the right.
2. Have a look at the Eye Spy Heuristic Bingo board and identify the best fit for this interface example. The example can be either negative (causes problems because it goes against the heuristic) or positive (is a clear example of the heuristic in action).
3. Place a copy of the image in the appropriate square, like we've done below.
4. Over a period of time as students use interfaces have them collect heuristic examples.

Note: Each example may fit into more than one heuristic square; students can select which heuristic they wish to place the image example.
** It is recommended that for privacy, security and to avoid other ethical issues that this activity is completed with paper or in a non-public manner.*

Eye Spy Heuristic Bingo

Recognition rather than recall Minimize the user's memory load by making elements, actions, and options visible. Avoid making users remember information.	**Error prevention** Good error messages are important, but the best designs prevent problems from occurring in the first place.	**Help & documentation** It's best if the design doesn't need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.	**Match between system & real world** The design should speak the user's language. Use words, phrases, and concepts **familiar to the user**, rather than internal jargon.  Error number -32	**Flexibility and Efficiency of Use** Shortcuts - hidden from novice users - may **speed up the interaction** for the expert user.
Consistency & standards Users should not have to wonder whether different words, situations, or actions mean the same thing. **Follow platform conventions.**	**Visibility of System Status** Designs should **keep users informed** about what is going on, through appropriate, timely feedback.	**User control & freedom** Users often perform actions by mistake. They **need a clearly marked 'emergency exit'** to leave the unwanted state.	**Aesthetic & minimalist design** Interface should not contain information which is irrelevant every extra unit of information in an interface **competes** with the relevant units of information.	**Visibility of System Status** Designs should **keep users informed** about what is going on, through appropriate, timely feedback.
Flexibility and Efficiency of Use Shortcuts - hidden from novice users - may **speed up the interaction** for the expert user.	**Recognition rather than recall** Minimize the user's memory load by making elements, actions, and options visible. Avoid making users remember information.	**Help & documentation** It's best if the design doesn't need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.	**Recognize, diagnose, & recover from errors** Error messages should be expressed in **plain language** (no error codes), precisely indicate the problem, and constructively suggest a solution.	**Consistency & standards** Users should not have to wonder whether different words, situations, or actions mean the same thing. **Follow platform conventions.**

Eye Spy Heuristic Bingo

Recognition rather than recall Minimize the user's memory load by making elements, actions, and options visible. Avoid making users remember information.	**Error prevention** Good error messages are important, but the best designs prevent problems from occurring in the first place.	**Help & documentation** It's best if the design doesn't need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.	**Match between system & real world** The design should speak the user's language. Use words, phrases, and concepts **familiar to the user**, rather than internal jargon.	**Flexibility and Efficiency of Use** Shortcuts - hidden from novice users - may **speed up the interaction** for the expert user.
Consistency & standards Users should not have to wonder whether different words, situations, or actions mean the same thing. **Follow platform conventions.**	**Visibility of System Status** Designs should **keep users informed** about what is going on, through appropriate, timely feedback.	**User control & freedom** Users often perform actions by mistake. They **need a clearly marked 'emergency exit'** to leave the unwanted state.	**Aesthetic & minimalist design** Interface should not contain information which is irrelevant every extra unit of information in an interface **competes** with the relevant units of information.	**Visibility of System Status** Designs should **keep users informed** about what is going on, through appropriate, timely feedback.
Flexibility and Efficiency of Use Shortcuts - hidden from novice users - may **speed up the interaction** for the expert user.	**Recognition rather than recall** Minimize the user's memory load by making elements, actions, and options visible. Avoid making users remember information.	**Help & documentation** It's best if the design doesn't need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.	**Recognize, diagnose, & recover from errors** Error messages should be expressed in **plain language** (no error codes), precisely indicate the problem, and constructively suggest a solution.	**Consistency & standards** Users should not have to wonder whether different words, situations, or actions mean the same thing. **Follow platform conventions.**
Aesthetic & minimalist design Interface should not contain information which is irrelevant every extra unit of information in an interface **competes** with the relevant units of information.	**Error prevention** Good error messages are important, but the best designs prevent problems from occurring in the first place.	**Recognize, diagnose, & recover from errors** Error messages should be expressed in **plain language** (no error codes), precisely indicate the problem, and constructively suggest a solution.	**Match between system & real world** The design should speak the user's language. Use words, phrases, and concepts **familiar to the user**, rather than internal jargon.	**User control & freedom** Users often perform actions by mistake. They **need a clearly marked 'emergency exit'** to leave the unwanted state.