2D & 3D UIs: What’s the Same?

- Users/tasks (sometimes)
- Design/evaluation principles (sometimes)
  - E.g., high-level heuristics
    https://www.nngroup.com/articles/ten-usability-heuristics/
- Displays/devices (sometimes)
  - “Desktop” / “laptop” / “palmtop” 3D UIs using classical 2D interaction devices
- Development environments (sometimes)
  - Use of “classical” textual languages / IDEs
2D & 3D UIs: What’s Different?

- Overall relative lack of maturity/standardization
  - But, this is changing: WebGL (http://www.khronos.org/webgl/), X3D (www.web3d.org), AR standards efforts (http://arstandards.org)
- Users/tasks
- Sensory modalities
  - More than just sight and sound
- Interaction techniques
  - Perceiving, selecting, manipulating, navigating in 3D
- Displays/devices
  - Stereo, 6DOF head/body-tracked, wearable, vision-based,…
- Evaluation criteria
  - Realism, presence, immersion, simulator-induced sickness (VR sickness, cybersickness)
- Development environments
  - Design of 3D assets and interaction

(Revisiting) Ten Usability Heuristics J. Nielsen
https://www.nngroup.com/articles/ten-usability-heuristics/

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation
(Revisiting) Ten Usability Heuristics J. Nielsen
https://www.nngroup.com/articles/ten-usability-heuristics/

Visibility of system status
The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world
The system should speak the users’ language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom
Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards
Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention
Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Recognition rather than recall
Minimize the user’s memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
(Revisiting) Ten Usability Heuristics J. Nielsen
https://www.nngroup.com/articles/ten-usability-heuristics/

Flexibility and efficiency of use
Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design
Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors
Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation
Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.