Current Metrics

1. Efficiency
2. Operability
3. Productivity
4. Effectiveness
5. Understandability
6. Learnability
7. Satisfaction
8. Attractiveness

Examples for requirements (Beavan 14, 15):

"all data entry clerks will be able to complete the task with at least 95% accuracy in under 10 minutes"

- ToT (uncontestable effort measure)
- Definition of user (experience level)
- Accuracy

"the mean score on the SUMI scale will be greater than 50" More information on quality in use requirements

Slide 4 (measuring usability)

Minimum effort – effort expended by an "expert" in completing a set of tasks.

Expected effort – effort expended by an experienced user

A "productive user" should expend effort at a level of <= 1.2 expert effort

The expert (level) Minimum effort level can be determined by measuring effort expanded by highly experienced users. Often, the developers can define the minimum level. Sometimes it can be mathematically inferred from curve (via the asymptote).

Emerson Control Studio - engineers design a system that simulates the control aspects of a process such as Oil Refinery. Used to enable (machine) control of the process. Alternatively enables a human operator to control the process.

– system A uses an old interface format of MS office. System B uses the Ribon a newer one

“Tool for measuring usability – learnability and operability - as a function of effort”:

System A. vs. System B. Usability Evaluation

- First, define a set of scenarios choose a subset.
- Next, for each scenario define a set of iid tasks (derived from the scenario with varying constraints.
- Determine the effort metrics (e.g., TOT, number of fixations, et.)
- Select a group of subjects for each system.
- Get the subjects to perform the tasks and measure the defined metrics.
- Use the average curve to identify learnability and operability (efficiency is highly correlated to operability)

System A. Vs. System B. is used for general assessment of Usability (WRT to tasks and apps)

e.g., TRACS vs. Canvas vs. blackboard (make vs. buy, buy-1 vs. buy-2, or continuous improvement)

Can we identify (pinpoint) usability issues in a way that directs the developers to the actual interface code.

In pinpoint analysis we segment the task (can be several minutes) into time segment units

1) Equal time (e.g., 30 seconds) uniform time segments
2) Event based a segment starts with an event (e.g., mouse click or KBD click) and ends with an event (mouse click, KBD click)

After segmentation, for each segment measure the selected metric[s] (e.g., number of fixations) for all the users executing this segment.

Decision function can be a threshold on the average value of the measured metric.

Training (with training subjects)

- Segment
- Measure the metric[s] per subject per segment
- Identify and set a threshold per segment (e.g., one standard deviation from the average of the result of measurements on a specific metric) for this segment.

Classification (E , NE) (with Classification subjects)

- Segment
- Measure the metric[s] per subject per segment
- Compare the average measurements to the threshold (per segment) of the average result with the training threshold. Above ➔ E (supervisor has to check), Below (NE – skip)

The recording of the entire sessions is several hours. How to find issues in the interaction?
Method 1) An expert evaluates the entire set of videos to identify issues.
Method 2 (pinpoint)) The expert only watches clips automatically identified as excessive effort clips.

Pros and cons
Method 2 pro reduces the time that the expert has to expand in watching video
Cons recognition errors

Type 1 - (False alarm) an NE segment is classified as E - increases expert time spent on watching interaction videos
Type 2 – False Negative an E is classified as NE - issues are not detected.

Possible "compromise" two pass; one which allows for higher error of one of the two types.
Second pass (tighter) to reduce the selected error.

There is an issue with including QTOpenGL in assignment I2.
Solution:
Add QT += opengl to the .pro file.

The assignments calls to add a graphics "window" to I1.
QT does not recognize GLU functions (glulookat(), gluperspective()) They can be used with the proper include (include glu.h in the right way).

```
//Draw a quad
glBegin(GL_QUADS);
  glColor3f (1.0, 0.0, 0.0);
  glVertex3f(-radius, -radius,  radius);
  glVertex3f( radius, -radius,  radius);
  glVertex3f( radius,  radius,  radius);
```

```
        glVertex3f(-radius,  radius,  radius);
    glEnd();


    glFlush ();
}
```

Static Polygon the enclosed area is included.

You can do several of the assignment objects as n-gons

Circle 100-gon

Can use the plotting for Circle (requires a loop) Using the parametric equation

with 360 degrees is equivalent to drawing a 360-gon

Design by contract

Start from Wikipedia, consider looking at Eifel

For Functional requirements specifications, testing validation

We are using it for non-functional requirements (of usability) using design by contract.

P (pre-assumptions)

Functional; Non-functional (e.g.,) User Interaction

Q (post assumption)

Stated in “Lawyer Language”

Alternative

Hoare Logic

P (logical assertion)

Functional; Non-functional (e.g.,) User Interaction

Q (logical assertion)

Current Metrics

1. Efficiency
2. Operability
3. Productivity
4. Effectiveness
5. Understandability
6. Learnability
7. Satisfaction
8. Attractiveness

What is effectiveness?

Consider the task of hotel reservation under constraints (e.g., budget)

Accuracy & percent completed

How do we verify accuracy and completeness

You (GUI designer) probably have to write a code snippet inside the GUI interface to check

How to measure it? Average of percent complete, measure of accuracy

"Design Principles"

Preconditions (Natural language contract language)

Activities:

System activity – prompt the user to enter password (widget?)

User activity (response) User enters a password

System activity – check the validity of the password

User activity

.

.

.

Postcondition

Procedure for A System A (Zoom) vs. System B (WebEx) Effort Based Usability Evaluation

For midterm TBD?

Zoom vs any other comparable app

TRACS vs Canvas

Some other systems

Procedure for A System A (Zoom) vs. System B (WebEx) Effort Based Usability Evaluation

1) Identify a set of "Scenarios" (scenarios for reservation activities)
    a. Choose 1 associated with one "mid complexity" task (task and constraints)
2) For each scenario (common to A and B)
    a. Select metrics
    b. Define a set of iid tasks derived from the scenario
    c. Test duration?
    d. Identify potential users and potential subjects (User centric)
    e. Obtain permission (ARB) to perform the tests (Tests on Human subjects)
    f. Consider incentivizing the subjects (competition?)
    g. Specify the task execution procedure (Tutorials overviews)
3) Get a set of subjects to execute the iid tasks (one by one) on system A, another subset of users for system B (same order, same level of expertise)
4) Measure the selected metrics
5) Obtain averages and approximating curves
6) Assess the data (curves) and identify the levels of learnability and operability with each system
7) Produce reports (e.g., using CIF)

Consider a system for automatic remote health management

Select a scenario think about a "basic" operation + set of varying "constraints"

Example scenario
Schedule an appointment with a physician
Constraints Main provider (PCO), expert provider, urgent, follow up, you dependent
Obtain test results
Get a test
Talk with a nurse about a concern

A scenario on a system like TRACS (Blackboard, Canvas)
A professor wants to administrate an assignment via TRACS
A student wants to complete an assignment
A student wants to find her standing in the class
A professor wants to generate a report about student performance in class

Procedure for **pinpoint** analysis on system A.

First perform an effort-based Usability analysis of A.: steps 1 – 7 above

1) Identify a set of "Scenarios" (scenarios for reservation activities)
    a. Choose 1 associated with one "mid complexity" task (task and constraints)
2) For each scenario (common to A and B)
    a. Select metrics
    b. Define a set of iid tasks derived from the scenario
    c. Test duration?
    d. Identify potential users and potential subjects (User centric)
    e. Obtain permission (ARB) to perform the tests (Tests on Human subjects)
    f. Consider incentivizing the subjects (competition?)
    g. Specify the task execution procedure (Tutorials overviews)
3) Get a set of subjects to execute the iid tasks (one by one) on system A, another subset of users for system B (same order, same level of expertise)
4) Measure the selected metrics
5) Obtain averages and approximating curves
6) Assess the data (curves) and identify the levels of learnability and operability with each system
7) Produce reports

At the end of step 7 we have obtained measurements for an individual tasks within a scenario.

The tasks take a few minutes have a few steps for completion

Next,

Pattern Recognitions (classify interaction in **time segments** into several classes based on the level of effort) EE, NE (excessive effort) identify Excessive effort segments

Training (training set – of subjects and the results)

Classification (classification subjects → results

Training (training subjects)

Decide how to segment

Uniform time (e.g., 20 seconds per segment)

Event driven from one interaction activity (e.g., mouse click) to next.

A specific interaction activity.

Choose the metrics

Chosen by an expert (supervised)

Chosen via a non-supervised training (e.g., clustering)

Select a decision function determine whether a segment is E or NE

Neural Networks, Clustering, **Thresholding**

Assume a threshold on saccade amplitude is selected as the decision function

Set the threshold. For example, one standard deviation above the average number of saccades (per segment) obtained in the training stage.

We have to get the training subjects to perform the tasks, measure the number of saccades per segment per task per user and average over segments

Classification (E, NE)

For the scenario, with a set of (classification) subjects each performs the test with iid tasks

i. Segment the task completion data (per subject per iid task)
ii. Measure the metrics per segment per subject (saccade amplituse)
iii. Apply the decision function; potentially on average results per segment per subject
iv. Obtain decision results
v. A supervisor is asked to evaluate the usability issues (if there are issues) in the
vi. segments classified as E.

Assume a threshold method is used. Generally applied to one measurement at a time.
Select one and only one metric for the threshold (say #saccade amplitude)
Use more than one metric (solve conflicts)
Use one threshold on a linear combination of the measurements

Apply the decision function

For each segment compare the average number of saccades per segment obtained with the classification subjects to the threshold (average plus one standard deviation obtained with the training subjects)

Decision

Label the segments where the result is above the threshold as E
Label the segments where the result is above the threshold as NE
Have an expert to evaluate the segments classified as E.

CIF – Common Interface Format – NIST Standard for reporting Usability
Current version is $$$ CR, I have an old version

SUMI

Guidelines for questionnaire

Announcements

- For Assignment I1 – a solution is (will be) posted on TRACS
- Assignment I2 can be implemented by opening a dock widget in addition to the Graphics window and placing the widgets in the doc widget (see /usr/lib/qt4/demos for a "lot" of relevant demos, specifically for the /usr/lib/qt4/demos/mainwindow – includes a dockwidget).
- /usr/lib/qt4/demos/mainwindow – includes an example of menu, doc-widget and an edit window. You may need a dock widget and a graphics window.
- The mid will be an open book material assigned by Friday due Tuesday. There will be a verbal component later on where I will meet students and ask them to explain their solution. The Mid is graded based on the written part.
- Mid you will have to specify how to do
- Assignment you will have to assess
- Posted the Quiz solution from last year can be used as an example for effort based usability questions

In exam

You get a system and asked questions about how would you?

Define, measure, tests, set requirements for effort-based system a vs system b and or pinpoint analysis

Maybe I will have an assignment with pairs doing effort-based usability on two systems

June 22

In Mid question 1:

Device D (not GUI with a human)

Generates events periodically X events per second.

Dedicated OS queue every occurrence of an event is logged into the Q in a FIFO manner

QT manages its own Q (difficult to access) by fetching events from the OS Q

Use retrieve takes it off the Q

1.a You have to access the Q. Implement a mock Q of your own use polling with this Q.

Option 1 use QT timer

Option 2 improvise your own timer

1.b Define the callback function figure out how it is being called and what to do when it is called

Check the slides for OGL call back functions

2. Detailed snippet

Part 2 – do not implement your Task.

Provide a design procedure

The exam is graded based on your written answer and written explanation

Verbal part

1) Enforces you to study in a way that increases your understanding of the problem
2) It is some kind of a gate keeper for making sure that you understand what you answer

I will randomly select students to meet via zoom and explain their solution

Students that do not provide a convincing written explanation are likely to be selected for Verbal

Back to OpenGL

iPhone, Android are using a GPU. You can program the GPU (for Graphics and acceleration) probably via middleware The GPU supports most of the OGL functions.

Process:

1) Define the camera

   Parallel projection (did so far, intuitive)

   Perspective Projection (less intuitive a trial and error, aperture camera position)

2) Construct and manipulate the model
   a. Construct: define a set of dynamic/static vertices
      i. A mesh is a set of vertices that describe an object; generally, via tiling of triangles or square (vertices) over an object. Create a library of objects. Instantiate object (duplicate objects)
      ii. Plotting functions
   b. Manipulate: transformation on objects. We are using a set of transformations that are called affine transformations. Why do we use these transformations?
      i. They represent a large set of desirable graphics effects
      ii. These are rigid body transformation the shape of the body does not change via the transformation
      iii. They preserve lines (not angles) ➔ We can apply them to vertices; the edges follow through (straight lines between vertices)
      iv. We can use these transformation to get the effect of moving the camera by moving objects relative to the stationary camera
      v. All of the affine transformation can be implemented via a set of basic transformations of Translation, Rotation, and Scaling (often text-books consider Shearing as a basic transformation).
      vi. Translation, Rotation, and Scaling can be implemented extremely efficiently by hardware for matrix multiplication (the Graphics Pipeline, GPU).
   c. Set lights and material interaction with lights
   d. Plot functions via approximation (the function s are not precisely given)
3) Project the scene generated in 1 and 2 into standard representation (OGL) NDC normalized device coordinates
4) Project the NDC onto a view-port, in a window, on the screen (Check slides)

The Camera

OGL camera is placed at the Origin of the Model (0, 0, 0, 1)

Facing the direction of the -z axis.

The Vew Volume

The default volume is a cube of size 2 centered at the origin.

Equivalent to glOrtho(-1, 1, -1, 1, -1, 1) this is also the NDC

In QT (GLUT) you define a window on the screen, the (0,0) is the leftmost upper most

point of the screen (static) the location changed through QT UI

Place a viewport inside the window (dynamic) changed by the program

glViewport(x, y, w, h) places a view port at a distance of (x, y) from the (0,0) of the window (left nost bottom most part of the window. The view port is of dimensions h, w

The mapping from NDC (-1, 1, -1, 1, -1, 1) to the view port is done by setting Z of every vertex to 0

The CTM (current transformation matrix) is obtained by multiplying two “major” open GL matrices (transformations 4x4 matrix) CTM is M*P. Every vertex in the scene is being multiplied by the CTM

The Projection matrix takes care of the actual projection (parallel via Ortho, Perspective via Frustum)

The Model-view matrix takes care of rotation, translation, scaling, etc.

Rotation translation and scaling change M. But since the CTM is M*P they also change the CTM

To set the camera (projection) Change P

Start with the default CTM “do nothing” matrix the Unit matrix

```
glMatrixMode (GL_PROJECTION);    We want to change the Projection matrix (P)
glLoadIdentity ();     load identity  No transformation
glOrtho(l, r, b, t, n, f);  ; changing P to enable the parallel projection

glMatrixMode (GL_PROJECTION);    We want to change the Projection matrix (P)
glLoadIdentity ();     load identity  No transformation
glFrostum(l, r, b, t, n, f);  ; changing P to enable the perspective projection
```

To make transformations we need to change M (it does change the CTM BC CTM is M*P)

glMatrixMode(GL_MODELVIEW); Working with M – meaning we want to do
glLoadIdentity(); Initiate to "no transformation"
glTranslatef(1.0, 2.0, 3.0);

Assume that I have a vertex at V= (5, 6, 7, 1) the precise way to denote a vertex

Result on V? (6.0, 8.0, 10.0)

June 23.

In the common mode of operation, Rotation, Translation and Scaling are applied to the Model_View Matrix

OGL has several matrices (part of the state). The Projection and the Model_View are 4x4 matrices.

They operate on vertices and on vectors which are now represented as 4x1 entities.

The result of the operation is either a change in the view volume (projection) or in the model Model_View

Coordinate set for vertices and vectors (used to enable all of the OGL transformations to be implemented by matrix mutliptication).

A point (Vertex) is now represented as a column [x, y, z, w] w != 0

A vertex is represented in Normal form as [x, y, z, 1] W=1

A vector is represented as [X, Y, Z, 0]

- ⇨ Enables performing all the transformations (P and M) as P*[Vertex/Vector]
- ⇨ M*[Vertex/Vector] ([4x4]*[4x1]=[4x1] Take a matrix of 4x4 multiply by a vertex (4x1) or by Vector (4x1) getting a "new" vertex/vector

The combination of M*P is saved by OpenGL and referred to the Current Transformation matrix.

Initialized to identity. Changed as a result of Projection / Model_View OpenGL functions

1	0	0	0		3		3
0	1	0	0	X	2	=	2
0	0	1	0		1		1
0	0	0	1		1		1

1	0	0	0		3		3
0	1	0	0	X	2	=	2
0	0	1	0		1		1
0	0	0	1		0		0

I*vertex/ vector ➔ same vertex / vector (I stands for identity)

M	M	M	M		V		V'
1	0	0	10		3		13
0	1	0	20	X	2	=	22
0	0	1	30		1		31
0	0	0	1		1		1

We started with the vertex $V=[3, 2, 1, 1]^T$ $M*V=V'$ $[13, 22, 31, 1]^T$

Actually, V' is the result of translating V by (10, 20, 30)

In OpenGL use glTranslatef(10, 20, 30)

M	M	M	M		V		V'
1	0	0	dx		x		x+dx
0	1	0	dy	X	y	=	y+dy
0	0	1	dz		z		z+dz
0	0	0	1		1		1

glTranslatef(dx, dy, dz)

Inverse glTranslatef(-dx, -dy, -dz)

glMatrixMode(GL_MODELVIEW); Working with M – meaning we want to do a transformation
glLoadIdentity(); Initiate to "no transformation" Loading I into CTM current state
glTranslatef(1.0, 2.0, 3.0); M ← $M*M_T = I* M_T$ ➔New CTM is M_T *P
Effect every vertex in the scene/object is multiplied by the new CTM before rendering ➔ translating the object

M	M	M	M		V		V'
cos (θ)	-sin(θ)	0	0		x		x*cos(θ) $- y *$ sin (θ)
sin(θ)	cos(θ)	0	0	X	y	=	x*sin(θ) $+ y *$ cos (θ)
0	0	1	0		z		z
0	0	0	1		1		1

Rotation by θ; z do not change. It is rotation about Z (on the XY plan)
Suppose I have rotated by θ how do I get the inverse rotation by θ
Cos(-θ)=cos(θ) sin(-θ) -sin(θ)

In OGL glRotatef(θ, a, b, c), θ in degrees, rotate by θ around the vector [a, ,b c, 0]^T
In the matrix above I should use glRotate($\theta, 0, 0, 1$) around the Z axis 0 1 0 around the Y axis
The inverse is:
glRotate($-\theta, x, y, z$)

M	M	M	M		V		V'
1/2	0	0	0		x		x /2
0	5	0	0	X	y	=	5y
0	0	3/5	0		z		3/5*z
0	0	0	1		1		1

Scaling
Inverse scaling? ½ ➔ 2/1 scaling by Sx, Sy, Sz
glScalef(Sx, Sy, Sz);
Inverse

glScalef(1/Sx, 1/Sy, 1,Sz).

M	M	M	M		V		V'
1/2	0	0	0		0		0
0	5	0	0	X	0	=	0
0	0	3/5	0		0		0
0	0	0	1		1		1

The point [0, 0, 0, 1]^T, the origin is the fixed point for scaling for rotation.
Scaling and rotation is done with respect to the origin

Note that Rotation Translation and Scaling are applied to points (vertices).BC it is Affine transformation it affects the entire object maintaining edges as straight lines

Scaling and rotation are done with respect to the fixed point, the origin.

What if we want to rotate around another fixed point, say the center of the object?

To scale/rotate around another fixed point we need to translate so that that fixed point is in the origin. Next, scale / rotate; then translate back.

BC Open GL use post multiplication we reverse the order to the center Check slide 8 in today's presentation.

How to get back to a previous state of transformations

glLoadidentity back to ground 0 no transformation
glPushMatrix() mush the current matrix to a stack
glPopMatrix() pop from the stack

What is the effect of:

```
glMatrixMode(GL_PROJECTION);   Working with P – meaning we want to do a projection transformation
glLoadIdentity();
glMatrixMode(GL_MODLEVIEW);   Working with M – meaning we want to do a Model transformation
glLoadIdentity();
```

Takes us back to the default of OGL
Modelview view volume is (-1, 1, -1, 1, -1, 1)
Parlalel projection

So now

[a, b, c, d] is by convention a row vector/points. $[a, b, c, d]^T$ means a column vector.

$[x, y, z, w]^T$ w!=0 means a point (a vertex) (3, 5, 4, 2), if w=1 the representation is referred to as normalized (3/2, 5/2, 4/2, 1) normalized

$[x, y, z, 0]^T$ denotes the vector that "starts" at the origin and points to $[x, y, z, 1]^T$.

Can distinguish between vectors and points more important all the OGL transformation can be implemented via 4x4 * 4x4 4x4 * 4x1

During the class meeting 4:00 – 6:05 tomorrow (via Zoom). I will manage the verbal part

I will interact individually with selected students.

Slide 6.

To rotate (scale) around an arbitrary fixed-point P, intuitively: 1) Translate the objects so that P is at the origin, 2) Rotate (scale) 3) Translate back (use inverse translation)

Practically, due to post-multiplication we reverse the order. 1) apply the inverse translation 2) Rotate (scale), 3) apply the forward translation

How to place the Camera.

The camera is at [0, 0, 0, 1] facing to the negative Z direction.

In the classroom example the camera can only see half of the class

Can only the back of the students.

Desired to see the faces of all the students.

Move the room (translate), move the students rotate.

To move the camera, slide XYZ show the sequence for the specific example of moving the camera

The is a glu function to make it easier. You have to include glu with QT

Or use macros available to DT

Glulookat is a macro that applies the right sequence given desired camera position, desired point that the camera can see, and the camera orientation

The preferred order is first rotate than translae

Projection transformations

Parallel: glOrtho this sets the camera to behave like a parallel projection camera (for drawing and plotting)

Perspective glFrustum this sets the camera to behave like a perspective projection camera like our vision

Operating on P

Operating on M rotate, translate, scale, gluLookat is actually using these M transformation to move the objects thereby relative to the objects moving the camera

UG 501 810 492; UG 502 531837; G 501 601612; G 502 327207

Shading (lighting), not shadows
OGL does not apply a strict physics-based model of shading.
To comply with the physics "rules" one has to use Ray-Tracing (global model)
OGL – provide a local model that is not physically accurate. But, provides many options to emulate realistic lighting.

We assume 3 types of light

1) Point source
    a. bulb ~point source at a fixed location the light is propagating from the point to all direction.
    b. Point source at "infinite" Sun ray of lights are ~parallel
2) Spotlight point source with restrictions light propagates only within a cone of angel Phi
3) Ambient light

Combine properties In R, G, B

We assume 3 types of material

Rough (diffusive, lambertian)
Smooth (specular)
Ambient (not really a physical property of material but provides us with on more degree of freedom
Combine properties In R, G, B, Plus alpha changes the appearance from ideal mirror to metal like to plastic like

OGL uses a model - Phong model that takes into account the above 19 parameters along with the location of the light source and the location of the camera to generate the actual shading

Process.

You define the 19 parameters, per light source and per vertex (vertices can be on different matrials)

Define location of light sources, viewer location (camera)

OGL uses the Phong (Blinn) model to produce the shading.