

PAGE 3

3D modeler requirements

A 3D modeler

Elqoo

Create models from basic shapes

Combine objects

Workplane

www.elqoo.com

PAGE 4

Rendering Requirements

Rendering requirements for our project

Elqoo

Quick Rendering

Full Rendering

www.elqoo.com

.....

Multiplatform

The 3D editor must run on Mac, Windows and tablets

PAGE 5

My first Lesson - Learning the moves

Workplane

Elqoo

www.elqoo.com



Design Problems

Design issues with our case study

Document Structure

How to represent the data?

Screen Rendering

Different ways to render screen

User operations

Capture user operations?

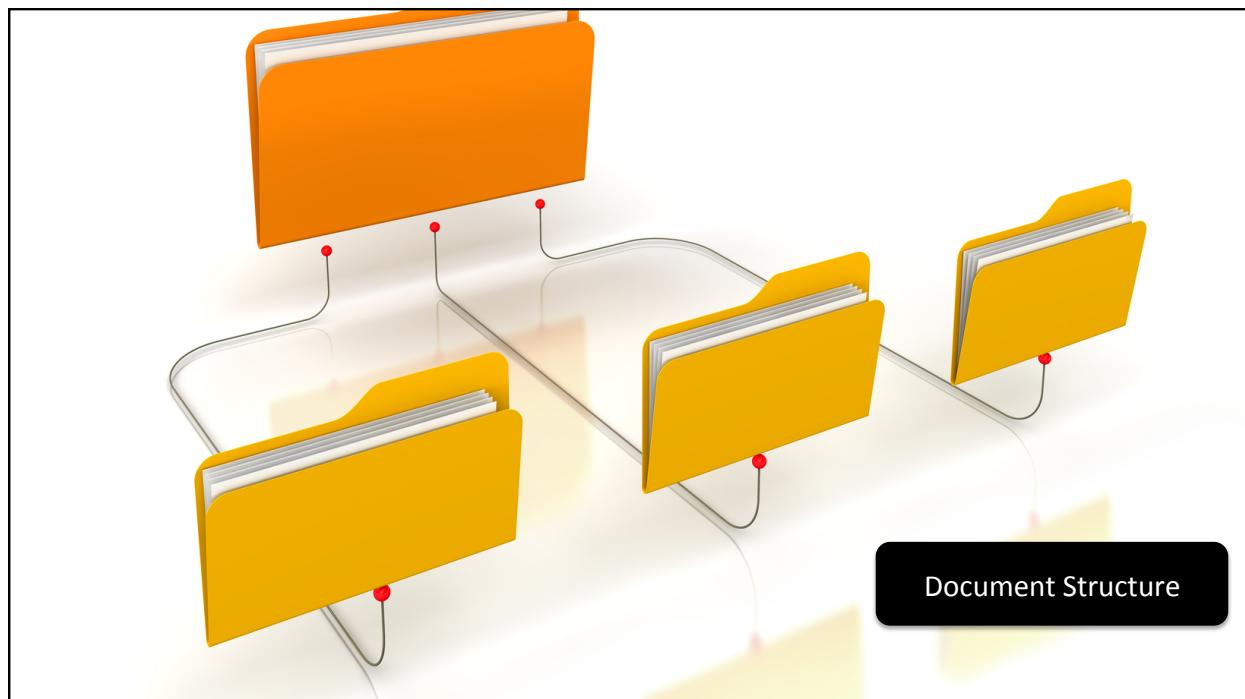
Support multiple platforms

Windows, Mac, Tablet, browser

Elqoo

www.elqoo.com

Document Structure



.....

Document Structure Importance

Why do we need to choose a good document structure?

PAGE 9

```

graph TD
    DS((Document Structure)) --> RS[Rendering System]
    DS --> E[Editing]
    DS --> Other[...]
    DS --> Heart["Heart of the system"]
    DS --> UsedBy["Used By"]
  
```

The diagram features a central blue cloud-like shape containing the text "Document Structure". Three arrows point from this central shape to three rectangular boxes below it: "Rendering System", "Editing", and "....". To the right of the central shape is a teal speech bubble containing the text "Heart of the system".

Elqoo

www.elqoo.com

.....

Document Structure Choice

UML design decisions

PAGE 10

- **Simple Shapes**
 - Define as object class
- **Composed Shapes**
 - Define as object class
 - Complex object → E.g house, car

Elqoo

www.elqoo.com

.....

Document Structure

UML representation of the document structure

```

classDiagram
    class Circle
    class Square
    class Composed {
        <<Composite>>
        Circle
        Square
    }
    Circle --> Composed
    Square --> Composed
  
```

PAGE 11

Elqoo

www.elqoo.com

.....

Document Structure Pseudo Code

UML representation of the document structure

```

classDiagram
    class Circle
    class Square
    class Composed {
        <<Composite>>
        Circle
        Square
    }
    Circle --> Composed
    Square --> Composed
  
```

```

If (object is composed) then
  Do action with children
Else
  Do action with object
  
```

PAGE 12

Elqoo

www.elqoo.com

Improved Document Structure
Tree representation of the document

```

graph TD
    World[World] --- Object1[Object1]
    World --- Square[Square]
    World --- Circle1[Circle]
    Object1 --- Circle2[Circle]
    Object1 --- Circle3[Circle]
  
```

PAGE 13

Elqoo www.elqoo.com

Improved Document Structure
Translate the tree structure into UML

```

classDiagram
    class WorldObject
    class Circle
    class Square
    class ComposedWorldObject

    Circle <|-- WorldObject
    Square <|-- WorldObject
    ComposedWorldObject --> Circle
    ComposedWorldObject --> Square
  
```

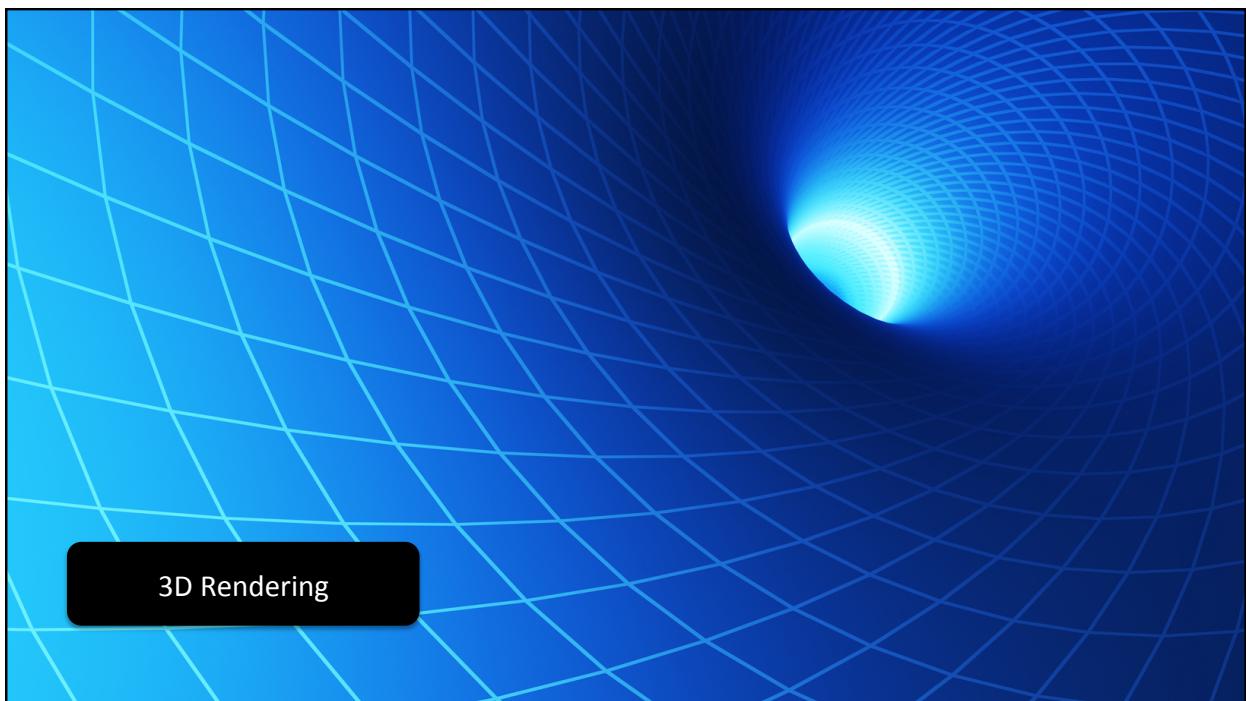
PAGE 14

Based on the Composite Pattern

Elqoo www.elqoo.com

Solved Problems

- **Fixed**
 - Composed Object **only** references other world objects
 - Client doesn't need to know the difference



Rendering Requirements
Rendering requirements for project duck sauce

PAGE 17

My first Lesson - Learning the moves

Workplane

Quick Rendering

Full Rendering

Elqoo

www.elqoo.com

Screen Rendering Problem
How to show the objects on the screen

PAGE 18

Multiple renderers?

Add Render method to WorldObject

Switch rendering at runtime?

```

classDiagram
    class WorldObject {
        render()
    }
    class Square {
        render()
    }
    class Circle {
        render()
    }
    WorldObject <|-- Square
    WorldObject <|-- Circle
  
```

Elqoo

www.elqoo.com

.....

Problem Solution

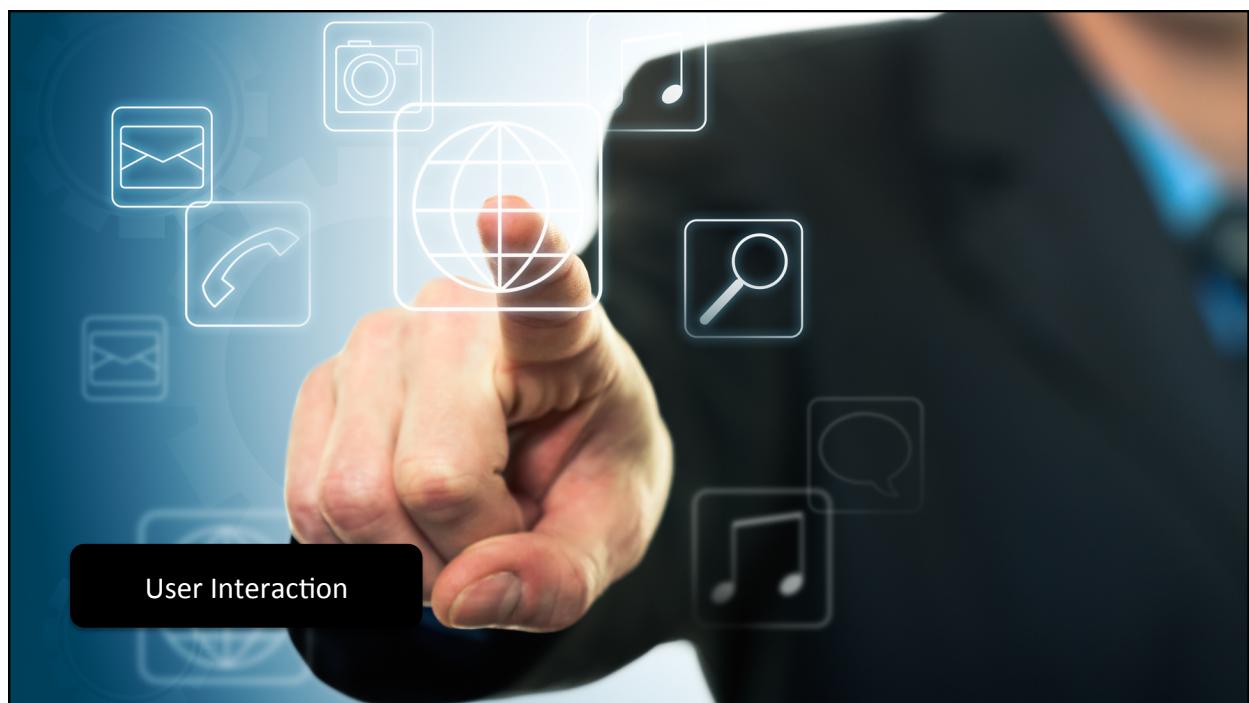
```
classDiagram
    class WorldObjectRenderer {
        render(WorldObject)
    }
    class QuickRenderer {
        render(WorldObject)
    }
    class DetailedRenderer {
        render(WorldObject)
    }
    WorldObjectRenderer <|-- QuickRenderer
    WorldObjectRenderer <|-- DetailedRenderer
```

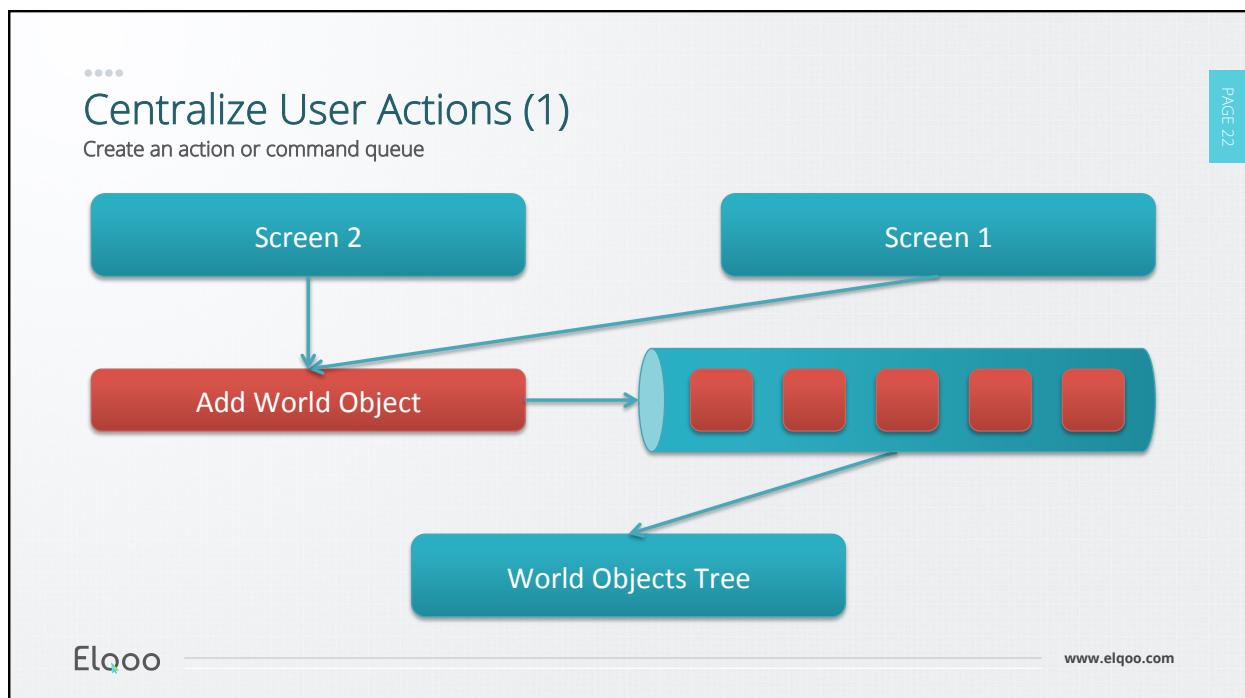
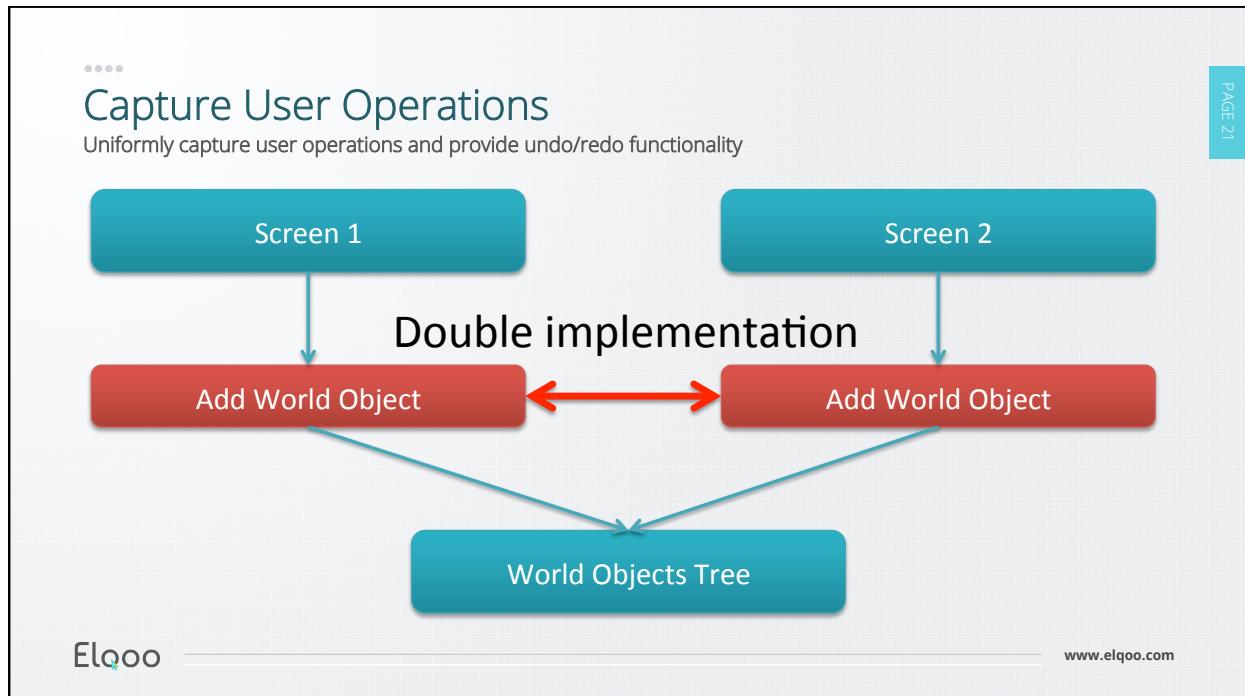
Centralize render logic in separate interface

Different render implementations

Based on the Strategy Pattern

Elqoo ————— www.elqoo.com





Centralize User Actions (2)

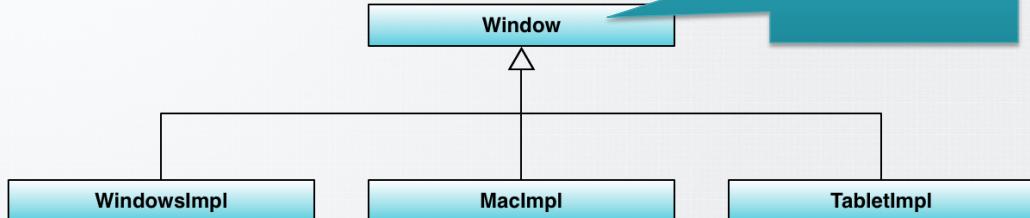
- Queue user actions → centralize functionality
- Actions → provide undo method

Based on the Command Pattern



Multiple Platform Support

Solved with standard inheritance



What about multiple types of windows: Iconwindow, Dialog, Application?

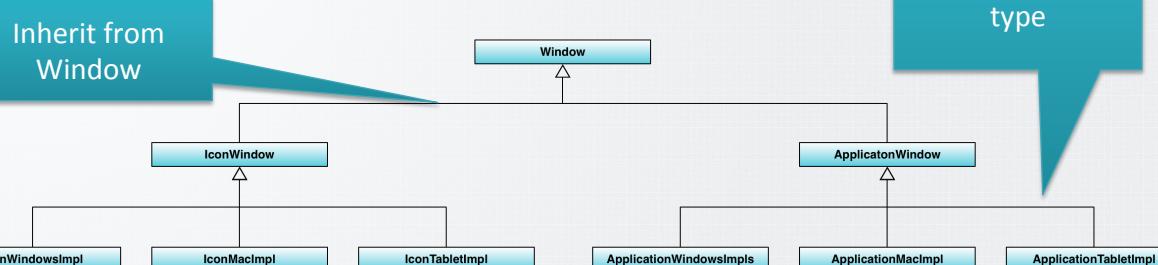
Elqoo

www.elqoo.com

Extend Inheritance Tree

With a strong inheritance structure we get an explosion of classes

Implementation for each window type



Explosion of class definitions!

Elqoo

www.elqoo.com

