aCAD – an Android CAD Application for Developing Irrigation Schematics

Project By
SOCS

Date
December 6, 2012

Overview

- Team composition
- Project description
- Functional requirements
- User interface design
- Use case diagram
- Demo
- Test cases and test report
- Deployment issues
- What we learned?

Project Description

- Android tablet application for drawing and publishing irrigation schematics

Constraints:

- Development time
- Developer's lack of experience and knowledge of irrigation systems

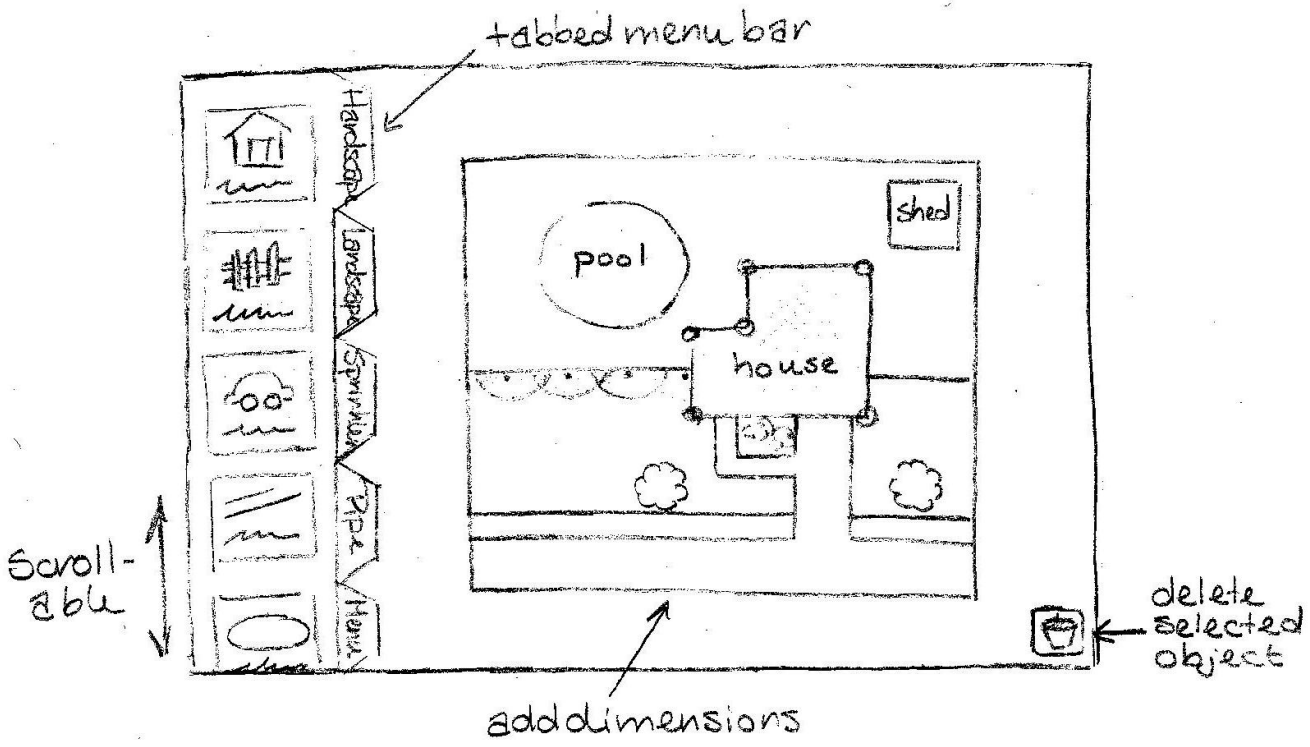
Scope of the Project

- A graphical user interface (GUI) will be used to create irrigation schematics
- Application will utilize gestures native to the Android operating system
- Create a prototype with basic functionality which can be expanded and refined by the Spring 2013 students

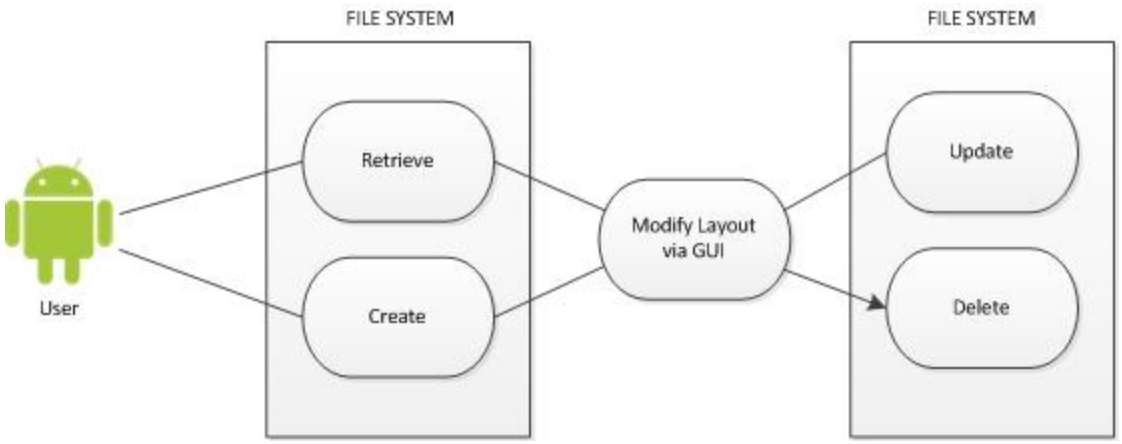
Functional Requirements

- File manipulation for opening and saving irrigation system layouts
- Save image representation of schematic
- Add/Edit database containing pipes and fittings
- Pan/Zoom functionality in the layout area
- Provide cost estimate for materials

User Interface Design



System Diagram



System Diagrams

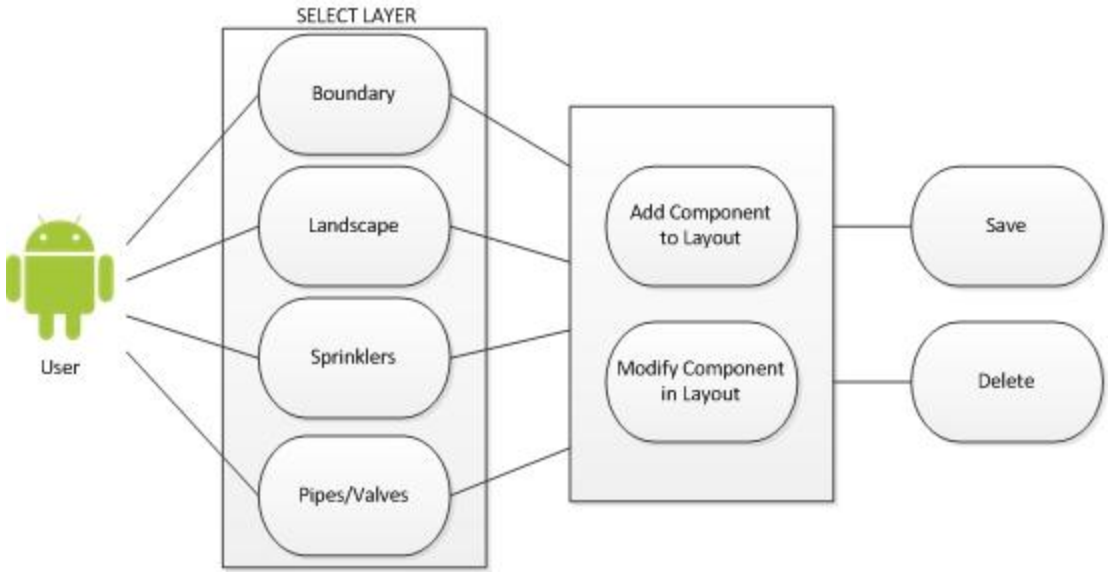
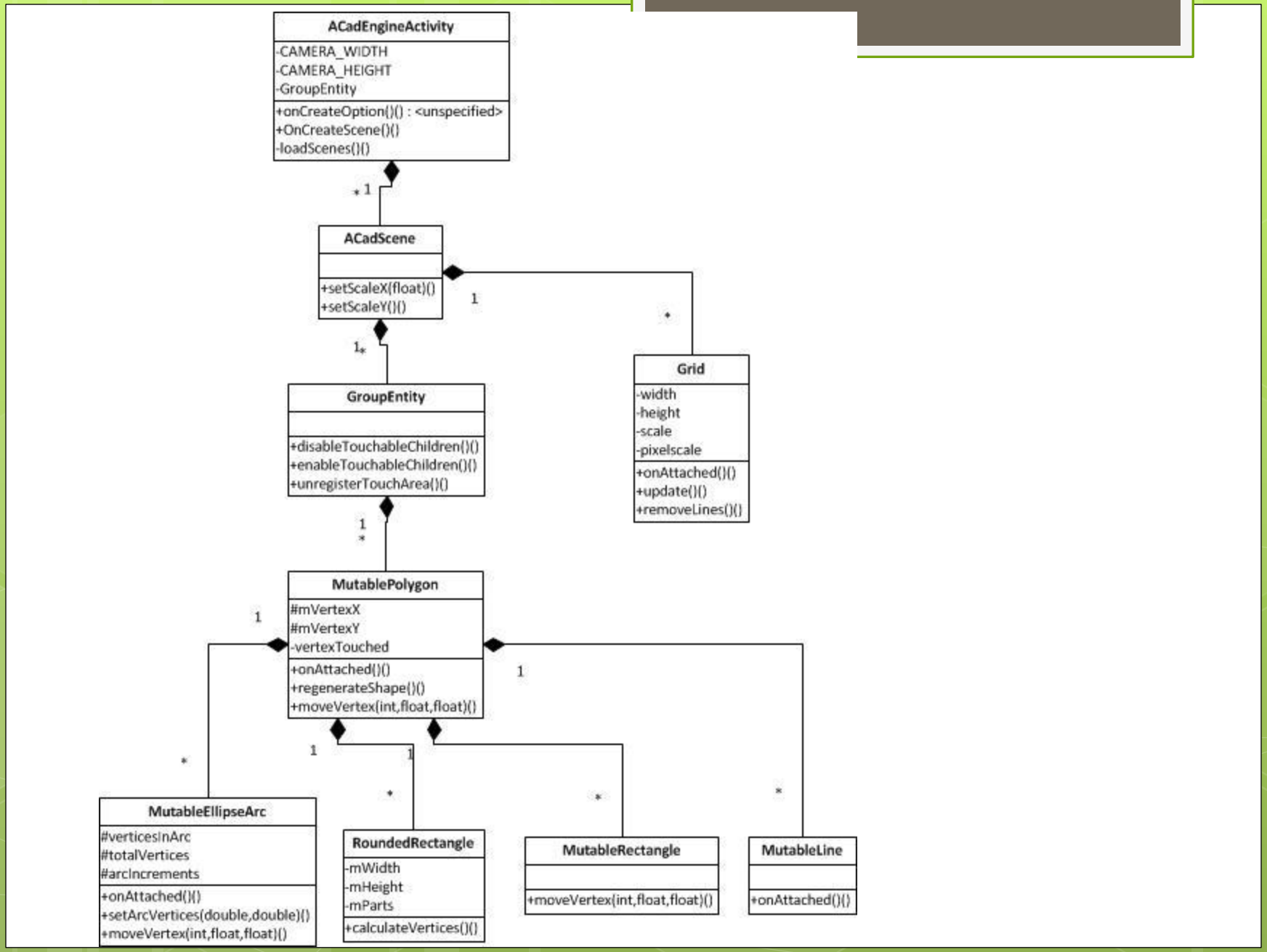


Figure 3 – User Interaction with GUI While Modifying Layout

Use Case Diagram


ACadEngineActivity
-CAMERA_WIDTH
-CAMERA_HEIGHT
-GroupEntity
+onCreateOption(){} : <unspecified>
+OnCreateScene(){}
-loadScenes(){}
* 1
ACadScene
+setScaleX(float){}
+setScaleY(){}
1
*
1*
Grid
-width
-height
-scale
-pixelscale
+onAttached(){}
+update(){}
+removeLines(){}
GroupEntity
+disableTouchableChildren(){}
+enableTouchableChildren(){}
+unregisterTouchArea(){}
1
*
MutablePolygon
#mVertexX
#mVertexY
-vertexTouched
+onAttached(){}
+regenerateShape(){}
+moveVertex(int,float,float){}
1
1
1
1
*
*
*
*
MutableEllipseArc
#verticesInArc
#totalVertices
#arcIncrements
+onAttached(){}
+setArcVertices(double,double){}
+moveVertex(int,float,float){}
RoundedRectangle
-mWidth
-mHeight
-mParts
+calculateVertices(){}
MutableRectangle
+moveVertex(int,float,float){}
MutableLine
+onAttached(){}

Demo

Functional Requirements

- ✓ File manipulation for opening and saving irrigation system layouts
- ✓ Save image representation of schematic
- Add/Edit database containing pipes and fittings
- ✓ Pan/Zoom functionality in the layout area
- Provide cost estimate for materials

Test cases/results for menu

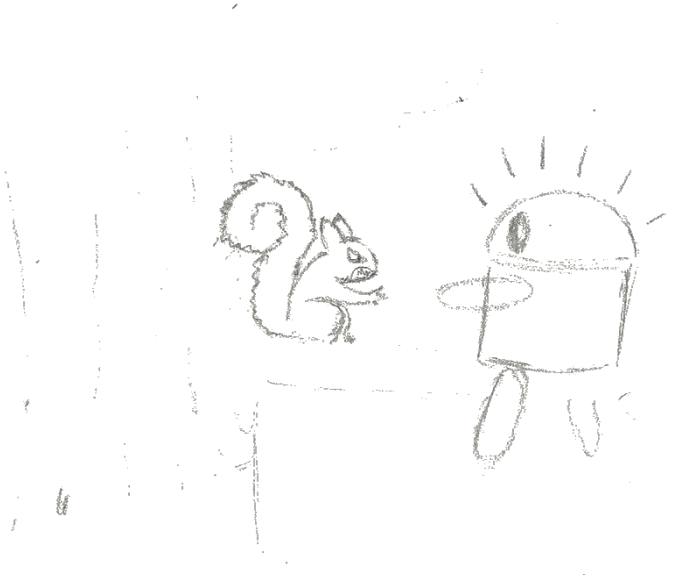
Test Cases	Expected Result	Actual Result	Result
Click on Menu	Display list of menu options	Displayed list of menu options	Pass
Open file	Open selected file and load design	Design not loaded	Fail
New	Create new design entering dimensions	Created new design	Pass
New	Creates new design not entering dimensions	Application crashed when Ok button clicked	Fail
Save as	Asks for file name and saves design	Asked file name and saved design with that	Pass
Validation of input entered in alert dialog	Report proper error	Not yet implemented	Fail
Checking file exists	Saving file with same name of some other file	Not yet implemented	Fail
Exit	Asks for confirmation and close based on selection	Asked for confirmation and closed based on selection	Pass
Recenter	Center drawing in grid viewing area	Centered drawing in grid viewing area	Pass

Test Cases/Results for shapes

Test Case	Expected Result	Actual Result	Pass/Fail
Add circle/pie	Tap layout to draw circle/pie on	Drew circle/pie on layout	Pass
Edit circle/arc	Drag corner to Increase angle/radius	Increased angle/radius	Pass
Add polygon	Tap layout to draw polygon	Drew polygon on layout	Pass
Edit polygon	Tap polygon to add more sides	Added more sides	Pass
Edit polygon	Drag edges out to alter polygon shape	Shape altered	Pass
Add rectangle	Tap layout to draw rectangle	Drew rectangle on layout	Pass
Edit rectangle	Drag corner points to increase width/height	Increased width/height	Pass
Add line	Tap layout to draw line	Drew line on layout	Pass
Edit line	Drag end points to increase length/direction	Increased length/direction	Pass

Known problems with system

- Hangs when a hole is created within a shape (system problem with ear-cutting algorithm)



If we had more time

- Use FPS AndEngine Activity to reduce battery consumption
- Add piping and valves
- Implement optimization of sprinkler placement and zone calculations
- Calculate pressure losses
- Saving as image file for printing
- Adding different types of line styles and colors for the different layers
- Add a legend and title block
- Implement animations
- Add soc(k)s to the android on the splash screen

What we would do again

- Use Google Drive for document sharing
- Github for version control
- Skype for online meetings
- AndEngine for object management
- Planning Poker for making the difficult decisions

Conclusion

- Interesting concept, but too little time
- App still needs a lot of development
- We improved our Android programming skills
- Learned how to work as a group