Sweet Home 3D – Agile Development

# Introduction

This document specifies the system requirements and generation information of the proposed “Advanced home plan editing plug-in” enhancement to the Sweet Home 3D interior design tool.

## Purpose

The purpose of this document is to give detailed specification of the software requirements for this plug-in. The plug-in is designed to assist users resize selected objects in a given home plan. The intended audience for this document are the developers, the system designers, the test engineers and the project manager. This specification describes in detail the requirements for each of the functions of the plugin.

## Scope

This document applies only to the relevant people of this project. This specification does not concern itself with any actual system design or implementation issues.

## Definitions, Acronyms, and Abbreviations

* SRS - Software Requirements Specifications
* IEEE - Institute of Electrical and Electronic Engineering
* C/S - Client / Server
* B/S - Browser / Server

## References

[1] IEEE 830-1993: IEEE Recommended Practice for Software Requirements Specifications "IEEE Standards Collection, IEEE 1997"

## Overview

In the following sections of this specification the User Interface, Functional Requirements and the relevant test cases will be presented. In section 2, the general features will be introduced. In section 3, all detailed requirements will be specified and grouped.

# General Description

## Product Perspective

This plug-in allows an interior designer to resize selected objects in the home plan. It also allows the designer to edit room points in a dialog and join two walls in the home. The plug-in will add Rotate, Move, Flip and Resize action buttons inside the Edit tab.

## Product Functions

The plug-in provides the following functions:

"Flip" is implemented to apply a flip (mirroring) transformation to selected objects.

"Resize" is implemented to apply a change of size for selected objects with an aspect ratio, in order to enlarge and reduce objects.

"Resize" is implemented to apply a movement of selected objects with 2d coordination (horizontal and vertical distance in the 2d region of a home).

"Rotate" is implemented to apply a rotation of selected objects with a specific number of degrees.

## User Characteristics

Specifications of a user’s computer are not required. A mouse would be required to handle the on-screen item manipulation, however this is no different to the normal utilisation of the Sweet Home 3D Tool.

## General Constraints

This plug-in is developed under J2se 1.6, and has dependencies on the classes files of the ‘SweetHome3D-4.6’ which could be downloaded from “http://sourceforge.net/projects/sweethome3d/files/SweetHome3D/SweetHome3D-4.6/SweetHome3D-4.6.jar/download”.

The target platform is SweetHome3D 4.6.

No additional dependent libraries would be required for this plugin, since it directly integrates into the system.

# Specific Requirements

## External Interface Requirements

### User Interfaces

#### There should be a “Rotate” button under the Edit tab

#### There should be a “Move” button under the Edit tab

#### There should be a “Flip” button under the Edit tab

#### There should be a “Resize” button under the Edit tab

This requirement modifies layout and content of the Edit tab.

## Functional Requirements

### The “Rotate” action should have a field for inputting the degrees of an angle

### The “Move” action should have two fields for inputting horizontal and vertical move distance

### The “Flip” action should have two checkboxes to allow for a horizontal or vertical flip

### The “Resize” action should have two fields for inputting width and height, the measurement fields should be in “cm” and “%”, and an aspect ratio field.

## Software Quality Attributes

Different functions and views should be in separate source files and inherit from the abstract view class. The plug-in should be able to be integrated into the system with no interference with other components. The installation file for this plug-in should be a standalone.

# Development Plan

The project team contains 4 developers, 2 test engineers and 1 team leader.

|  |  |  |
| --- | --- | --- |
| **Time** | **Activity** | **Team Involved** |
| Monday AM | Team leader will host the start-up meeting using ‘WebEx’, collection and analysis of requirements, description of the collected requirements | All Team members |
| Monday PM | Write software and test case design.  Generate documentation and submit to the HP quality centre | All Developers and the Test Engineers |
| Tuesday AM | Review software design in pairs of the four Developers.  Test Engineers review test cases between each other. | All Developers and the Test Engineers |
| Tuesday PM | Pair programming with ‘linter’ (a real time code style checker)  One Test Engineer writes the system test cases  One Developer writes integration test cases | All Developers and the Test Engineers |
| Wednesday AM | Pair programming  One Developer writes integration test cases of the modules | All Developers |
| Wednesday PM | A meeting is held for the whole team to trace the team’s progress  Code review with the ‘collaborator’ tool takes place (for online code review) | All Team members |
| Thursday AM | Both Test Engineers perform system tests on the beta version in the staging test environment  Developers fix bugs according to the priority and time designated by the HP quality centre | All Developers and the Test Engineers |
| Thursday PM | Control system tests and regression tests are conducted based on the test report generated by the quality centre | All Team members |
| Friday AM | The documentation is finalised for presenting the demonstration  Team leader presents this demo. | All Team members |
| Friday PM | The final demonstration is given using the ‘WebEx’ tool with the clients and the whole team | All Team members |

# Detailed Plan - Design and Development

## Implementation of the Base component

|  |  |  |
| --- | --- | --- |
| **Time** | **Activity** | **Team Involved** |
| Tuesday 1pm-2pm | Implement Base components(adding new source files -AdvancedEditingPlugin.java, ImmediateEditDialogView.java, add “Flip” and “Move” into AdvancedEditingPlugin.java) | All Developers |
| Tuesday 2pm-3pm | Code review of base Components | All Developers |

## Implementation of "Flip"

|  |  |  |
| --- | --- | --- |
| **Time** | **Activity** | **Team Involved** |
| Tuesday 3pm-3:30pm | Implement the view of the Flip function(adding a new view source file - FlipView.java) | Developer 1 |
| Tuesday 3:30pm-4pm | Code review of view source file,  Checking that the view fulfils the requirements | Developer 1 and 2 |
| Tuesday 4pm-5pm | Implement the actual logic of the function (adding a new action source file - FlipAction.java, modify apply method in FlipView.java)  Write test cases | Developer 1 and 2 |
| Tuesday 5pm-6pm | Code review of the action and view source files  Run the test cases | Developer 1 and 2 |

## Implementation of "Move"

|  |  |  |
| --- | --- | --- |
| **Time** | **Activity** | **Team Involved** |
| Tuesday 3pm-3:30pm | Implement the view of the Move function (adding a new view source file - MoveView.java) | Developer 3 |
| Tuesday 3:30pm-4pm | Code review of view source file,  Checking that the view fulfils the requirements | Developer 3 and 4 |
| Tuesday 4pm-5pm | Implement the actual logic of the function (adding a new action source file - MoveAction.java, modify apply method in MoveView.java)  Write test cases | Developer 3 and 4 |
| Tuesday 5pm-6pm | Code review of the action and view source files  Run the test cases | Developer 3 and 4 |
| Tuesday 6pm-7pm | Prepare 1.0.1 Plug-in Jar file  Deploy into the integration testing environment  Perform integration tests | All Developers |

## Implementation of "Rotate"

|  |  |  |
| --- | --- | --- |
| **Time** | **Activity** | **Team Involved** |
| Wednesday 8am-8:30am | Implement the view of Rotate function (adding a new view source file - RotateView.java) | Developer 1 |
| Wednesday 8:30am-9am | Code review of view source file,  Checking that the view fulfils the requirements | Developer 1 and 2 |
| Wednesday 9am-10am | Implement the actual logic of the function (adding a new action source file - RotateAction.java, modify apply method in RotateView.java)  Write test cases | Developer 1 and 2 |
| Wednesday 10am-11am | Code review of the action and view source files  Run the test cases | Developer 1 and 2 |
| Wednesday 11am-11:30am | Implement the functions’ integration(modify AdvancedEditingPlugin.java, integrate "Rotate" and "Resize" into the file) | Developer 1 and 2 |

## Implementation of "Resize"

|  |  |  |
| --- | --- | --- |
| **Time** | **Activity** | **Team Involved** |
| Wednesday 8am-8:30am | Implement the view of Resize function (adding a new view source file - ResizeView.java) | Developer 3 |
| Wednesday 8:30am-9am | Code review of view source file,  Checking that the view fulfils the requirements | Developer 3 and 4 |
| Wednesday 9am-10am | Implement the actual logic of the function (adding a new action source file - ResizeAction.java, modify apply method in ResizeView.java)  Write test cases | Developer 3 and 4 |
| Wednesday 10am-11am | Code review of the action and view source files  Run the test cases | Developer 3 and 4 |
| Wednesday 11:30am-12pm | Code review of AdvancedEditingPlgun.java | Developer 3 and 4 |

## Final Code Review and progress meeting

|  |  |  |
| --- | --- | --- |
| **Time** | **Activity** | **Team Involved** |
| Wednesday 1pm-2pm | Prepare 1.0.2 Plug-in Jar file  Deploy into the integration testing environment  Perform integration tests | All Developers |
| Wednesday 2pm-3pm | Alternate code review  (FlipAction.java FlipView.java RotateAction.java RotateView.java - Developer 3 and 4)  (MoveAction.java MoveView.java ResizeAction.java ResizeView.java - Developer 1 and 2) | All Developers and the Team Leader |
| Wednesday 3pm-5pm | Document the code review results  Pack the development version of plug-in jar file to 1.1-alpha version  Deploy into integration environment  Perform integration tests  Be ready to deploy into staging test environment for the following day | All Team members |

# Test procedures

|  |  |
| --- | --- |
| **Testing Typing** | **Items to be Tested** |
| White box tests/Unit Tests | Test cases of:   * AdvancedEditingPlugin.java * ImmediateEditDialogView.java * FlipAction.java * FlipView.java * RotateAction.java * RotateView.java * MoveAction.java * MoveView.java * ResizeAction.java * ResizeView.java |
| Daily integration/regression test s | Version 1.0.1 plugin jar file contains “Flip” and “Move”  Version 1.0.2 plugin jar file contains “Rotate” and “Resize”  Version 1.1-alpha plugin jar file contains “Flip”, “Move”, “Rotate” and “Resize” |
| Block box tests/System test s | Version 1.1-beta plugin jar file contains “Flip”, “Move”, “Rotate” and “Resize” |