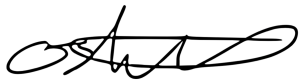
Software Development Agreement

Response Due Date: 27/02/2020

CUBIXEL

Document Approval

All authors of the document are required to proofread, mandate, and sign-off before the document's official publication.

Author	Signature	Date
Oliver Still		25/02/2020
Daniel Bishop		25/02/2020
James Gardner		25/02/2020
Che McKirgan		25/02/2020

Preface

All brands and trademarks of CUBIXEL used are registered trademarks of CUBIXEL, unless otherwise indicated. It is forbidden to use, copy, reproduce, republish, upload, forward, transmit, distribute or modify in any way brands or logos owned by CUBIXEL, without the prior written approval from the respective company itself.

This Software Development Agreement (the "**Agreement**" or "**Software Development Agreement**") states the terms and conditions that govern the contractual agreement between Goose Software Design (the "**Developer**", "**Service Provider**", "**Vendor**"), and Cubixel (the "**Company**", "**Client**") who agrees to be bound by this Agreement.

WHEREAS, the Client has conceptualized the JavaFX implementable Graphics Handler and Audio Handler Modules (the "**Software**"), which is described in further detail on Exhibit A, and the Developer is a contractor with whom the Client has come to an agreement to develop the Software.

NOW, THEREFORE, in consideration of the mutual covenants and promises made by the parties to this Software Development Agreement, the Developer and the Client (individually, each a "**Party**" and collectively, the "**Parties**") covenant and agree as follows:

Contents

1. Developer Duties

The Client hereby engages the Developer and the Developer hereby agrees to be engaged by the Client to develop the Software in accordance with the specifications and standards attached hereto as Exhibit A (the "**Specifications**").

1. The Developer shall complete the development of the Software according to the milestones described on the form attached hereto as Exhibit B. In accordance with such milestones, the final product shall be delivered to the Client by **23/04/2020** (the "**Delivery Date**").
2. For a period of six weeks after delivery of the final product, the Developer shall provide the Client attention to answer any questions or assist solving any problems with regard to the operation of the Software up to two hours a week free of charge and billed to the Client at a rate of £12.50 per hour for any assistance thereafter. The Developer agrees to respond to any reasonable request for assistance made by the Client regarding the Software within two days of the request.
3. Except as expressly provided in this Software Development Agreement, the Client shall not be obligated under this Agreement to provide any other support or assistance to the Developer.
4. The Client may terminate this Software Development Agreement at any time upon material breach of the terms herein and failure to cure such a breach within one week of notification of such a breach.
5. The Developer shall provide to the Client after the Delivery Date, a cumulative two hours of training with respect to the operation of the Software if requested by the Client.
6. A penalty fee will be enforced upon the developer where the deliverables defined in Exhibit B are perceived to be late by the client at a rate of £100 per day overdue.

2. Delivery

The Software shall function in accordance with the Specifications on or before the Delivery Date.

1. If the Software as delivered does not conform with the Specifications, the Client shall within one week of the Delivery Date notify the Developer in writing of the ways in which it does not conform with the Specifications. The Developer agrees that upon receiving such notice, it shall make reasonable efforts to correct any non-conformity.
2. The Client shall provide to the Developer written notice of its finding that the Software conforms to the Specifications within two weeks of the Delivery Date (the "**Acceptance Date**") unless it finds that the Software does not conform to the Specifications as described in Exhibit A herein.

On the defined Delivery Date, the Developer will provide the Client with all source code files and additional documentation associated with the Software on or before the Delivery Date.

3. Compensation

In consideration for the Service, the Client shall pay the Developer at the rate of £12.50 per hour. The estimated workload for the completion of deliverables is 20 hours of labour. Thus, £250.00 will be paid for to cover the Developer's variable expenditure. Including, the overhead recovery rate of the Developer at £22.47 per hour, the fee for this Software Development Agreement is paid for with the amount of £718.40 (the "**Fee**").

Of this total amount, 25% is paid upfront, 50% on the handover of the completed code and 25% when the contract is accepted as being complete by the client. All hours must be accounted for and justified by the developer. Fees billed under the paid amounts shall be due and payable upon the Developer providing the Client with an invoice.

4. Intellectual Property and Rights to the Software

The Parties acknowledge and agree that the Client will hold all intellectual property rights in the Software including, but not limited to, copyright and trademark rights. The Developer agrees not to claim any such ownership in the Software's intellectual property at any time prior to or after the completion and delivery of the Software to the Client. The Client will have the right to modify, change, update and edit the Software however the Client chooses.

5. Changes to Specifications

The Client may request that reasonable changes be made to the Specifications and tasks associated with the implementation of the Specifications. If the Client requests such a change, the Developer will use its best efforts to implement the requested change at no additional expense to the Client and without delaying delivery of the Software. In the event that the proposed change will, in the sole discretion of the Developer, require a delay in the delivery of the Software or would result in additional expense to the Client, then the Client and the Developer shall confer and the Client may either withdraw the proposed change or require the Developer to deliver the Software with the proposed change and subject to the delay and/or additional expense. The Client agrees and acknowledges that the judgment as to if there will be any delay or additional expense shall be made solely by the Developer.

6. Confidentiality

The Developer shall not disclose to any third party the business of the Client, details regarding the Software, including, without limitation any information regarding the Software's code, the Specifications, or the Client's business (the "**Confidential Information**"), (ii) make copies of any Confidential Information or any content based on the concepts contained within the Confidential Information for personal use or for distribution unless requested to do so by the Client, or (iii) use Confidential Information other than solely for the benefit of the Client.

7. Developer Warranties

The Developer represents and warrants to the Client the following:

1. Development and delivery of the Software under this Agreement are not in violation of any other agreement that the Developer has with another party.
2. The Software will not violate the intellectual property rights of any other party.
3. For a period of two years after the Delivery Date, the Software shall operate according to the Specifications. If the Software malfunctions or in any way does not operate according to the Specifications within that time, then the Developer shall take any reasonably necessary steps to fix the issue and ensure the Software operates according to the Specifications.

8. Indemnification

The Developer agrees to indemnify, defend, and protect the Client from and against all lawsuits and costs of every kind pertaining to the software including reasonable legal fees due to the Developer's infringement of the intellectual rights of any third party.

9. Agreement Modification

No modification of this Agreement shall be valid unless in writing and agreed upon by both Parties.

10. Service Agreement

The following signatures confirm that an agreement between the Developer and the Project Manager, on behalf of the Company (CUBIXEL), has been made for the Client to provide the Developer with the proposed Fee defined in this document to proceed with the development of the software.

In addition, both the Client and Developer agree to the terms laid out in section 6 to non-disclose all information relating to the project and company operations.

Payments of the proposed Fee will be made at an agreed time following the completion of the Software and handover of all deliverables as detailed in Exhibit B.

Signed by:

OLIVER STILL

for and on behalf of

THE CLIENT

Signature:

Date:

27/02/2020

Signed by:

ALEX KNELLER

for and on behalf of

THE DEVELOPER

Signature:

Date:

27/02/2020

Signed by:

TONY WARD

for and on behalf of

THE CLIENT'S SUPERVISOR

Signature:

Date:

27/02/2020

Exhibit A - Software Specification

I. Audio Handler

Overview

The Audio Handler Module's function is to play audio files streamed to it from a specified URL or local file path. It should provide the ability to loop the audio file once it has finished, and to stop a currently playing audio file.

Functional Requirements

Constructors

No arguments are required for the class constructor.

Methods

playAudioFile

Argument	Type	Description
url	String	URL for the source of the Audio File.
looping	Boolean	If true, the audio should loop back to the start after finishing
id	String	ID assigned to the audio instance (used to later stop playback of a looping file/stop a file during playback). If the requested ID is already in use, then the method should refuse the request.

stopPlayback

Argument	Type	Description
id	String	Stop playback of the specified audio file

II. Graphics Handler

Overview

The Graphics Handler Module's function is to draw JavaFX Graphics objects to individual JavaFX canvases and add those canvases to a JavaFX StackPane object, storing a reference to the added canvas (linked to an ID) so that each canvas can be individually removed as required. The Graphics Handler will take a specific set of inputs as detailed below and draw a Line, Rectangle or Oval Object, or remove a previously drawn object. The drawRectangle and drawOval methods should be overloaded to allow for either a single colour or a gradient fill.

Functional Requirements

Constructors

The Graphics Handler should provide the option to draw a line, rectangle and a circle via separate methods. The Graphics Handler should take as argument:

Argument	Type	Description
targetPane	StackPane (JavaFX)	This is the StackPane object onto which the Graphics Handlers methods should draw.

Methods

drawLine

Argument	Type	Description
xStart	float	X-coordinate for the starting point of the line, as a percentage of the width of the pane from the left edge.
xEnd	float	X-coordinate for the end point of the line, as a percentage of the width of the pane from the left edge.
yStart	float	Y-coordinate for the starting point of the line, as a percentage of the height of the pane from the top edge.
yEnd	float	Y-coordinate for the end point of the line, as a percentage of the height of the pane from the top edge.

lineColor	String	The colour of the line, in the form '#ffffff' where each pair of characters after the hash are the hexadecimal representation of red, green, blue respectively.
id	String	The ID assigned to the drawn shape, used to later remove it. If the requested ID is already in use, then the method should refuse the request.

drawRectangle

Argument	Type	Description
xStart	float	X-coordinate for the top left point of the rectangle, as a percentage of the width of the pane from the left edge.
yStart	float	Y-coordinate for the top left point of the rectangle, as a percentage of the height of the pane from the top edge.
width	float	As a percentage of the width of the pane.
height	float	As a percentage of the height of the pane.
fillColour	String	The colour with which to fill the rectangle, with the form '#ffffff' where each pair of characters after the hash are the hexadecimal representation of red, green, blue respectively. This is a block fill of the entire shape as one colour.
id	String	ID assigned to the drawn shape, used to later remove it. If the requested ID is already in use, then the method should refuse the request.

drawOval

Argument	Type	Description
xStart	float	X-coordinate for the top left point of the oval, as a percentage of the width of the pane from the left edge.
yStart	float	Y-coordinate for the top left point of the oval, as a percentage of the height of the pane from the top edge.
width	float	As a percentage of the width of the pane.
height	float	As a percentage of the height of the pane.
fillColour	String	The colour with which to fill the oval, with the form '#ffffff' where each pair of characters after the hash are the hexadecimal representation of red, green, blue respectively. This is a block fill of the entire shape as one colour.
id	String	ID assigned to the drawn shape, used to later remove it. If the requested ID is already in use, then the method should refuse the request.

drawRectangle

Argument	Type	Description
xStart	float	X-coordinate for the top left point of the rectangle, as a percentage of the width of the pane from the left edge.
yStart	float	Y-coordinate for the top left point of the rectangle, as a percentage of the height of the pane from the top edge.
width	float	As a percentage of the width of the pane.

height	float	As a percentage of the height of the pane.
id	String	ID assigned to the drawn shape, used to later remove it. If the requested ID is already in use, then the method should refuse the request.
shading_x1	float	X-coordinate for the anchor point of colour 1 as a percentage of the width of targetPane
shading_y1	float	Y-coordinate for the anchor point of colour 1 as a percentage of the width of targetPane
shading_colour1	String	The colour anchored to the first specified point, with the form '#ffffff' where each pair of characters after the hash are the hexadecimal representation of red, green, blue respectively.
shading_x2	float	X-coordinate for the anchor point of colour 2 as a percentage of the width of targetPane
shading_y2	float	Y-coordinate for the anchor point of colour 2 as a percentage of the width of targetPane
shading_colour2	String	The colour anchored to the second specified point, with the form '#ffffff' where each pair of characters after the hash are the hexadecimal representation of red, green, blue respectively.
shading_cyclic	Boolean	True if the gradient pattern should cycle repeatedly between the two colors; False otherwise (see java.awt.GradientPaint)

drawOval

Argument	Type	Description
xStart	float	X-coordinate for the top left point of the oval, as a percentage of the width of the pane from the left edge.
yStart	float	Y-coordinate for the top left point of the oval, as a percentage of the height of the pane from the top edge.
width	float	As a percentage of the width of the pane.
height	float	As a percentage of the height of the pane.
id	String	ID assigned to the drawn shape, used to later remove it. If the requested ID is already in use, then the method should refuse the request.
shading_x1	float	X-coordinate for the anchor point of colour 1 as a percentage of the width of targetPane
shading_y1	float	Y-coordinate for the anchor point of colour 1 as a percentage of the width of targetPane
shading_colour1	String	The colour anchored to the first specified point, with the form '#ffffff' where each pair of characters after the hash are the hexadecimal representation of red, green, blue respectively.
shading_x2	float	X-coordinate for the anchor point of colour 2 as a percentage of the width of targetPane
shading_y2	float	Y-coordinate for the anchor point of colour 2 as a percentage of the width of targetPane

shading_colour2	String	The colour anchored to the second specified point, with the form '#ffffff' where each pair of characters after the hash are the hexadecimal representation of red, green, blue respectively.
shading_cyclic	Boolean	True if the gradient pattern should cycle repeatedly between the two colors; False otherwise (see java.awt.GradientPaint)

removeGraphic

Argument	Type	Description
id	String	The id of the shape to be removed

III. Coding Standards

All code produced must meet the coding standards as laid out in the companies style guide, which can be found via the link below or else will be potentially rejected for revision, or else will be potentially rejected for revision.

https://google.github.io/styleguide/javaguide.html

Exhibit B - Milestone Schedule

Milestone	Delivery Deadline
Draft Contract Provided to Developer	26/02/2020
Proposal of Changes to Draft Contract	26/02/2020
Contract Agreed	27/02/2020
Client Demo	20/04/2020
Final Handover of all Deliverables	23/04/2020