lively

Design and Standards

MDM Students:
Alexia Lou - Camila Burbano - Camila Serrano - Hang (Henry) Wu - Luan Jiang - Mahdiyar Biazi

UVIC students:
Matt Hemmings - Voltaire Bazurto - Riz Panjwani

MDM Lead Faculty:
Dave Fracchia - Dan Scott

Client - SAP Researchers:
Dan Ingalls - Rick McGeer - Marko Roeder

Content

Lively Design and Standards

Introduction

The Lively Web team at the CDM, supported by students at UBC and UVic, worked with researchers at SAP to **improve and streamline the Lively Web User Experience**. The Lively Web is a browser-based runtime and development environment with graphical composition written fully in JavaScript using standard browser graphics. Along with its application development abilities, the Lively Web can also function as its own integrated development environment (IDE), making it completely self-sufficient.

Purpose

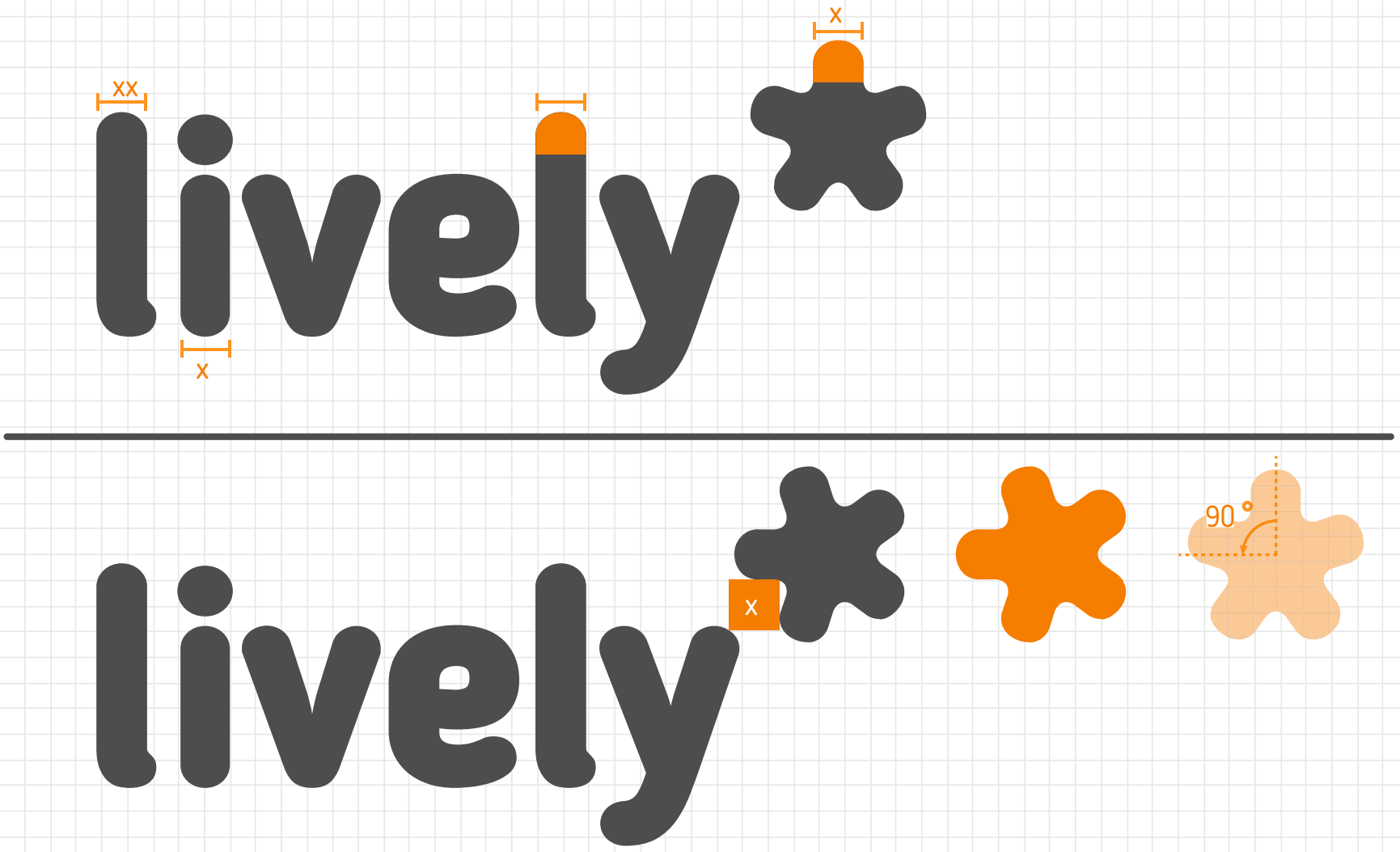
The purpose of this project was to **ensure that Lively Web users will have a streamlined experience**, one which encourages creativity and productivity through ease of use and access to all the available tools and functions within the Lively Web Environment.

Logo / Branding

Logo Design Principles

In the new proposed branding identity, we focused on a more friendly Lively experience. The new logo is created with a sans-serif typeface with rounded corners. The rounded elements were used to create a **unified design that were consequently applied to other design elements** including the entire user interface, menus and windows.

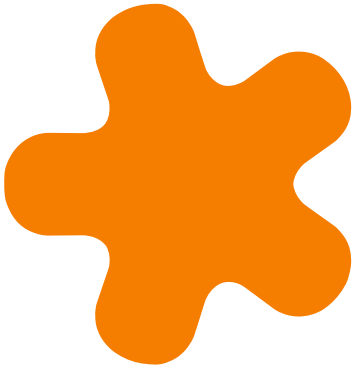
Logo proportions



Star Design Principles

The “Lively Star” implies the following:

- In English, the **asterisk** is usually a five-pointed shape created with a sans-serif typeface
- The five-pointed shape resembles an **abstract little man** that is open to the world.
- In computer science, the asterisk is commonly used as a wildcard character, or to denote pointers, repetition, or multiplication and is often simply referred to as the **“star”.**

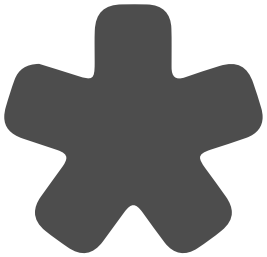


star
asterisk
little man

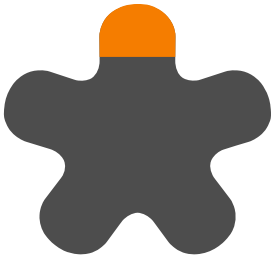
Star / Asterisk / Little Man

original star with 90° degrees rotation

proportional length to font length line



rounded borders as logo font



Minimum Size and Margins

To ensure the logo is easy to recognize, **it should not be smaller than the measurement shown below**. Additionally, the minimum margin should be as big as the size of the star. This is to guarantee that the logo will not be hidden under any other shape, font or element.

Web: 100 x 46.8 pixels
Print: 3.6 x 1.7 cm

Correct Logo Usage

lively*

Correct Complementaty Logo Usage

lively*forum

*forum

Incorrect Logo Usage

No additional elements

No rotations

Use Lively's orange colour only

Colour

The reason behind using a bright orange colour is to **leave a distinct impression in user's mind**; one that implies liveliness and creativity. The lively orange colour is contrasted with the dark grey colours used in designing the user interface.

#f57c00

R:245
G:124
B:0

#4d4d4d

R:77
G:77
B:77

Font

Blogger Sans is a **friendly and welcoming sans-serif font with t borders.** This is a clear and legible font with bolder weights that make it suitable for titles.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

1 2 3 4 5 6 7 8 9 0

User Experience

Lively User Experience Design Principle

During our discovery phase, we decided to try a new concept; one that will allow us to carefully design the user experience without limiting creativity. It was necessary to have a guideline on how we wanted to manage the new lively web experience. The guideline that we followed to create this user experience has four different stages:

Engage: We aimed to engage and inspire new users to create new projects.

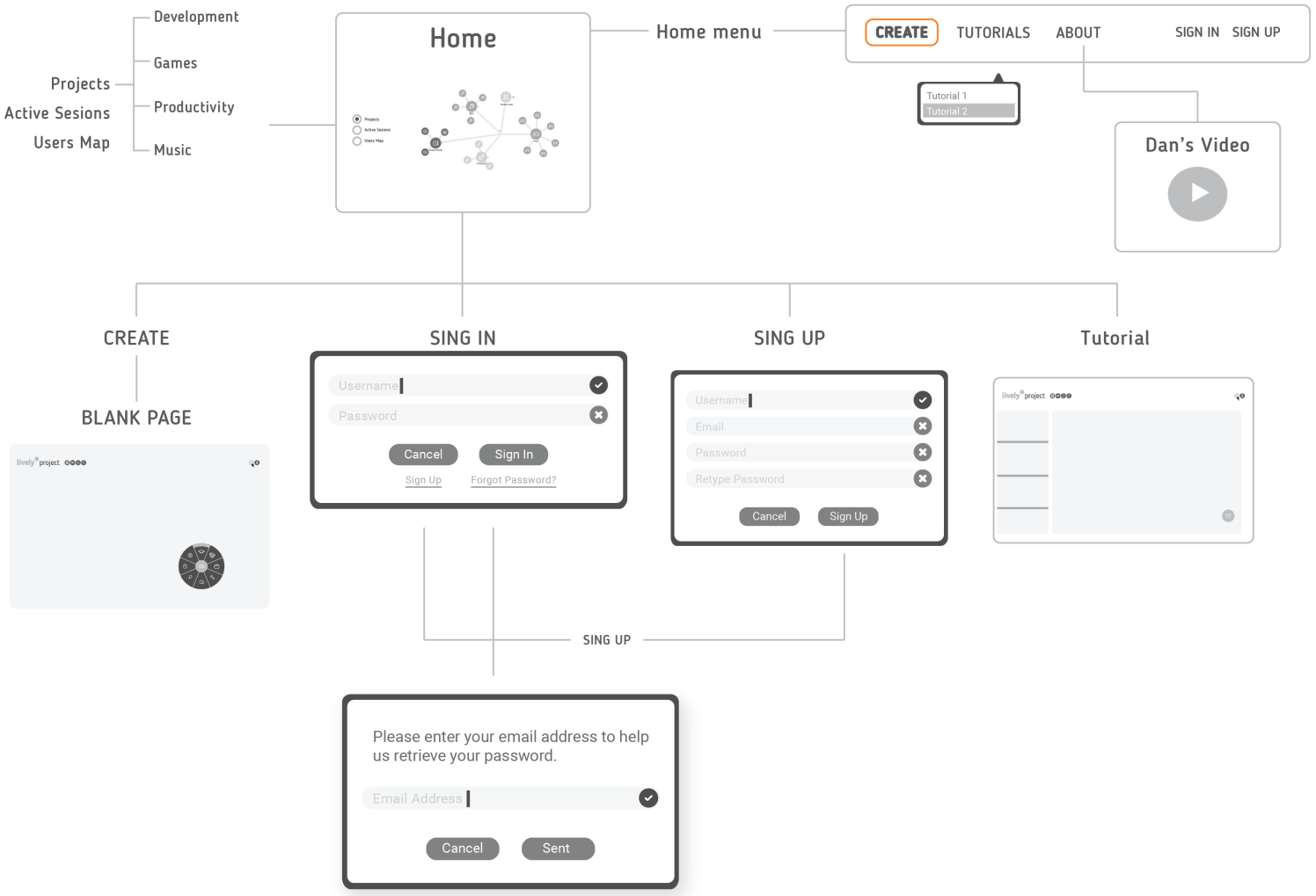
Learn: With an easy to use interface, users will have the chance to create and learn.

Play: The next step after learning the basics of Lively web, will be to create projects and discover new tools and options. The workspace combined with fun tutorials will help teach the basics of Lively and keep the users engaged at the same time.

Share: To close the cycle, the Lively web users will be able to share their projects on the public gallery so they can inspire and engage future users.

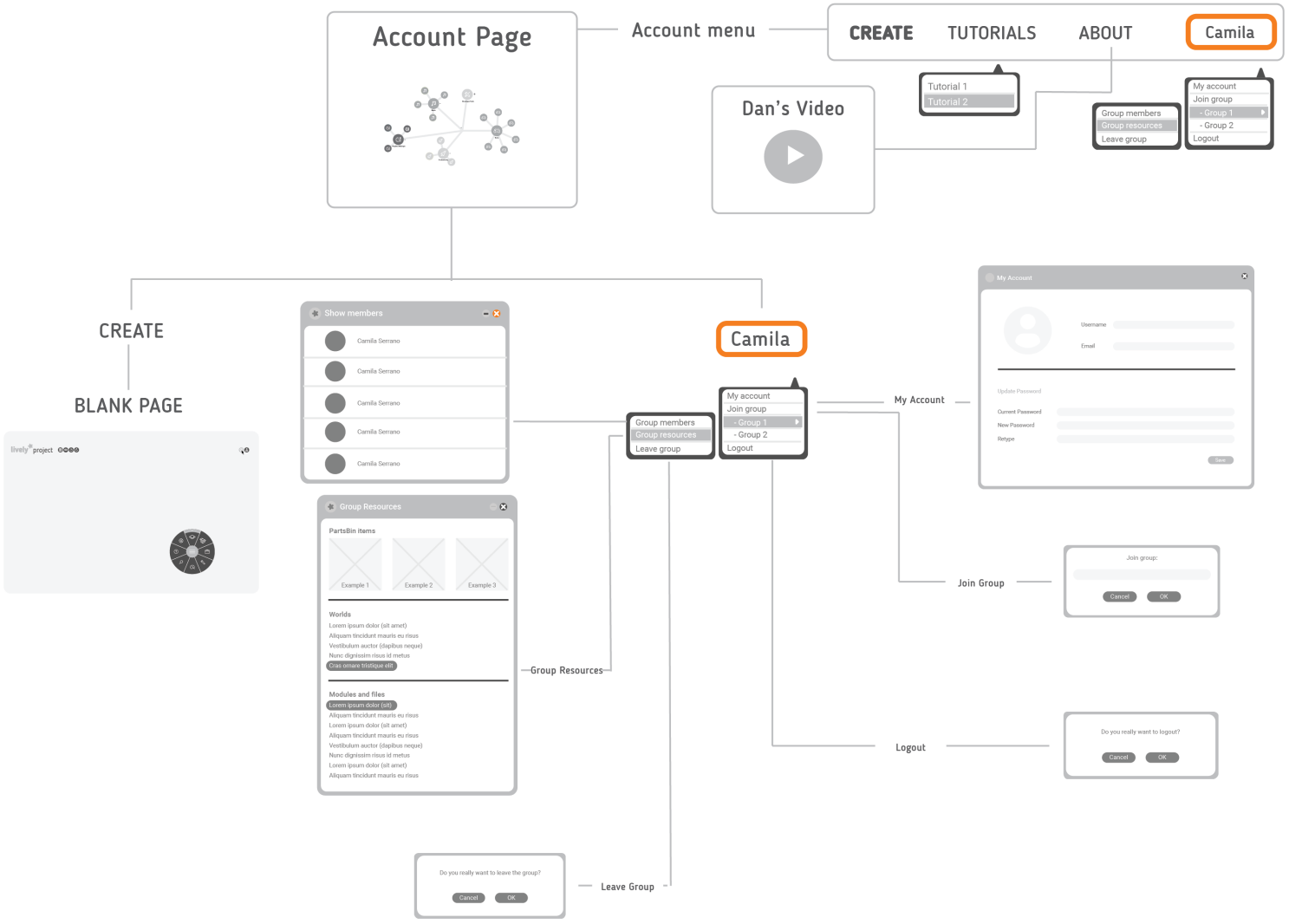
Homepage

An attractive gallery with different projects will provide the new users with different options and tools to interact with the Lively web. The visualization of the projects will reinforce a sense of community. Upon exploring the projects' gallery, the user can immediately begin creating by clicking on the "create" menu button. The users should be able to create a new project without having to sign in. To save their work, they are required to sign up by creating a user account from the homepage (sign in menu button).



Account Page

The account page is a visual representation of all the projects and parts created by the user. This page is visually consistent with the visualization on the homepage galleries. In future, the users should be able to create a new project, change their account information, join groups and manage group resources.



Workspace

The reorganization of the tools in the new workspace was a key factor in this project. The workspace is organized in a way to avoid overwhelming the first-time users. The workspace is comprised of two parts:

Top Menu

The top menu has the following options:

1. **Change project name**
2. **Save:** In future, a message should appear when the user has been using lively without saving changes as a reminder to save.
3. **Preview:** In future iterations, this option should allow the user to see the project without any menu or window.
4. **Undo and Redo**
5. **Chat:** The user will be able to start a chat session with another user.
6. **User Account:** In future iterations, this icon should turn “green” or “red” to replace the current connected/ disconnected network status indicator.

Pie Menu

Lively web users should be able to discover Lively features and menu options with relative ease. To enhance the current and future Lively web user experience, we decided to introduce a pie menu that is easy to interact with and contain the most important tools in Lively. The new pie menu functions as effectively on tablets as it does on desktop version of Lively. We chose this simple design mainly for its intuitive interactive experience and made sure that it has a straightforward design. The new pie menu reorganizes the lively tools into eight categories.

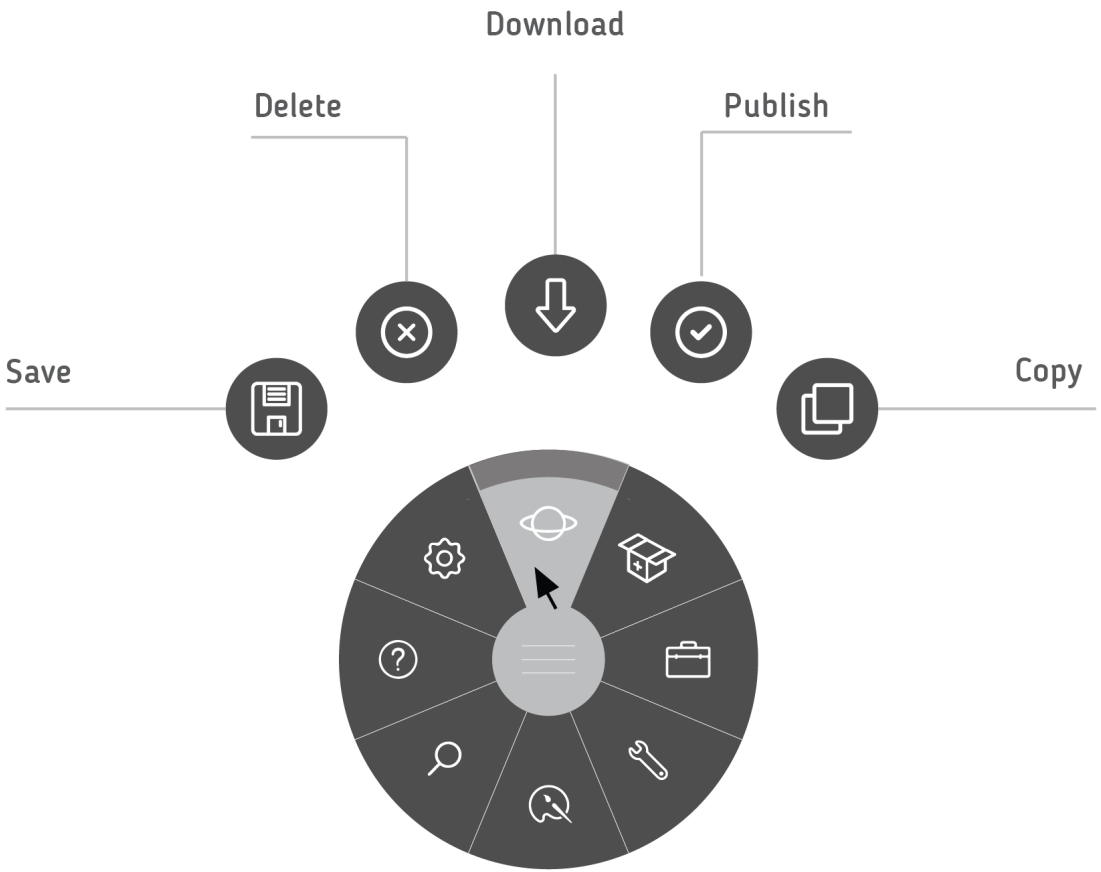

Workspace
Workspace menu
lively project
World Title
preview
undo
redo
Save
Chat
User
World Name:
Cancel
Save
Save Changes to World Name
Cancel
Save
Select member to chat with
Add members to chat
Camila Serrano
Elizabeth Carr
Omar Martinez
Matt Solinsky
Omar Martinez
Group Resources
PartsBin items
Example 1
Example 2
Example 3
Worlds
Modules and files
My account
Join group
- Group 1
- Group 2
Logout
Group members
Group resources
Leave group
Account Page
Join group:
Cancel
OK
Do you really want to leave the group?
Cancel
OK
Do you really want to logout?
Cancel
OK
Voltaire Bazurto
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Ut vitae tincidunt neque. Nullam pellentesque ex in ante lellentesqobo
Ut vitae tincidunt neque. Nullam pellentesque ex...
Show members
Camila Serrano
Camila Serrano
Camila Serrano
Camila Serrano
Camila Serrano

Information Architecture

Lively web tools are reorganized into eight categories. We decided to keep the design simple and apply a standard approach to help the user find the tools they need in the workspace. This was to ensure minimum effort is needed to discover and use Lively web.

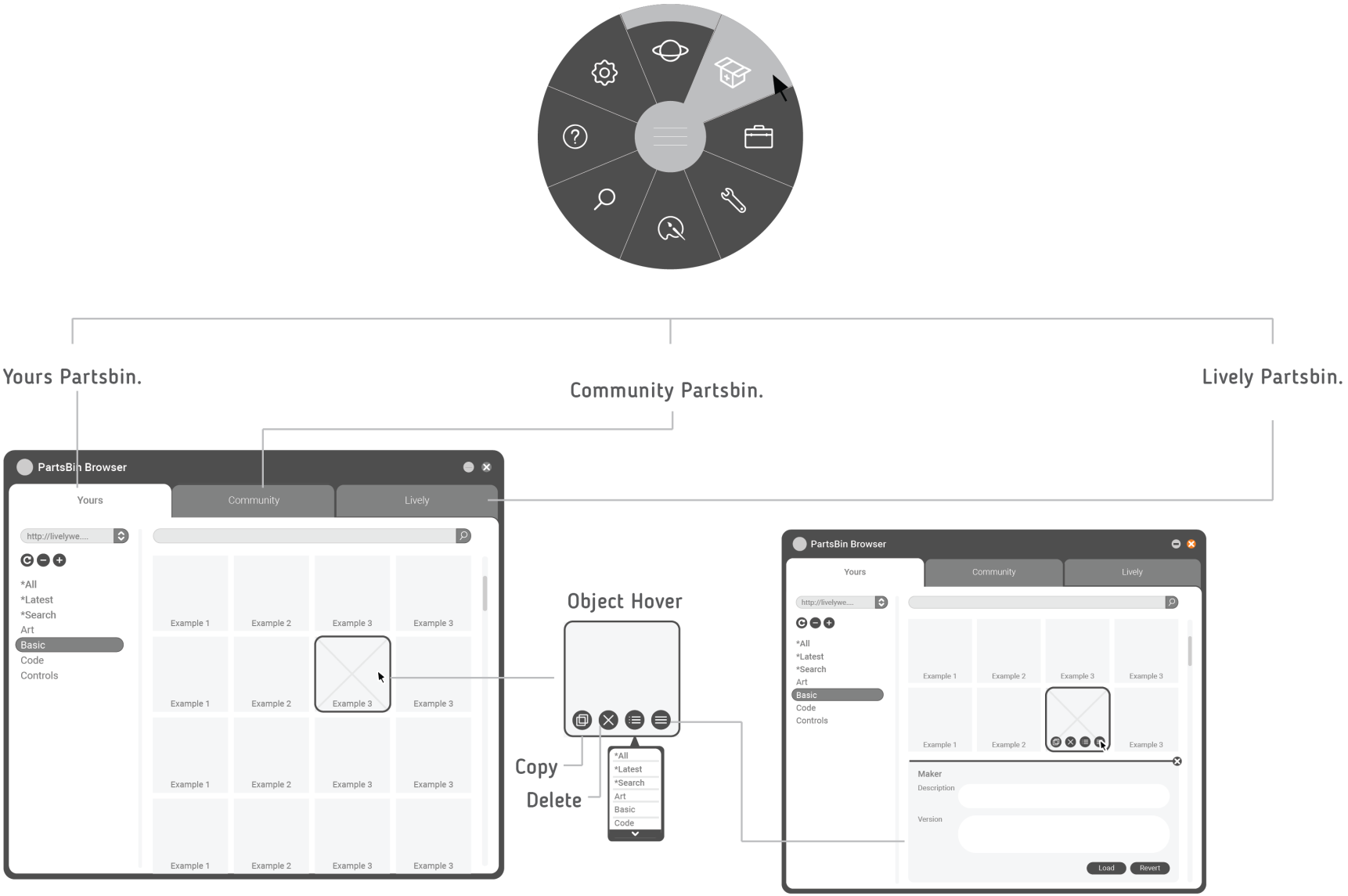
Minimized Wheel

World	Partsbin	Advance Tools	Object Editor	Style Editor	Search	Help	Settings
Save. Copy. Delete. Download.	Yours Partsbin. Community Partsbin. Lively Partsbin.	Test Runner. Git Control. System Console. Server Workspace. File Tree. Terminal.	JS Workspace. Morphs Tree. Script, Connections and Properties.	View CSS Layout Text Custom	Search Code. Search world, modules and parts. Search Files. Search in Files.	Cheat Sheet. Tutorials. Documentation. Play Quick Starter.	Account Configuration. World Configuration. Preferences.


Download
Delete
Publish
Save
Copy

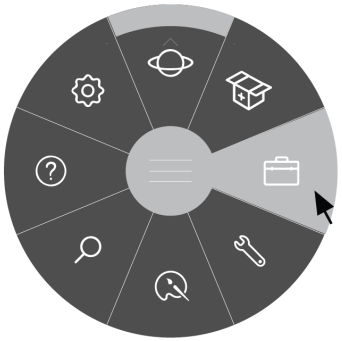
2. Partsbin

Based on our analysis of the PartsBin, we proposed that in future iterations, it should contain three categories. These include a default partsbin called “Lively partsbin”, a user’s partsbin called “Yours” as well as a community partsbin which is comprised of objects, projects or widgets that other lively users have created.

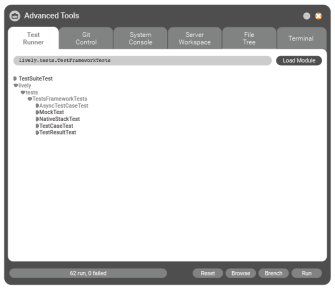


3. Advanced Tools

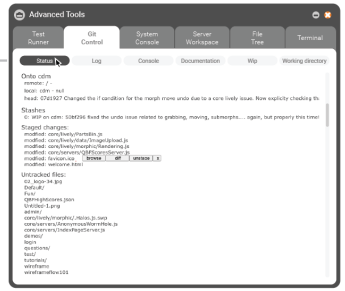
The advanced tools window includes all the tools that are primarily used by a developer. This category includes six tools:



Test Runner

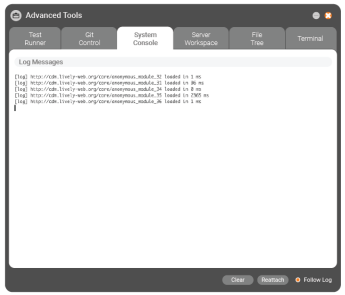


Git Control

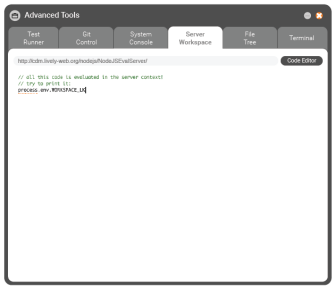


-Status
-Log
-Console
-Documentation
-Wip:
 -Quick Publish,
 -Workspace
 -To do List
-Working Directory

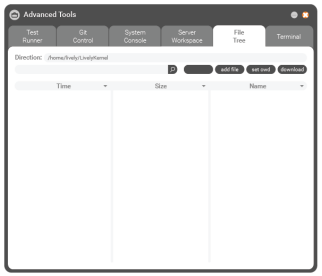
System Console



Server Workspace



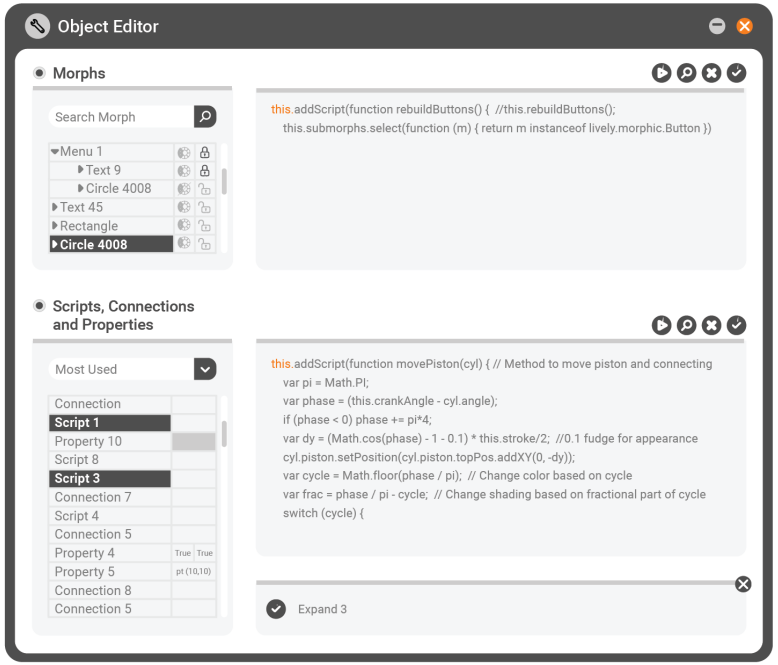
File Tree



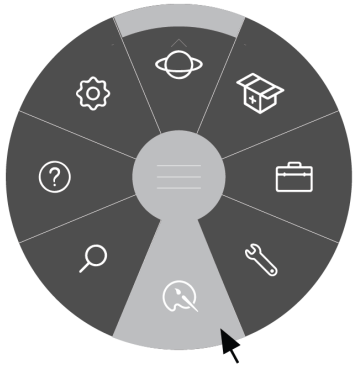
Terminal

4. Object Editor

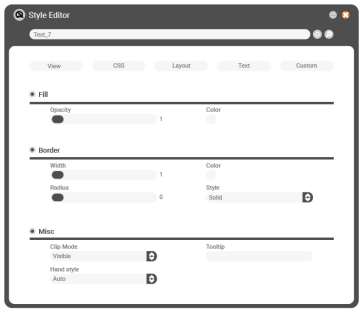
In this project the object editor and inspector were combined into one window. This allows the user to find all the information they need in one place and work more efficiently. The object editor allows the user to lock or unlock a morph and enable or disable the halo of a morph. Scripts, connections and properties can be filtered by the default tag; alternatively, the user can create and store a tag. Moreover, the users will be able to modify properties without having to write any code.

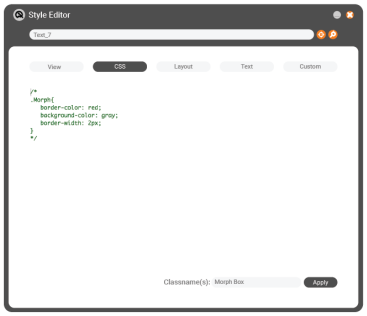


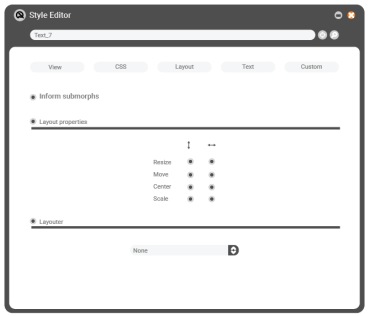
5. Style Editor

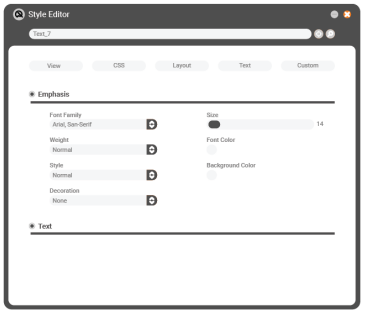


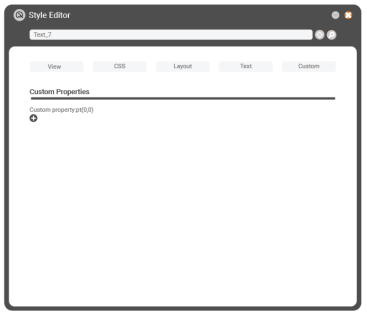
View CSS Layout Text Custom





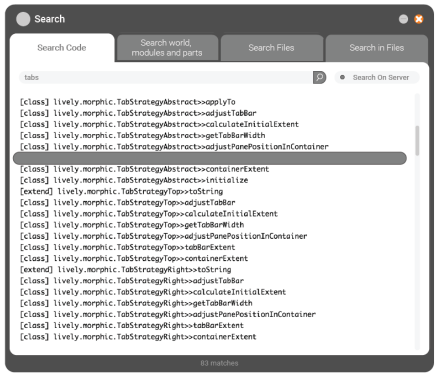




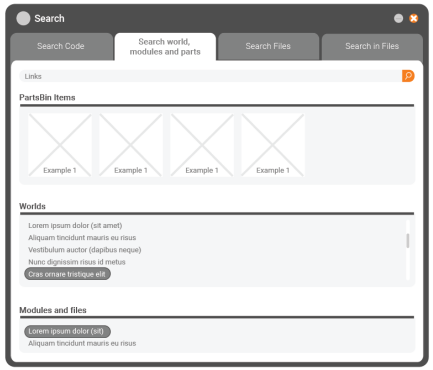


6. Search

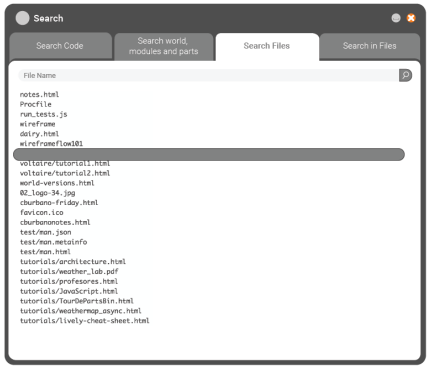
Search Code



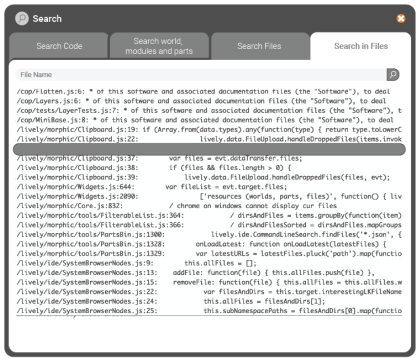
Search world, modules and parts



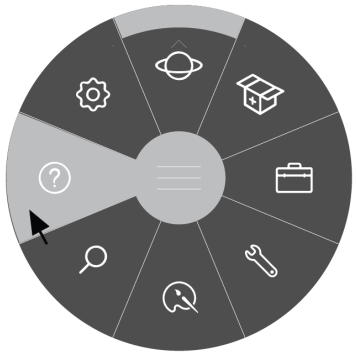
Search Files



Search in Files



7. Help

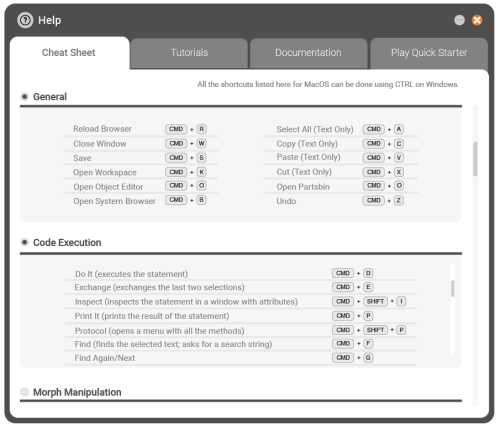


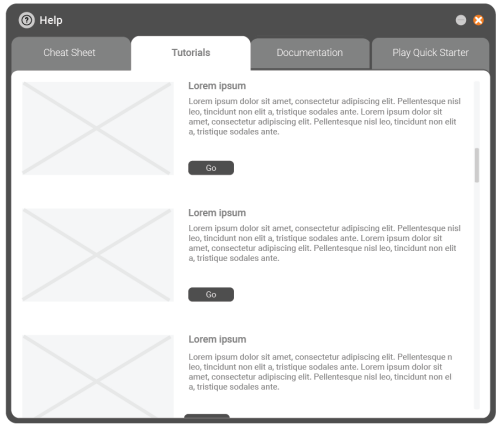
Cheat Sheet

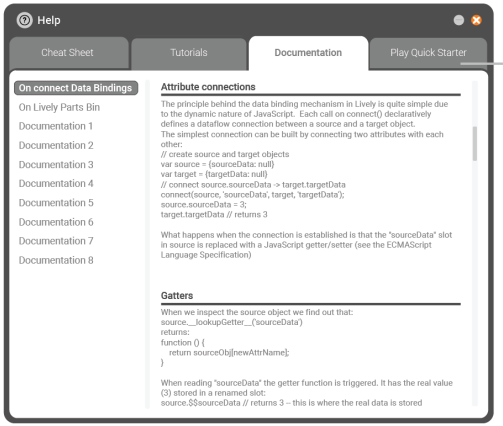
Tutorials

Documentation

Play Quick Starter

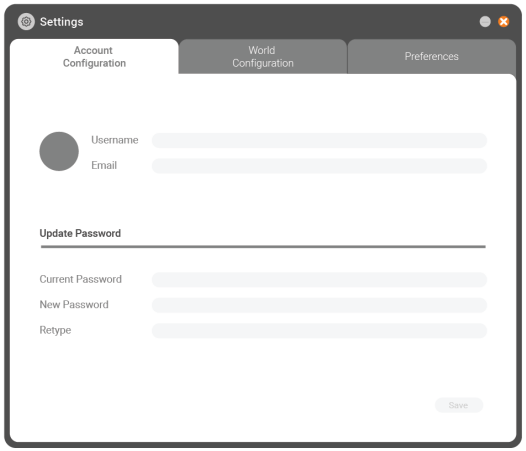




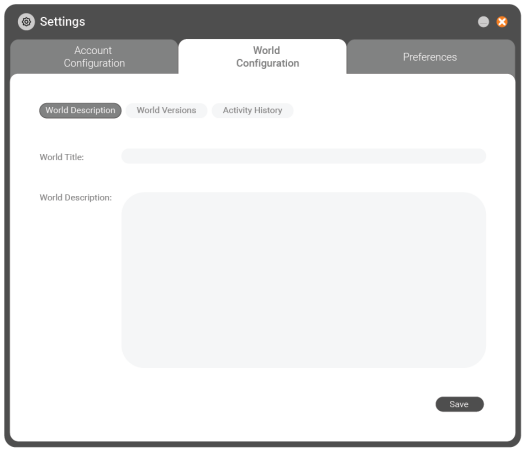


8. Settings

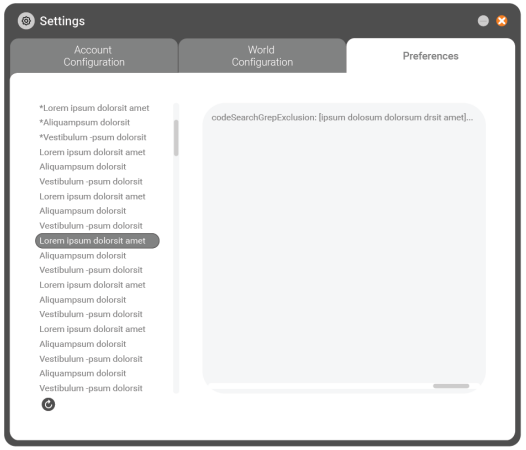
Account configuration



World Configuration



Preferences



User Interface

UI Design Principles

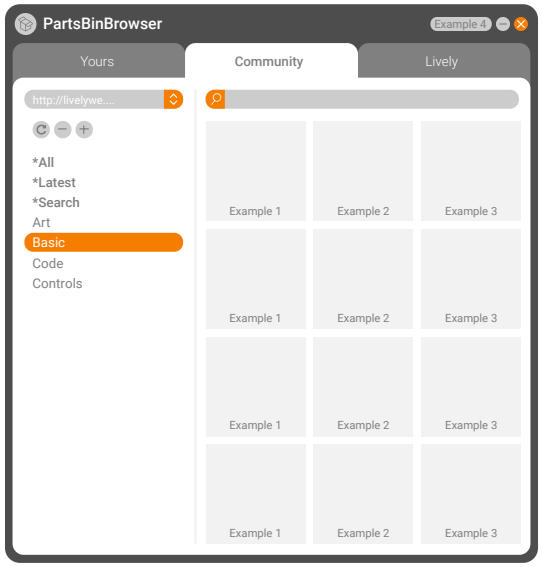
An important part of the process of creating a new experience for the user, was improved esthetics, as well as easy-to-understand elements. Users interaction was a core component of this project.

Components Specifications

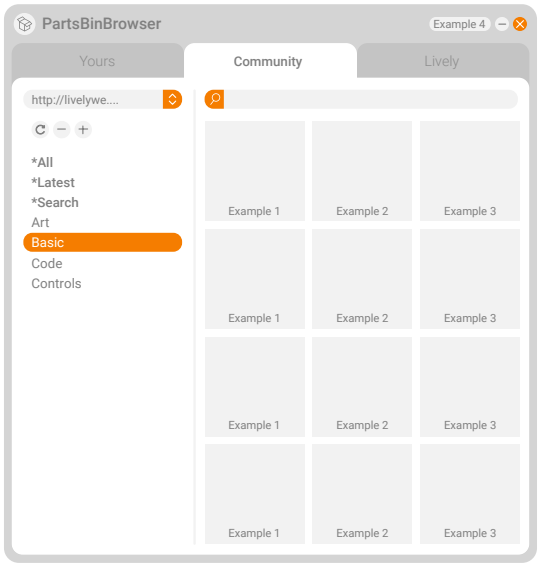
An important part of the process of creating a new experience for the user, was improved esthetics, as well as easy-to-understand elements. Users interaction was a core component of this project.

1. Active / Inactive Windows

It is important to provide clear feedback to the user. Advanced users might have more than one window open at once. We made sure that active and inactive windows use two different colours.
Active Windows have a darker interface that create a contrast making text easy to read. Inactive Windows have a lighter interface.



Active Windows



Inactive Windows

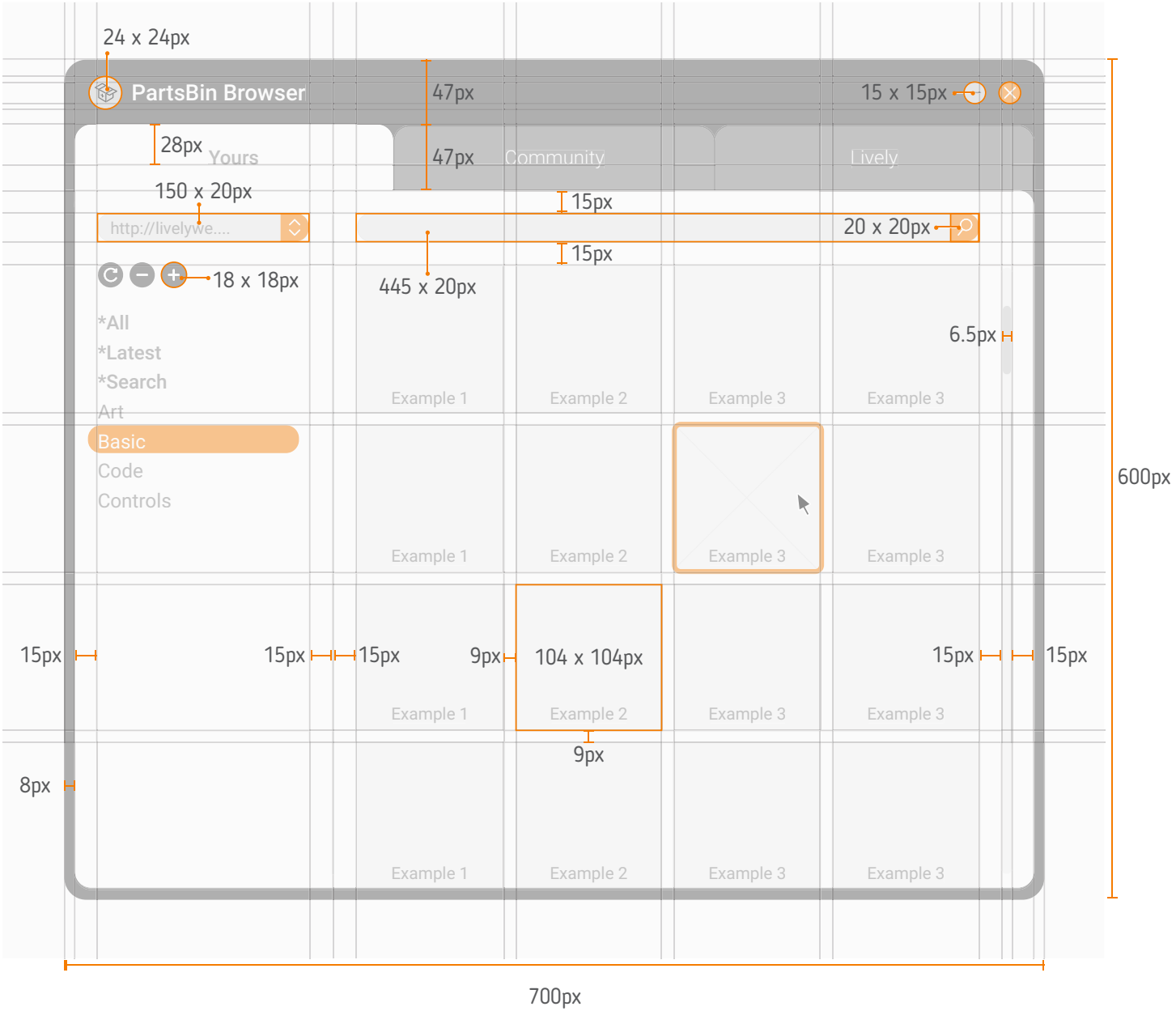
2. Main Windows

There are 7 Main Windows for each of the sections in the Wheel Menu:

Partsbin | Advance Tools | Object Editor | Style Editor | Search | Help | Settings

The main purpose of these windows is to collapse the secondary windows. As such, we needed to show the higher hierarchy by using a darker colour tone.

- Main Windows Specifications



- Main Windows Headings

H1 **HEADING 1**
font-family : Roboto; Medium
font-size : 16px;
color : #FFFFFF;
color : rgb(255, 255, 255);

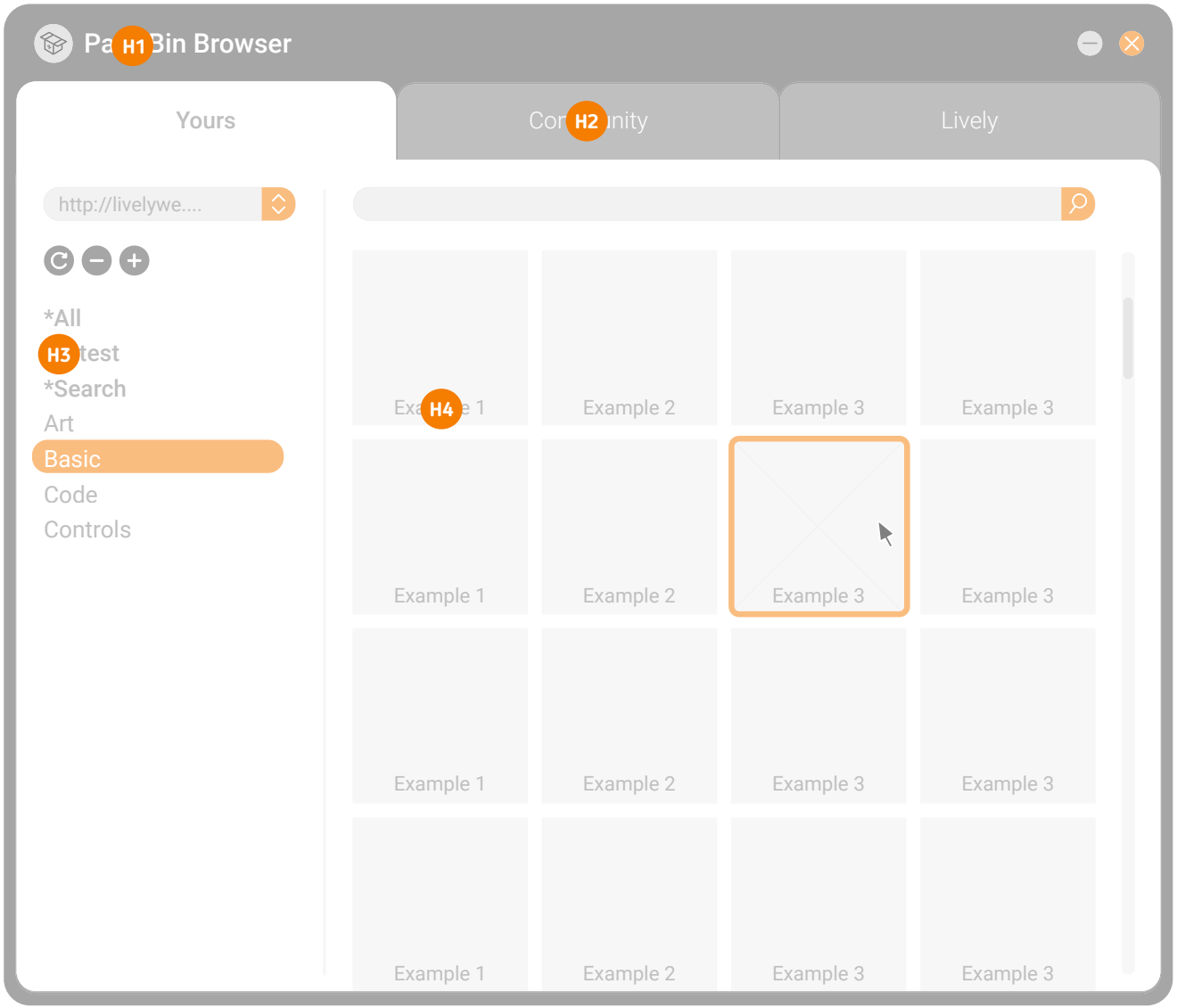
H2 **HEADING 2 – Highlighted**
font-family : Roboto; Medium
font-size : 14px;
color : #808080;
color : rgb(128, 128, 128);

HEADING 2 – Not Highlighted
font-family : Roboto; Light
font-size : 14px;
color : #FFFFFF;
color : rgb(255, 255, 255);

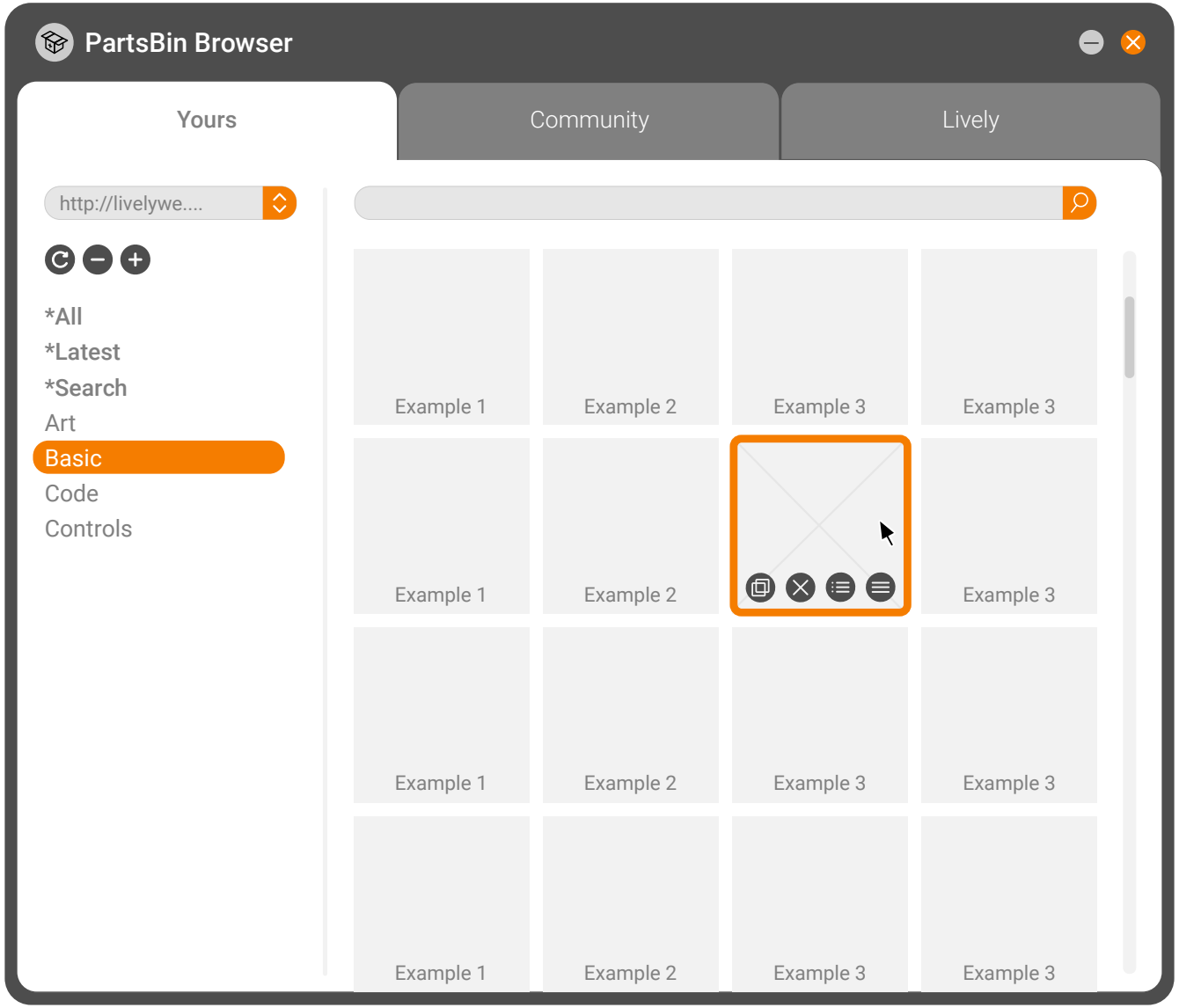
H3 **HEADING 3 – Highlighted**
font-family : Roboto; Medium
font-size : 14px;
color : #808080;
color : rgb(128, 128, 128);

HEADING 3 – Not Highlighted
font-family : Roboto; Regular
font-size : 14px;
color : #808080;
color : rgb(128, 128, 128);

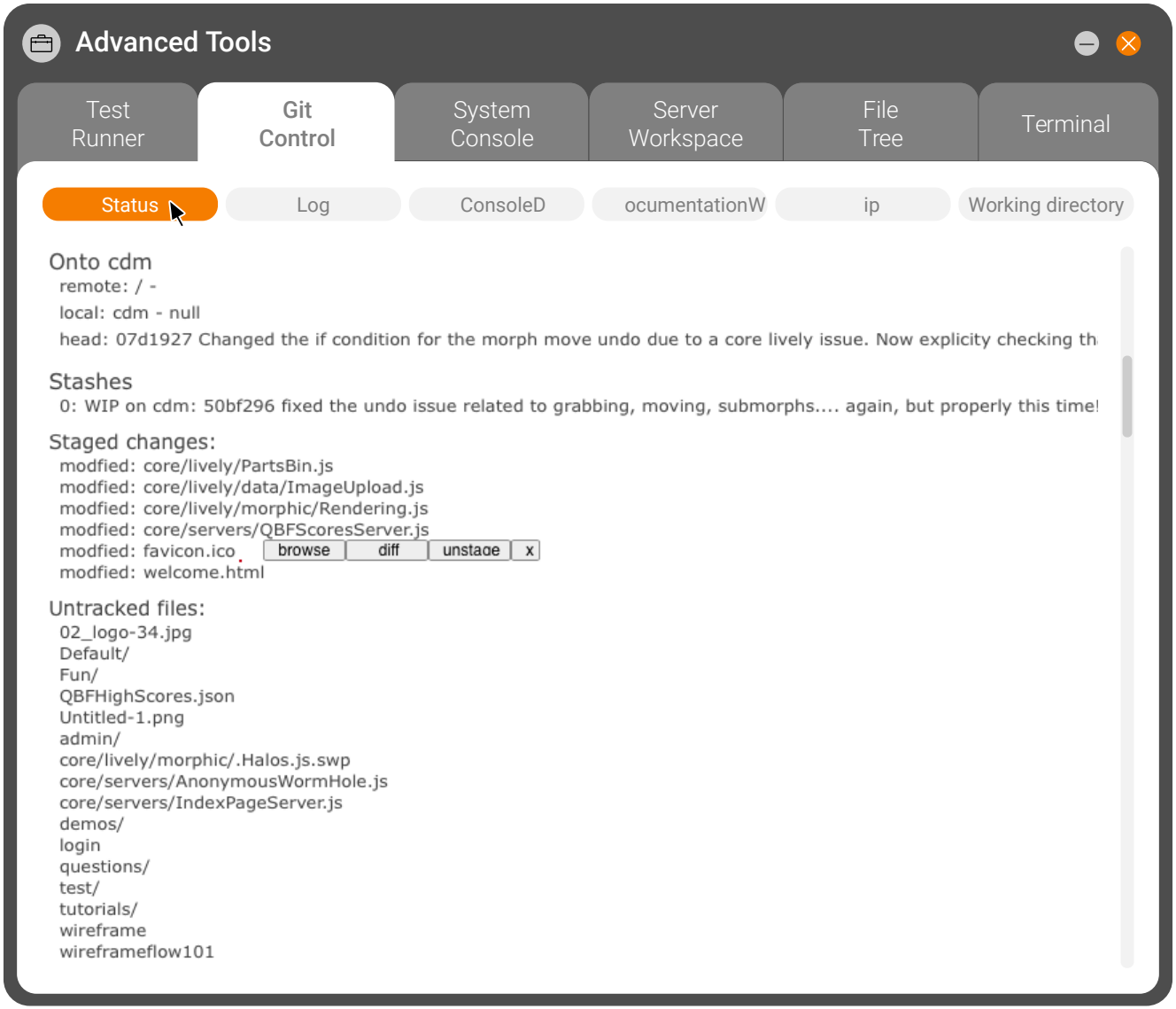
H4 **HEADING 4**
font-family : Roboto; Regular
font-size : 12px;
color : #808080;
color : rgb(128, 128, 128);



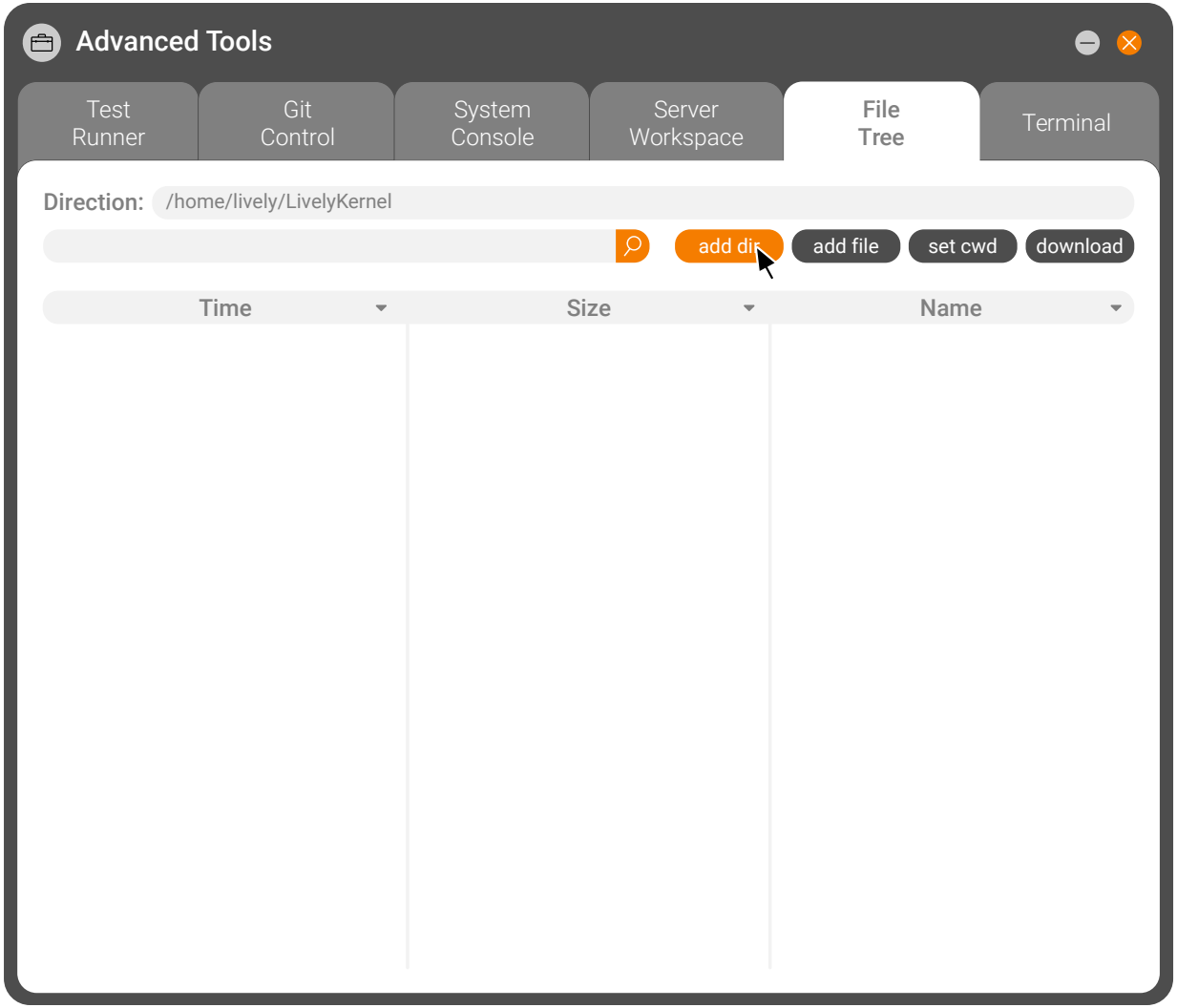

Partsbin
Advance Tools
Object Editor
Style Editor
Search
Help
Settings

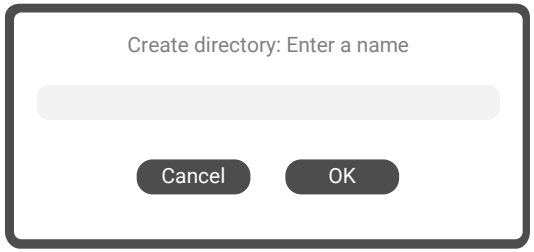

PartsBin Browser
Yours
Community
Lively
http://livelywe....
*All
*Latest
*Search
Art
Basic
Code
Controls
Example 1
Example 2
Example 3
Example 3
Example 1
Example 2
Example 3
Example 1
Example 2
Example 3
Example 3
Example 1
Example 2
Example 3
Example 3

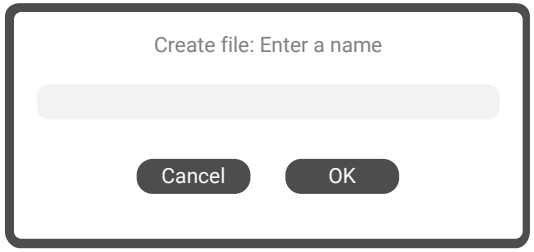

Partsbin
Advance Tools
Object Editor
Style Editor
Search
Help
Settings


Advanced Tools
Test Runner
Git Control
System Console
Server Workspace
File Tree
Terminal
Status
Log
ConsoleD
ocumentationW
ip
Working directory
Onto cdm
remote: / -
local: cdm - null
head: 07d1927 Changed the if condition for the morph move undo due to a core lively issue. Now explicity checking th
Stashes
0: WIP on cdm: 50bf296 fixed the undo issue related to grabbing, moving, submorphs.... again, but properly this time!
Staged changes:
modfied: core/lively/PartsBin.js
modfied: core/lively/data/ImageUpload.js
modfied: core/lively/morphic/Rendering.js
modfied: core/servers/QBFScoresServer.js
modfied: favicon.ico
browse
diff
unstage
x
modfied: welcome.html
Untracked files:
02_logo-34.jpg
Default/
Fun/
QBFHighScores.json
Untitled-1.png
admin/
core/lively/morphic/.Halos.js.swp
core/servers/AnonymousWormHole.js
core/servers/IndexPageServer.js
demos/
login
questions/
test/
tutorials/
wireframe
wireframeflow101


Partsbin
Advance Tools
Object Editor
Style Editor
Search
Help
Settings


Advanced Tools
Test Runner
Git Control
System Console
Server Workspace
File Tree
Terminal
Direction:
/home/lively/LivelyKernel
add dir
add file
set cwd
download
Time
Size
Name


Create directory: Enter a name
Cancel
OK


Create file: Enter a name
Cancel
OK

Object Editor

Morphs

Search Morph

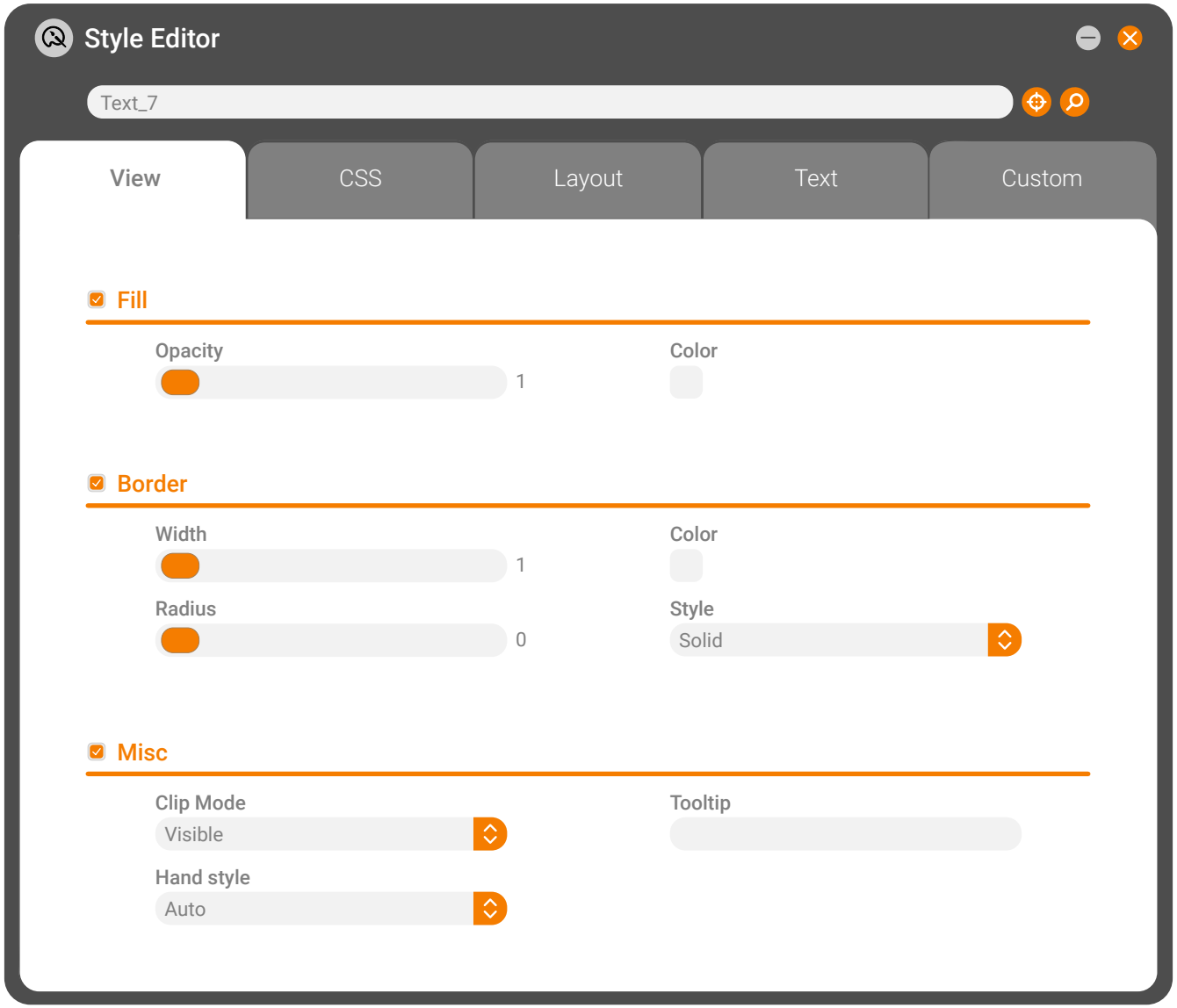
- Menu 1
  - Text 9
  - Circle 4008
- Text 45
- Rectangle
- **Circle 4008**
- Menu 1
  - Text 9
  - Circle 4008
- Text 45
- Rectangle
- **Rectangle 002**
- Menu 1
  - Text 9
  - Circle 4008
- Text 45
- Rectangle
- Circle 4008
- Menu 1
  - Text 9
  - Circle 4008
- Text 45
- Rectangle

```
this.addScript(function rebuildButtons() {  //this.rebuildButtons();
  this.submorphs.select(function (m) { return m instanceof lively.morphic.Button })
  var pi = Math.PI;
  var phase = (this.crankAngle - cyl.angle);
  if (phase < 0) phase += pi*4;
  var dy = (Math.cos(phase) - 1 - 0.1) * this.stroke/2;  //0.1 fudge for appearance
  cyl.piston.setPosition(cyl.piston.topPos.addXY(0, -dy));
  var cycle = Math.floor(phase / pi);  // Change color based on cycle
  var frac = phase / pi - cycle;  // Change shading based on fractional part of cycle
  switch (cycle) {
  var dy = (Math.cos(phase) - 1 - 0.1) * this.stroke/2;  //0.1 fudge for appearance
  cyl.piston.setPosition(cyl.piston.topPos.addXY(0, -dy));
  var cycle = Math.floor(phase / pi);  // Change color based on cycle
  var frac = phase / pi - cycle;  // Change shading based on fractional //
```

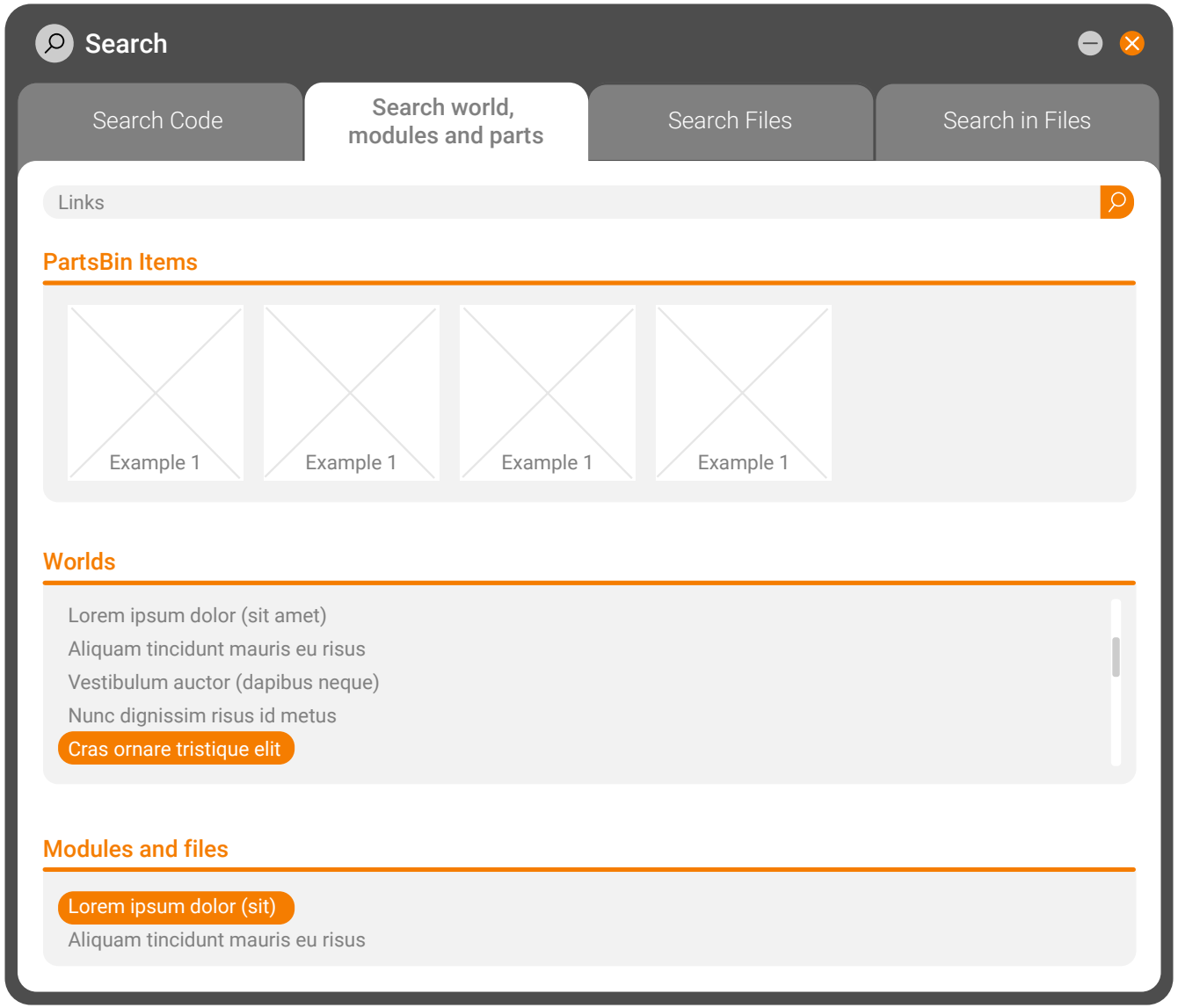
```
this.addScript(function movePiston(cyl) { // Method to move piston and connecting
  var pi = Math.PI;
  var phase = (this.crankAngle - cyl.angle);
  if (phase < 0) phase += pi*4;
  var dy = (Math.cos(phase) - 1 - 0.1) * this.stroke/2;  //0.1 fudge for appearance
  cyl.piston.setPosition(cyl.piston.topPos.addXY(0, -dy));
  var cycle = Math.floor(phase / pi);  // Change color based on cycle
  var frac = phase / pi - cycle;  // Change shading based on fractional part of cycle
  switch (cycle) {
```

Scripts, Connections and Properties


Partsbin
Advance Tools
Object Editor
Style Editor
Search
Help
Settings


Style Editor
Text_7
View
CSS
Layout
Text
Custom
Fill
Opacity
1
Color
Border
Width
1
Color
Radius
0
Style
Solid
Misc
Clip Mode
Visible
Tooltip
Hand style
Auto


Partsbin
Advance Tools
Object Editor
Style Editor
Search
Help
Settings


Search
Search Code
Search world, modules and parts
Search Files
Search in Files
Links
PartsBin Items
Example 1
Example 1
Example 1
Example 1
Worlds
Lorem ipsum dolor (sit amet)
Aliquam tincidunt mauris eu risus
Vestibulum auctor (dapibus neque)
Nunc dignissim risus id metus
Cras ornare tristique elit
Modules and files
Lorem ipsum dolor (sit)
Aliquam tincidunt mauris eu risus

Help

Cheat Sheet | Tutorials | Documentation | Play Quick Starter

All the shortcuts listed here for MacOS can be done using CTRL on Windows.

General

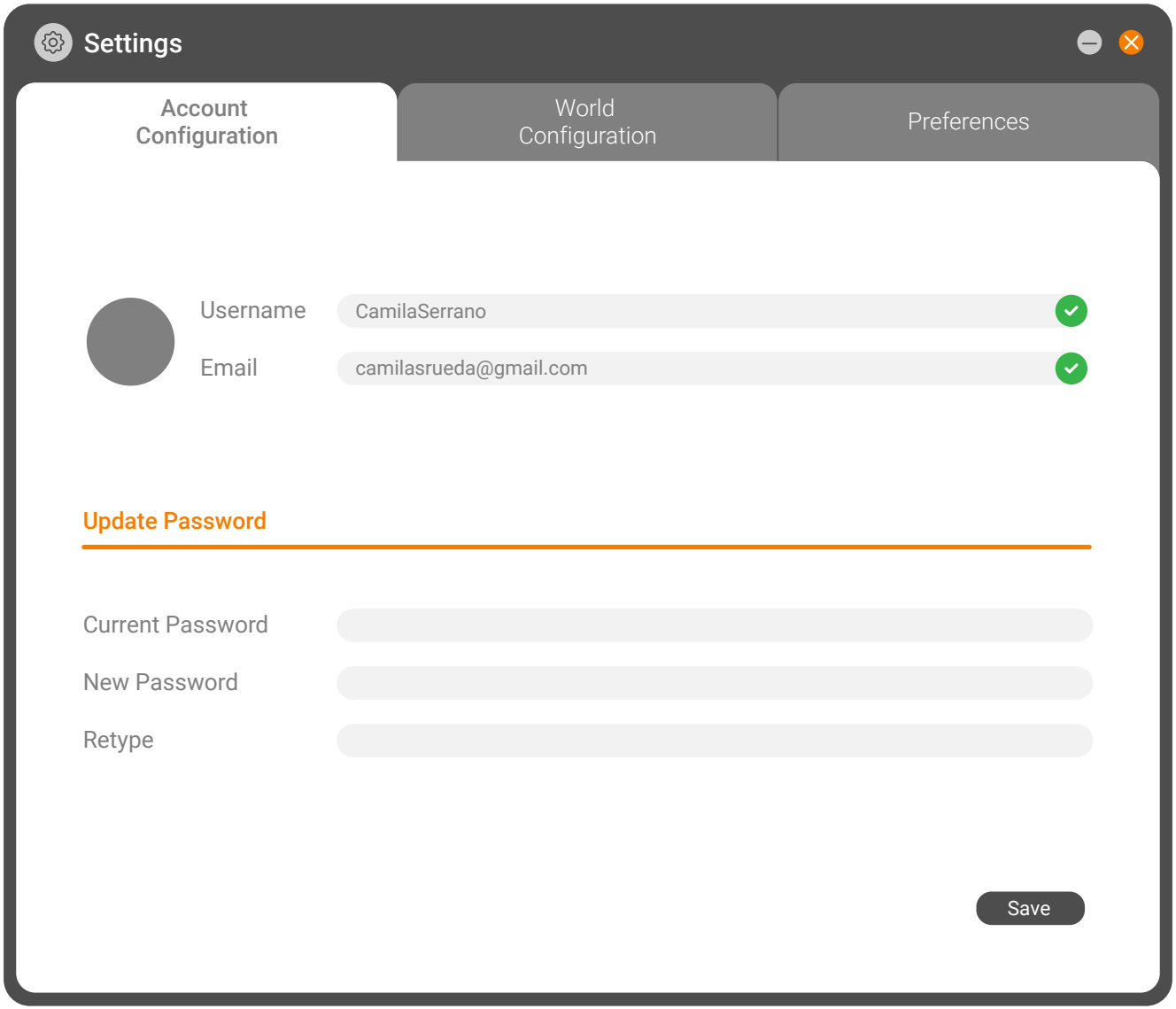
Reload Browser	CMD + R	Select All (Text Only)	CMD + A
Close Window	CMD + W	Copy (Text Only)	CMD + C
Save	CMD + S	Paste (Text Only)	CMD + V
Open Workspace	CMD + K	Cut (Text Only)	CMD + X
Open Object Editor	CMD + O	Open Partsbin	CMD + O
Open System Browser	CMD + B	Undo	CMD + Z

Code Execution

Do It (executes the statement)	CMD + D
Exchange (exchanges the last two selections)	CMD + E
Inspect (inspects the statement in a window with attributes)	CMD + SHIFT + I
Print It (prints the result of the statement)	CMD + P
Protocol (opens a menu with all the methods)	CMD + SHIFT + P
Find (finds the selected text; asks for a search string)	CMD + F
Find Again/Next	CMD + G

Morph Manipulation

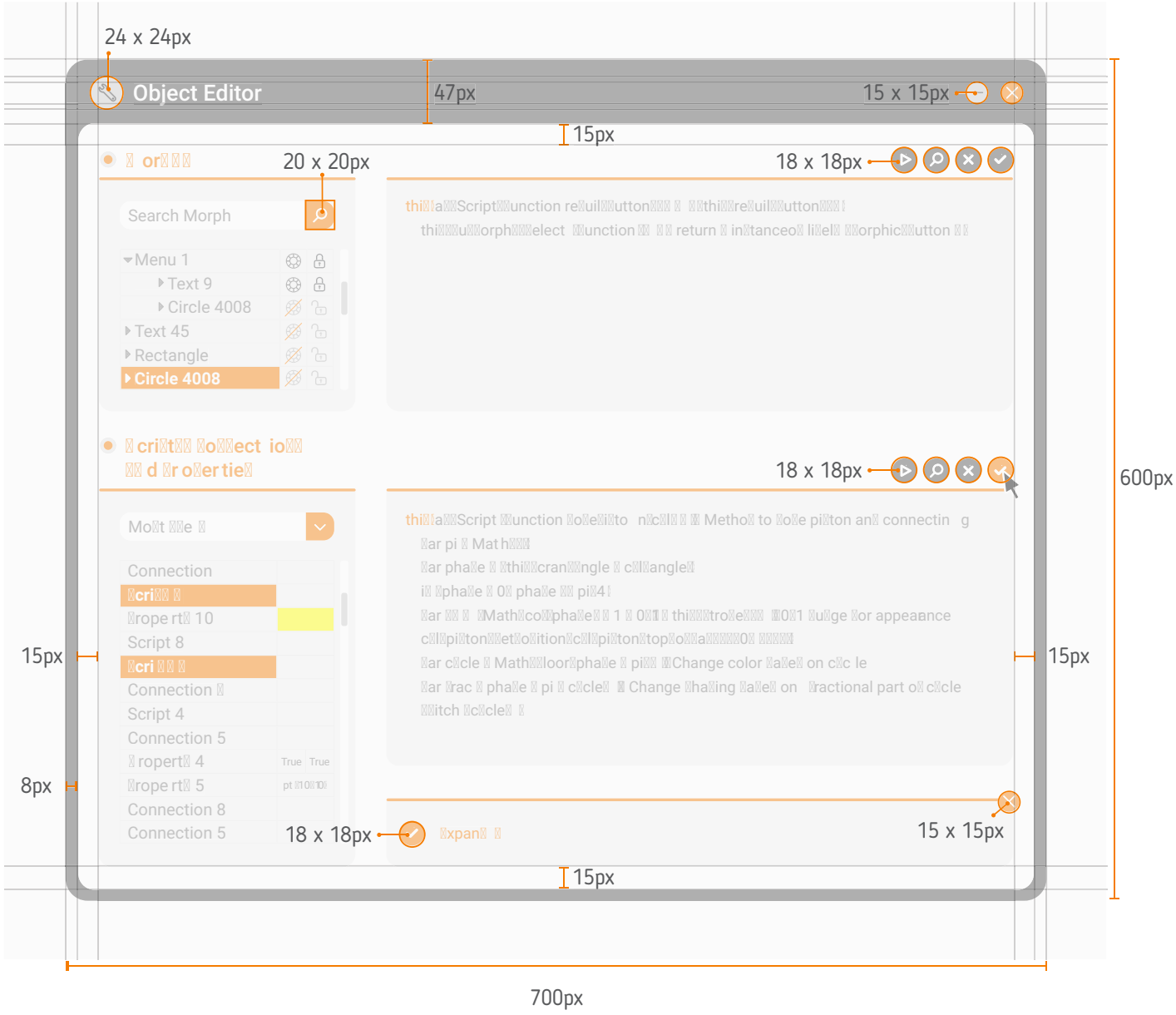

Partsbin
Advance Tools
Object Editor
Style Editor
Search
Help
Settings


Settings
Account Configuration
World Configuration
Preferences
Username
CamilaSerrano
Email
camilasrueda@gmail.com
Update Password
Current Password
New Password
Retype
Save

3. Secondary Windows

Secondary Windows have a secondary hierarchy that is shown using a lighter gray colour. This lighter colour allows the user to easily distinguish the words and make the content easy to read.

- Secondary Windows Specifications

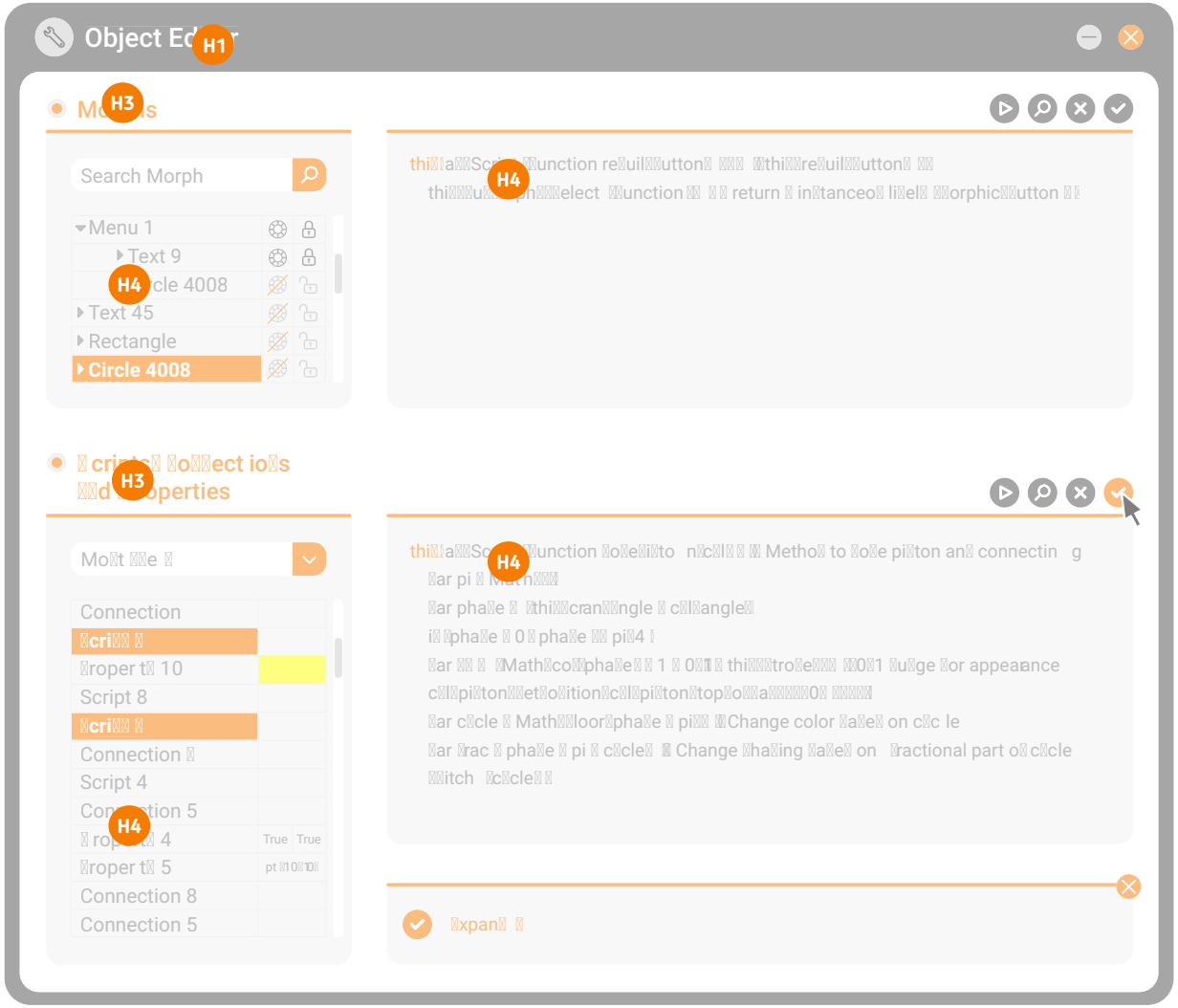


H1 **HEADING 1**
font-family : Roboto; Medium
font-size : 16px;
color : #FFFFFF;
color : rgb(255, 255, 255);

H3 **HEADING 3 – Highlighted**
font-family : Roboto; Medium
font-size : 14px;
color : #808080;
color : rgb(128, 128, 128);

HEADING 3 – Not Highlighted
font-family : Roboto; Regular
font-size : 14px;
color : #808080;
color : rgb(128, 128, 128);

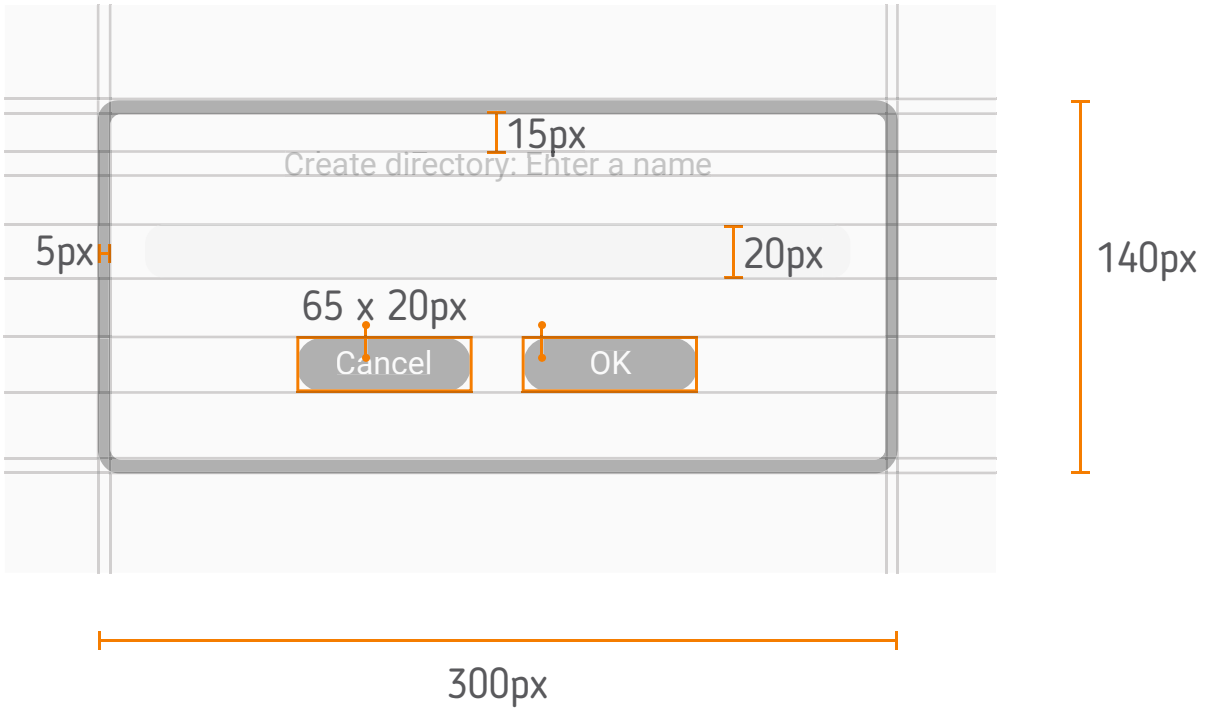
H4 **HEADING 4**
font-family : Roboto; Regular
font-size : 12px;
color : #808080;
color : rgb(128, 128, 128);



4. Dialog Windows

The main purpose of the Dialog Windows is to allow the user to quickly make decisions, hence the simple design of the windows. To make it simple, we are not displaying window's titles or minimizing and closing icons; the user should be able to select the correct option by clicking the button or closing the window by clicking the cancel button. When the user hovers over any button, the colour of the button will change to bright orange, making it more "lively", colourful and visually recognizable.

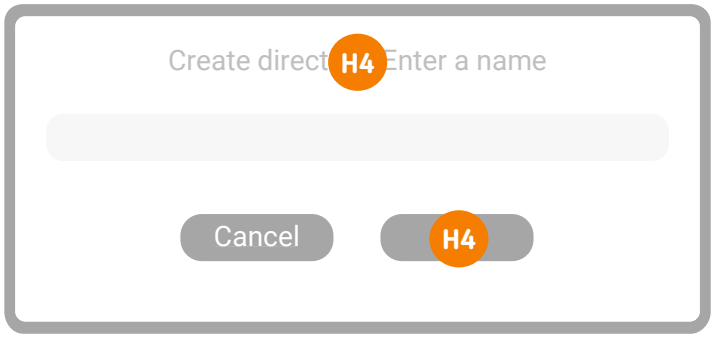
- Dialog Windows Specifications



H3 **HEADING 3 - Highlighted**
font-family : Roboto; Medium
font-size : 14px;
color : #808080;
color : rgb(128, 128, 128);

HEADING 3 - Not Highlighted
font-family : Roboto; Regular
font-size : 14px;
color : #808080;
color : rgb(128, 128, 128);

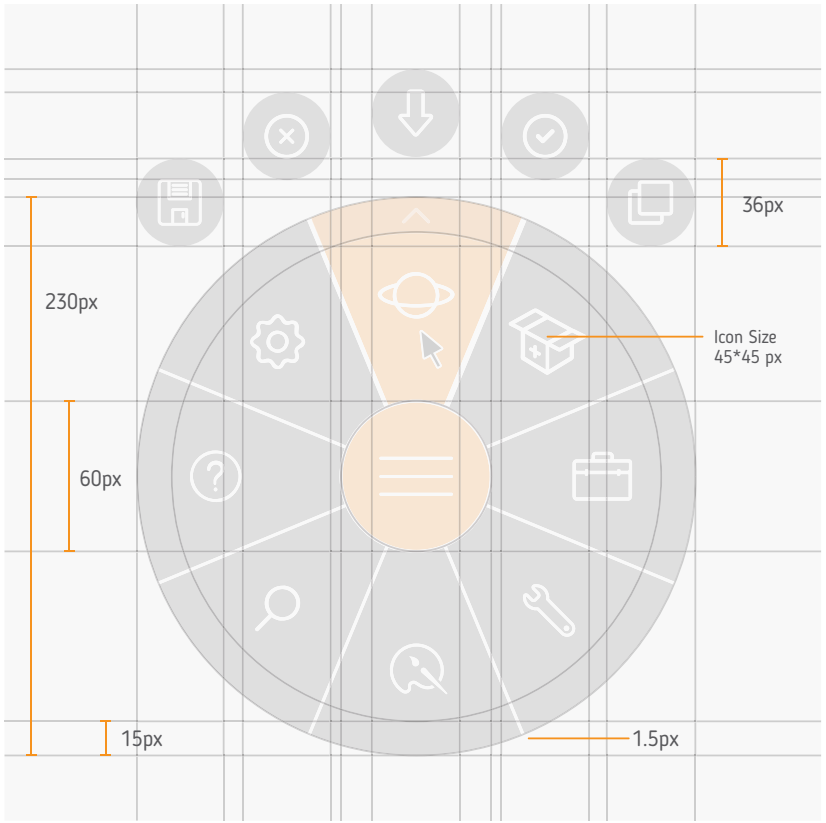
H4 **HEADING 4**
font-family : Roboto; Regular
font-size : 12px;
color : #808080;
color : rgb(128, 128, 128);



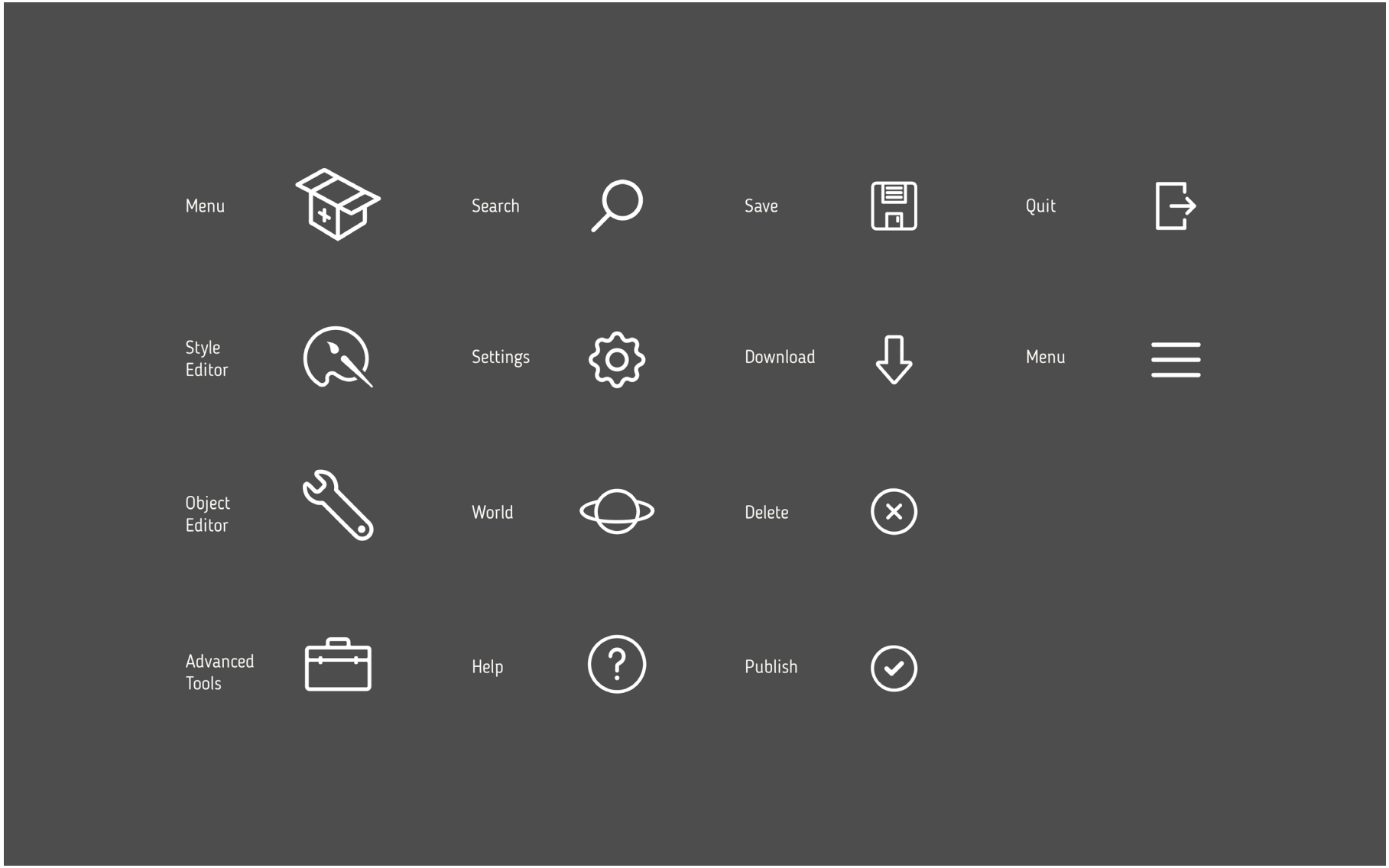
5. Pie Menu

The main purpose of the Pie Menu is to help the user recognize and distinguish the eight main tools in Lively.
To make it more intuitive and easy to understand, we followed a design pattern already established for the users, the four cardinal and four intercardinal directions. As a result, we can show the eight tools in a balanced and effective way.

- Pie Menu Specifications

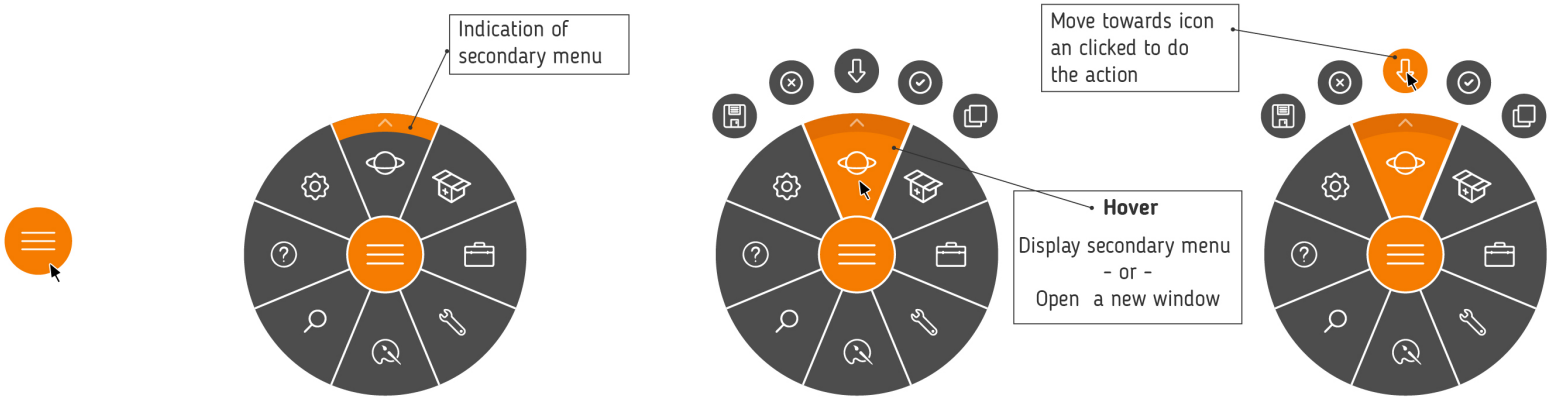


- Pie Menu Icons



Pie menu

Pie interactions

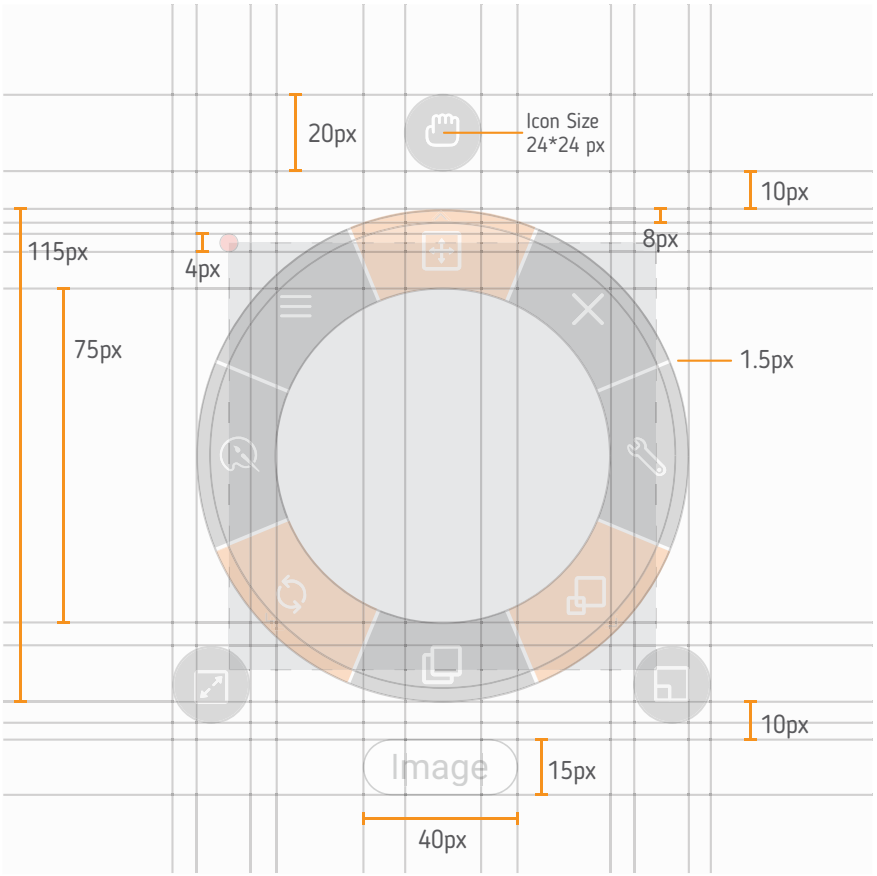


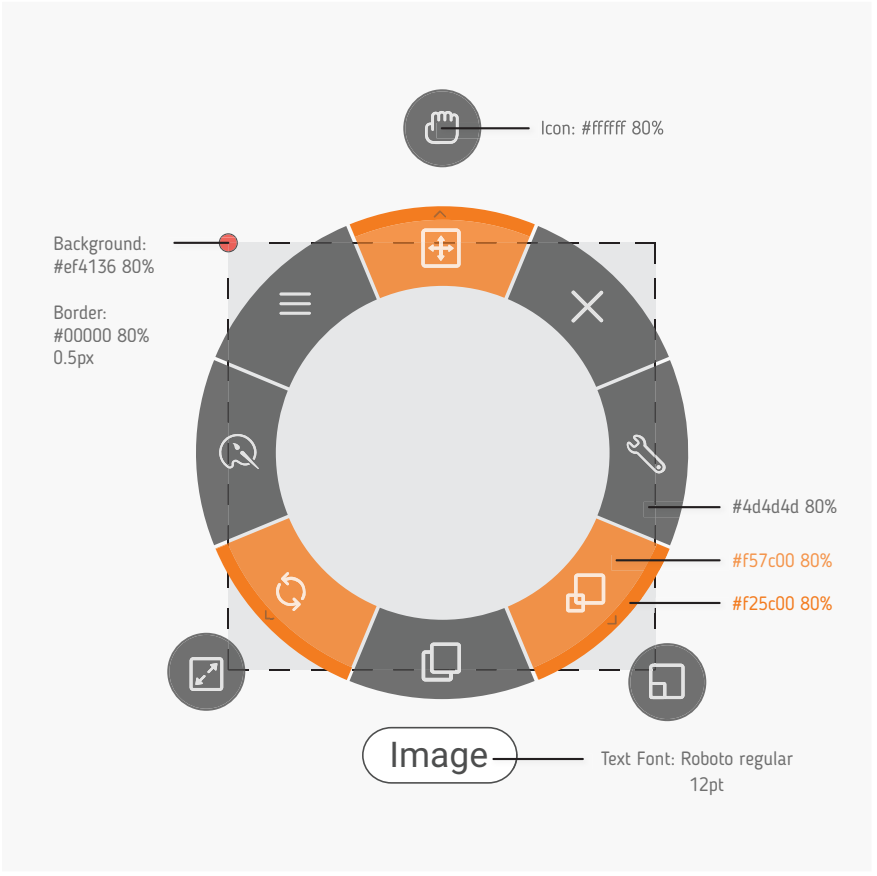
Tool tips

6. Halo

The Lively Halo was completely redesigned.

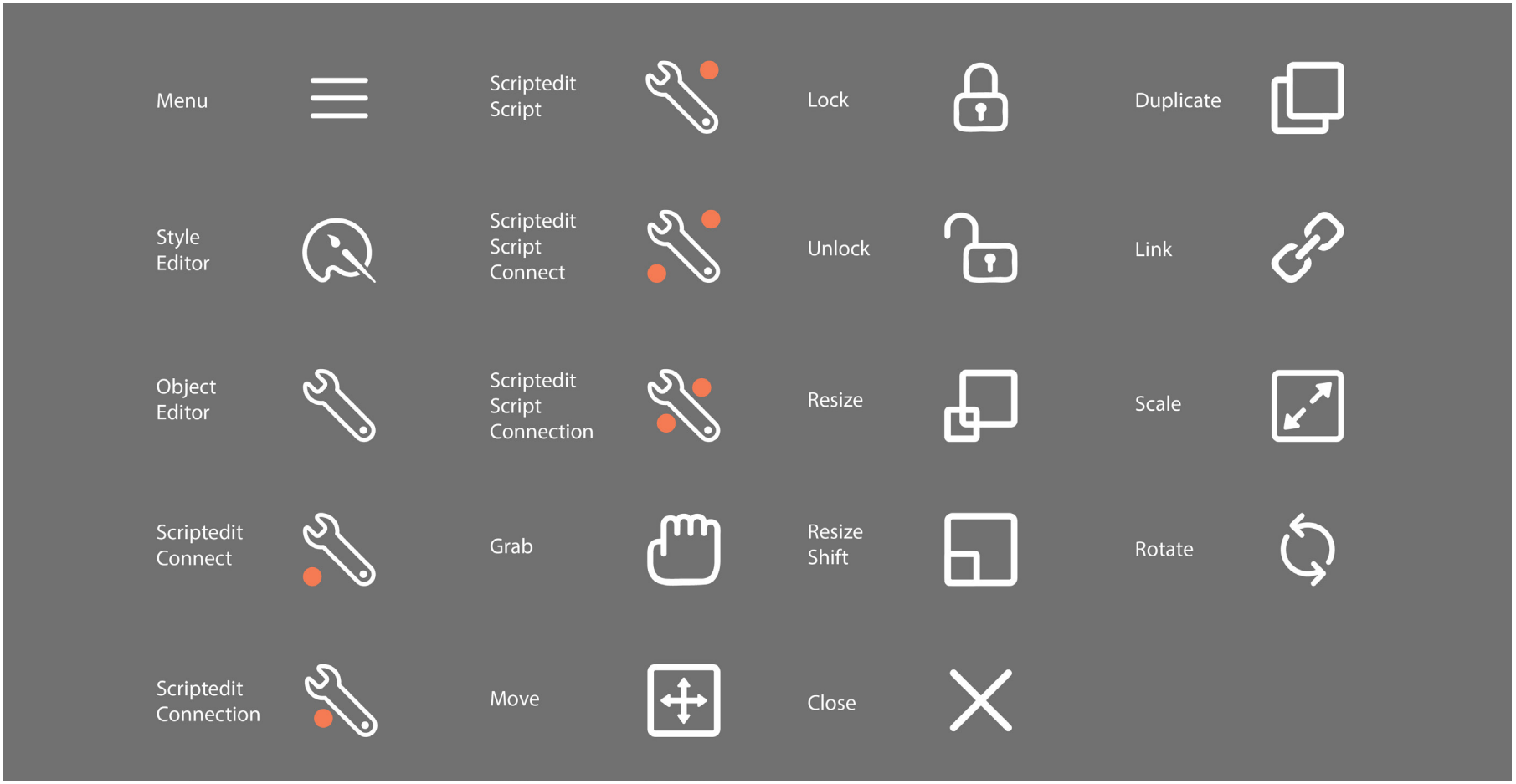
- Halo Specifications





- Halo Icons

There are 11 icons in the Halo: North (Group and Grab), Northeast (Close), East (Object Editor), Southeast (Resize and Proportionally Resize), South (Duplicate), Southwest (Rotate and Rescale), West (Style Editor), Northwest (Menu). What's more, there is a name tag right beneath the Halo. The design of all the icons in the Halo is consistent the other icons in Lively Web, which is documented in UI Design Principle section.



Colour

In order to create contrast in the interface and make it more appealing for the users, we are contrasting the main bright orange colour from the logo with a dark gray colour. We used colours to distinguish hierarchy levels in Lively (dark vs. light grey, orange).

#f57c00	#f57c00	#cccccc	#f2f2f2
R:245 G:124 B:0	R:245 G:124 B:0	R:204 G:204 B:204	R:242 G:242 B:242

Font

Roboto Main Font

Roboto is a Google Web Font that features friendly and open curve forms that match with our rounded elements design. Additionally, it is a clean and easy to read sans-serif font that complements the font used in the logo.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

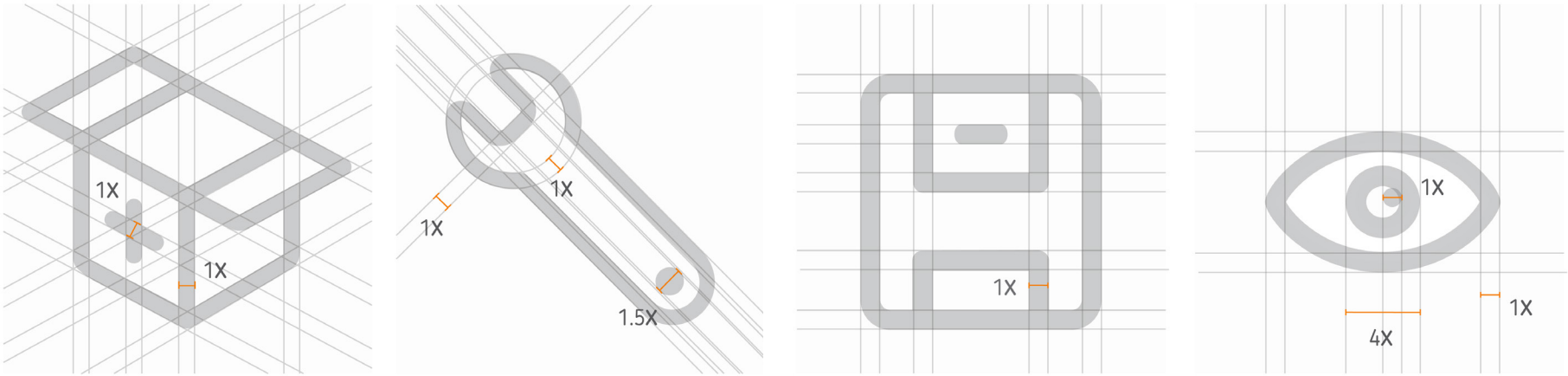
a b c d e f g h i j k l m n o p q r s t u v w x y z

1 2 3 4 5 6 7 8 9 0

Iconography

The icon design is following the same principles of branding by using the same weight for lines and round corners.

- Icons Specifications



Save 

publish 

preview 

account 

redo 

collaboration 

undo 

chat 

Folders

Parts

Lively Web

Server

User

World

Broken image

Account Page Icons

Game

Music

Graphic & Design

Developer Tools

Productivity

Active Sessions Gallery

Project Gallery

Partsbin Reorganization

The need for reorganizing the partsbin prompted the following charts. The first chart explains the workspace elements while the second one offers recommendation for reorganizing the partsbin.

Workspace

- About the World
- User Account
- Communication Frameworks — *Chat / MediaStreaming*
- Partsbin
  - Default
    - Atoms — *Basic Shapes (Line / Rectangle / Ellipse / Polygon / Star / SVG Morph)*, *Insert (Images / Videoes / Audioes)*, *Text*, *Pen*
    - Molecules — *Widgets /* *Tools : Notes / ToDolist / AudioVideoPlayer / Recorder / Layout / Electrical Components*
    - Cells — *Games /* *Visualization / Scrum (or in Molecules)*
    - Platform — *Presentation System (slides)*, *Web Development System*, *Collaboration System*
  - User's
    - Favorites
    - User created category
    - User created category
    - User created category

    Private
    Share to friend
    Share to public
  - Community
    - Friends' Partsbin
    - Recommended
    - Explore by tags
- Object Editor
- Style Editor
- Advanced Tools — Droppable Behaviors
- Settings
- Search
- Help & Tutorials
- History Management Framework

Partsbin Category

Current Category	Recommended Locations
Backgrounds	World settings
Basic	Partsbin / Default / Atoms / Basic Shapes
BYOIE	BYOIE
Clojure	Object Editor
CodeSnippets	CodeSnippets
Collaboration	Partsbin / Default / Platform / Collaboration System
Constraints	Constraints
Controls	Behavoirs & Control /
Debugging	Advanced Tools
Demo	Partsbin / User's
Demos	Partsbin / User's
Dialogs	Dialogs
DockedParts	Partsbin / Default / Atoms / Basic Shapes
Documentation	Help & Tutorials
DroppableBehaviors	Behavoirs & Control /
ElectricalComponents	Partsbin / Default / Molecules /
FRP	FRP
Fun	Partsbin / Default / Cells / Games
Games	Partsbin / Default / Cells / Games
GStreamer	GStreamer
HTML	Object Editor
Inputs	Behavoirs & Control /
iPadWidgets	Partsbin / Default / Molecules / iPadWidgets
Layout	Partsbin / Default / Molecules / Layout
MattH	Partsbin / User's

Partsbin Category

Current Category	Recommended Locations
MDM	Partsbin / User's
MediaStreaming	Communication Frameworks
ParableOfPolygons	Partsbin / Default / Atoms / Basic Shapes
Physics	Physics
Pictures	Partsbin / Default / Atoms / Insert
PollutionDemoControls	PollutionDemoControls
Presentation	Partsbin / Default / Platform / Presentation System
Presenting	Partsbin / Default / Platform / Presentation System
Productivity	Partsbin / Default / Molecules /
Sandbox	Partsbin / User's
Scripting	Scripting
Sketching	Sketching
SketchyInputs	SketchyInputs
Slides	Partsbin / Default / Platform / Presentation System
Splittermond	Splittermond
Stacks	Stacks
Text	Partsbin / Default / Atoms / text
Toolbox	Toolbox
Tools	Tools
~~Uncategorized~~	~~Uncategorized~~
UVicLab2	Partsbin / User's
Visualization	Partsbin / Default / Cells / Visualization
Web	Partsbin / Default / Platform / Web Development System
WelcomePage	Partsbin / Default / Platform / Web Development System
Widgets	Partsbin / Default / Molecules / Widgets
Wiki	Help & Tutorials

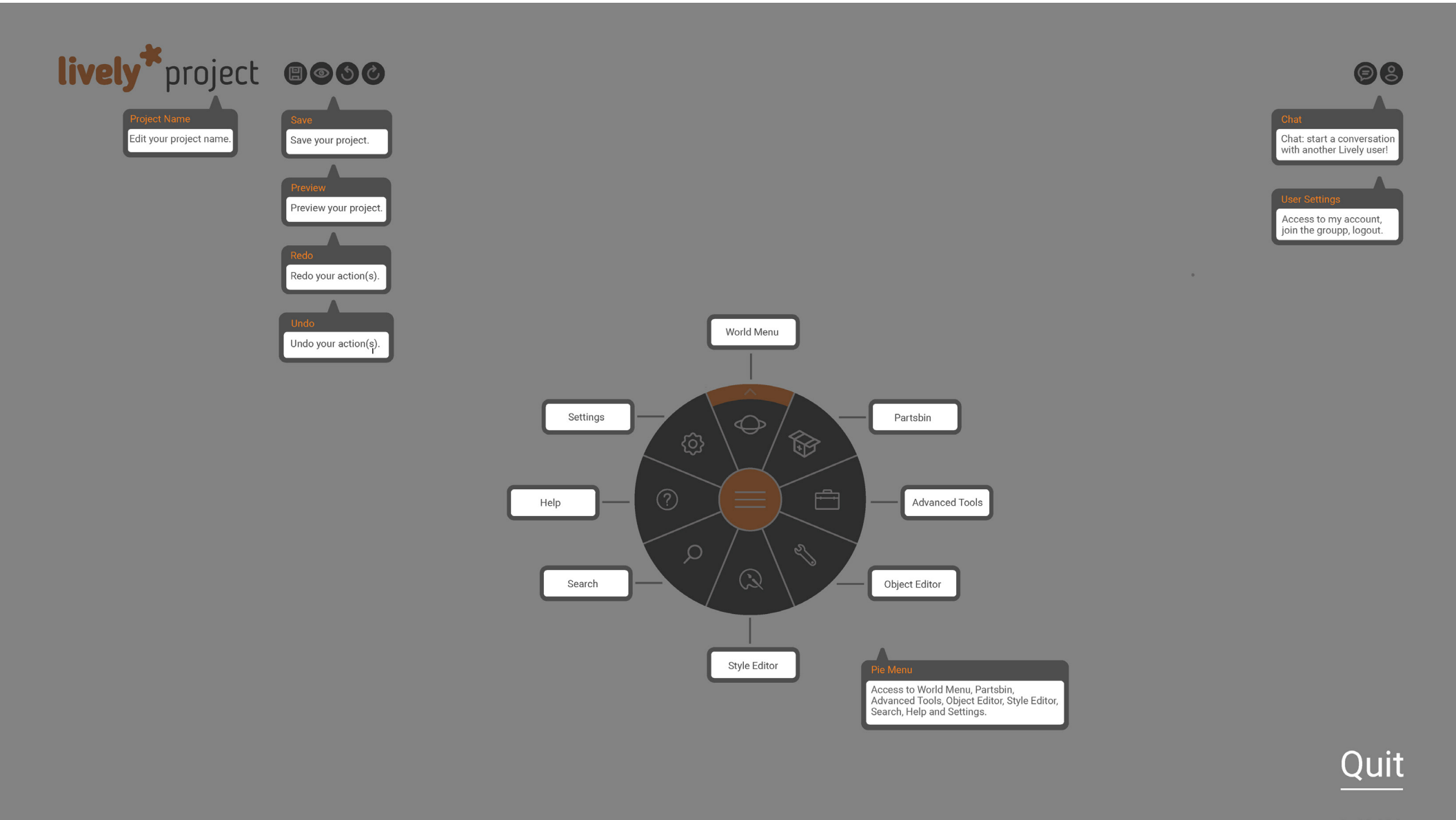
Notes:

1. The items with gray background are the things I am not sure how to organize.

2. Different colors correspond to diffrent categories in Chart 1.

Quick Start

Quick start is part of the “help system” in Lively Web. It helps users especially, those using the system for the first time to understand the basics of the workspace interface. With a button triggering the quick start, it appears on top of the workspace with several windows explaining the functionality of the interface.



Tutorials

We developed two step-by-step tutorials for Lively Web. One based on design and one based on a game with step by step instructions.

Tutorial 1 - Amusement Park

Objectives:

Open Partsbin to find all the assets needed in the project(s).
Use the Halo to change the properties of the objects, such as size and position
Use a droppable behaviour to make the experience more dynamic.

Steps:

1. Open Partsbin by right click on the workspace or command+p, and find all the assets in the MDM category. Drag and Drop them to the workspace when needed.

2. Command+click on each asset to get the halo, use the shift+left-bottom icon to rescale. Move the assets for the layout that you like.

3. Find the droppable behaviour and the wheel in Partsbin, set a rotation and drop it to the wheel, move the wheel to the ferris wheel support.

Tutorial 2 - Flappy Bird

Objectives:

Different ways to access the PartsBin
How to access the Object Editor
How to interact with the Object Editor

Steps:

1. Open the PartsBin (also accessible by right-clicking anywhere in the world or by clicking on the PartsBin icon on the wheel menu
2. Search for 'bird' drag it out from the PartsBin to the game scene, then press the [reset] button
3. Drag the [flap] icon from the list beside the launch pad onto the highlighted area in the launch pad, select 'flappy' in the drop down list, then click 'Go!'
4. Repeat for the [addPipe] icon
5. Press 't' and give it a try :) => it flies through all the pipes! :(
6. Cmd-click on the game scene TWICE to get the halo the the entire game
7. Click on the object editor icon in the halo to open the object editor
8. Select 'flap' in the selection panel
9. In the object editor, right-click where the codes are, then setting >> show line numbers.
10. Below line #6, paste the following text: else if(this.hitPipes()) this.reset()
11. Press [reset] in launch pad and have fun :)

Recommendations

Direct Manipulation

While the Lively Web provides one-of-a-kind feature in allowing users to interact directly with any given morph object, further refinement can be made During the process of this project, we proposed the crucial undo/redo feature.
Here is a list of changes and implementations pertaining to redo/undo:

1. Undo/Redo shortcuts: Implemented the ability to use Ctrl+Z, (CMD+Z for Mac) and Ctrl+Y (CMD+Y for Mac) to undo/redo morphic actions in lively.

2. Fixed a bug related to undoing a morph movement (The morph would not moved back to its exact previous location)

3. Fixed a bug related to undoing grabbing a parent morph with a submorph inside it (The morph would not moved back to its exact previous location due to boundary issues).

4. Fixed a bug related to undoing editor actions

5. Implementing undo/redo functionality for SVG transforms

The lack of direct visual feedback often leaves a novice user confused about the unexpected results. For example, when working with a composite morph, it is sometimes difficult for the user to realize the real target of a dropping action. With little to no visual feedback, it is very easy to place a morph onto the wrong target and bring unwanted interference to any future interactions with that morph. We have proposed a modified version of the dropping behaviour. In our proposal, the shadow appeared when grabbing a morph is replaced by transparency of the grabbed morph, and a green outline of the target morph would indicate the successfully dropping onto that morph. Visual cues should also be presented to indicate the application of a droppable behaviour. Preferably, the user should be able to undo the application of a droppable behaviour by directly interacting with the visual indication.

Even though it is the Lively Web's intent to provide equal dynamicity and malleability to both programmer and interactive user (Palacz); currently, it has only minimal support for direct SVG manipulation. In order to achieve better direct manipulation experience, it is recommended that the Lively Web should provide more sophisticated tools for direct SVG manipulation. Furthermore, because of the incident happened during our development process, it is also recommended that auto scaling should be provided to user-uploaded images.

Mouse Event Handling

Currently, the handling of any mouse event is initiated from the root of any morph tree. However, more often than not, it would be the user's intention for a sub-morph should be able to handle a mouse event differently than its owner. Therefore, the search for a mouse event handler should be initiated at the leaf-level of the tree instead of the root.

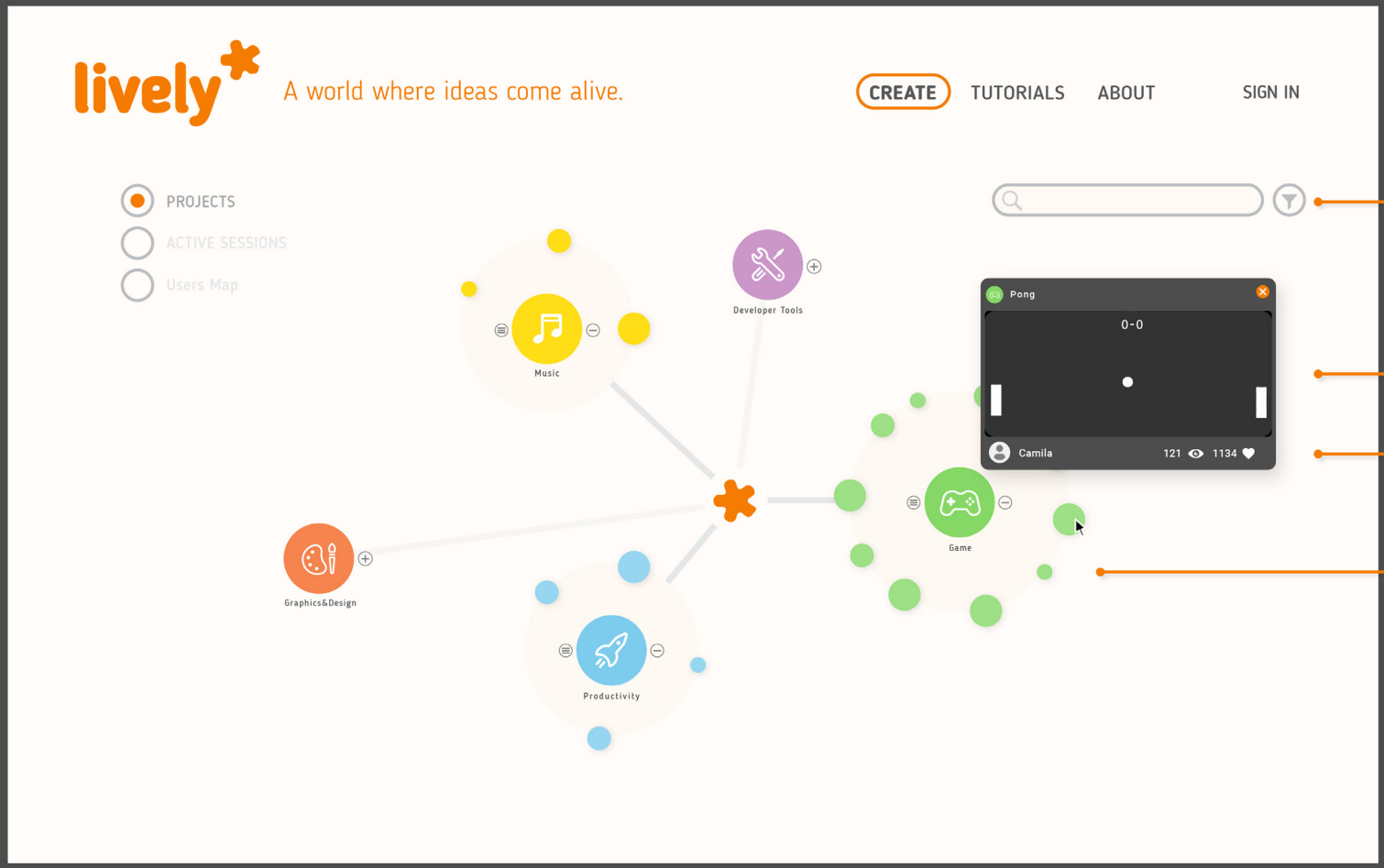
Live Editing

This happened many times during the first week's demo where a novice user is often surprised by the unintended grabbing of a submorph from a composite morph. Even though "mode-less" is desired in such live editing environment, (Maloney and Smith), to "distinguish editing gesture from operating gesture" is equally important. As a proposed recommendation, dragging on any part of a composite morph should, by default, move the entire morph. The intention of editing any particular submorph should be indicated only with the presence of the halo around that submorph.

Technical Documentation

It was also mentioned, upon initial interaction with the lively web, first-time user would often feel lost at the landing page. The lack of technical documentation and well-designed training program has created noticeable barrier of entry to the lively system. However, because of its 'liveliness' nature, traditional API document would become outdated upon any modification to the system. A 'lively' form of technical documentation is crucial for any first-time user to quickly adapt to the system. In the meantime, a carefully-designed training program would also help in easing the transition to the lively web.

Future First Page Iterations

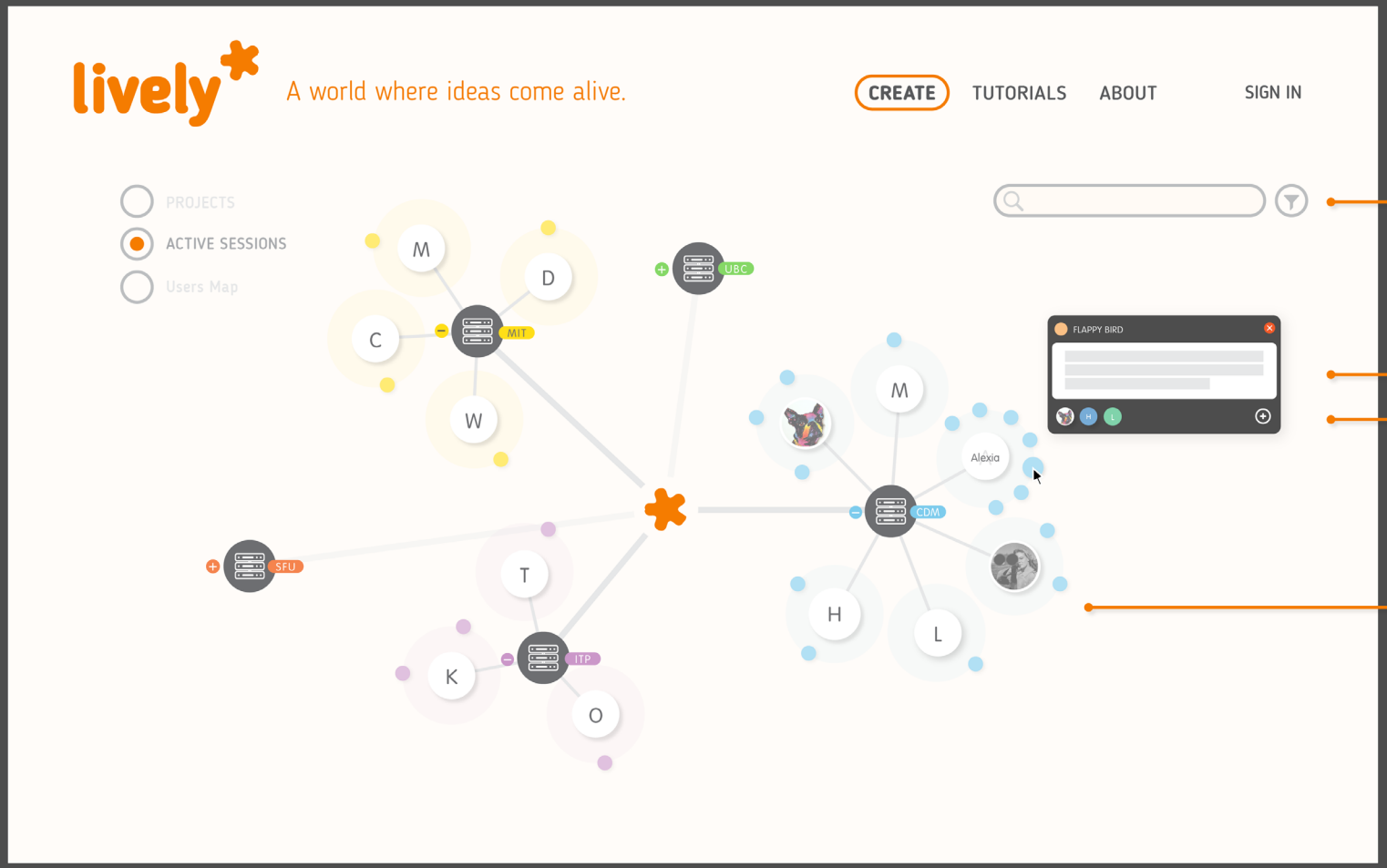


Search & Filter
The filter could be time, popularity, different categories etc.

Project Preview

Number of veiws
Number of likes

Different Sizes
Based on the popularity, projects could be presented in different sizes.



Search & Filter

The filter could be time, popularity, different categories etc.

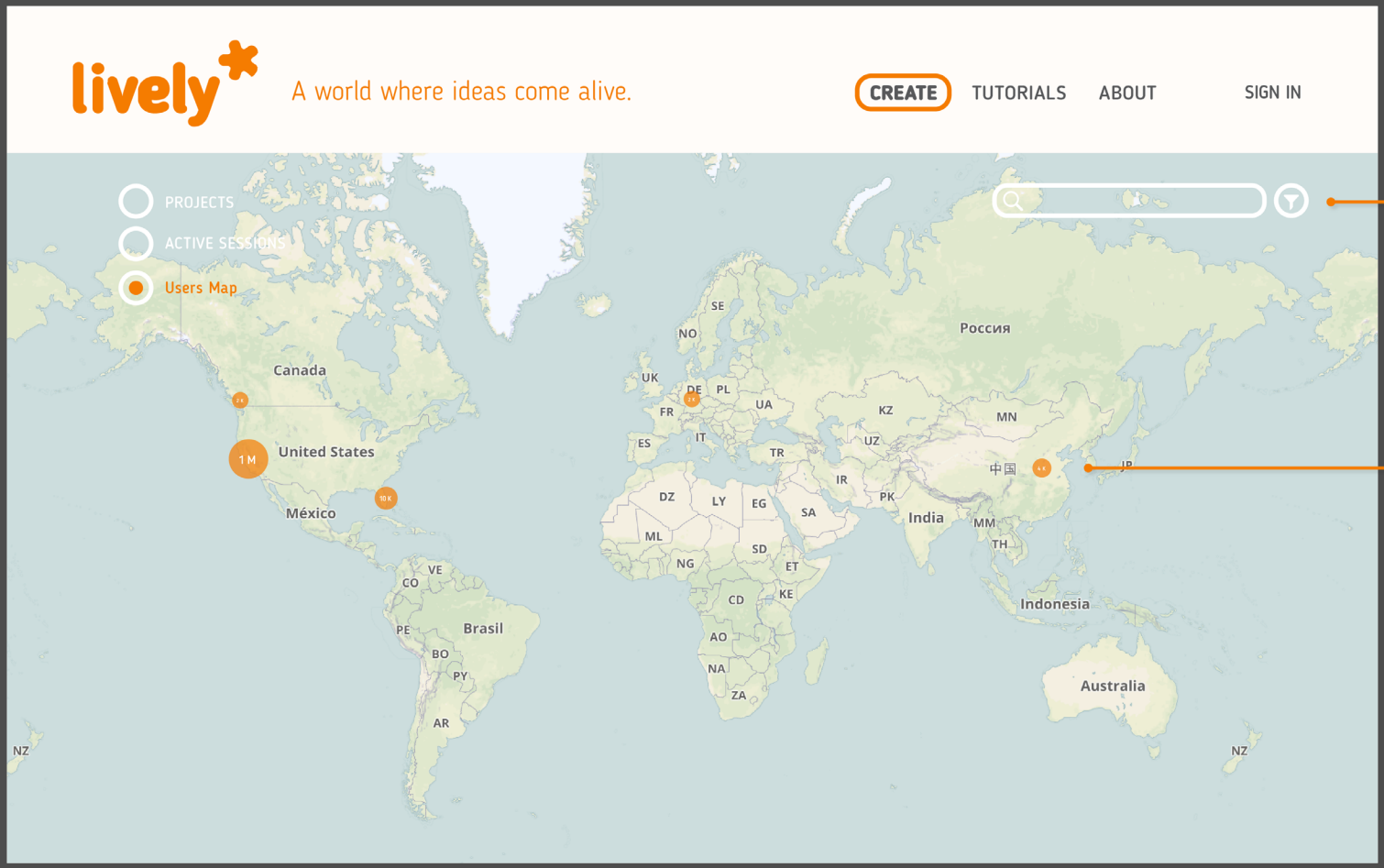
Preview

Collaboration

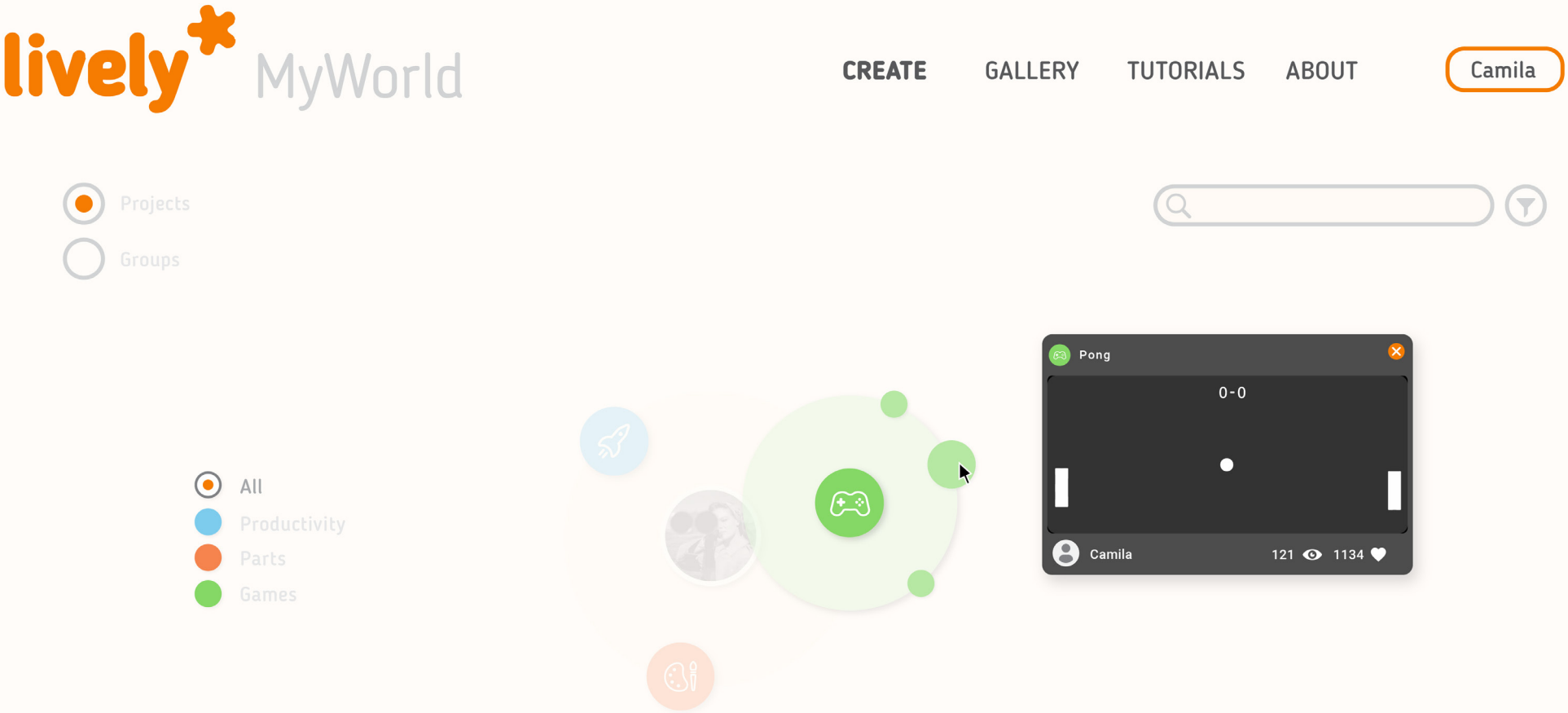
A world is a place under development and many people could collaborate to work on one world at the same time.

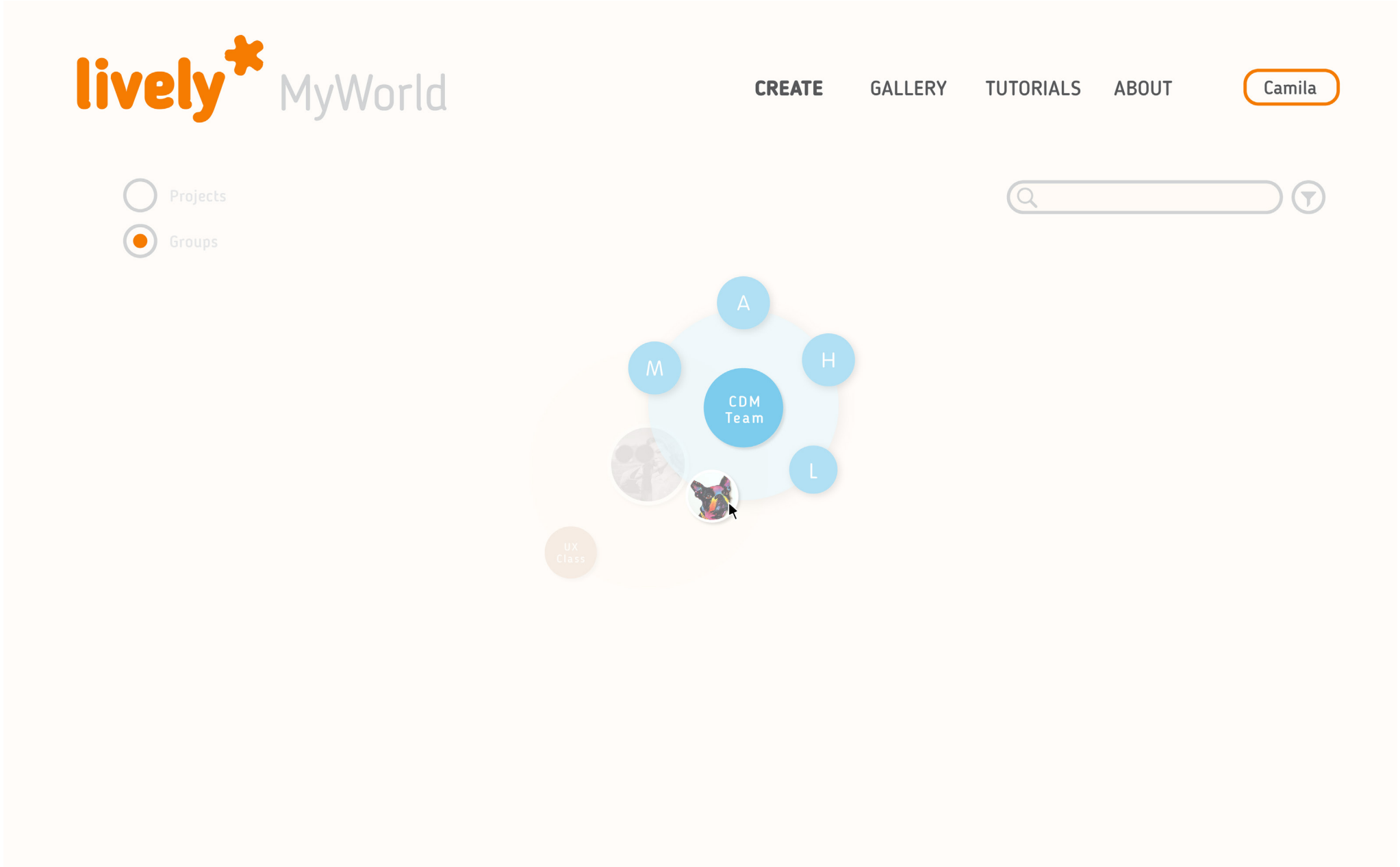
Custmized Avatar

Users could have differnet default avatar based on their name and they can also custmize it.


lively
A world where ideas come alive.
CREATE
TUTORIALS
ABOUT
SIGN IN
PROJECTS
ACTIVE SESSIONS
Users Map
Canada
United States
1 M
México
Россия
SE
NO
UK
DE
PL
UA
FR
ES
IT
TR
KZ
UZ
MN
中国
IR
PK
India
DZ
LY
EG
SA
ML
SD
NG
ET
CD
KE
AO
NA
ZA
MM
TH
Indonesia
Australia
NZ
VE
CO
PE
Brasil
BO
PY
AR
Search & Filter
The filter could be time, popularity, different categories etc.
Show User Numbers

Account Page Iterations




lively MyWorld
CREATE
GALLERY
TUTORIALS
ABOUT
Camila
Projects
Groups
A
H
M
CDM Team
L
UX Class

Lively Bug Documentation

The purpose of this document was to share the issues, suggestions and potential features with the Lively Web team. The CDM team reported over 30 issues since the start of the project. We strongly recommend to the Lively Team to use the updatable document to allow current and future users to report system bugs and difficulties on Lively.

The current Lively Bug Document on Google Drive: https://goo.gl/p9GGV2
(A similar document could be created on Lively.)

Bibliography

Maloney, John H. and Randall B. Smith. “Directness and Liveness in the Morphic User Interface Construction Environment.” User Interface Software and Technology Symposium. 1995. 21-28.

Palacz, Krzysztof. “The Lively Kernel Application Framework.” 2008. http://www.svgopen.org/. 09 08 2015 <http://www.svgopen.org/2008/papers/93-The_Lively_Kernel_Web_Applicattion_Framework/>.