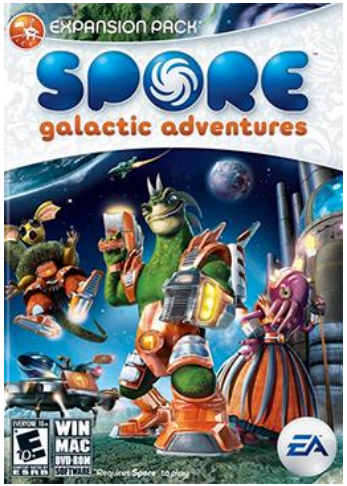
How to Implement AAA Game UI in HTML and JavaScript

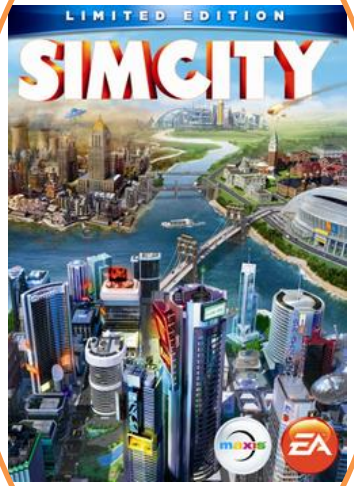
Yee Cheng Chin
Senior Software Engineer, Electronic Arts / Maxis

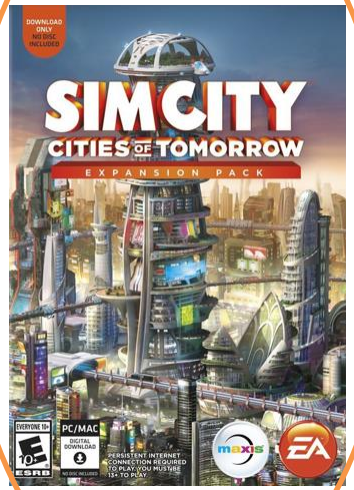
GAME DEVELOPERS CONFERENCE®
MOSCONE CENTER · SAN FRANCISCO, CA
MARCH 2-6, 2015 · EXPO: MARCH 4-6, 2015

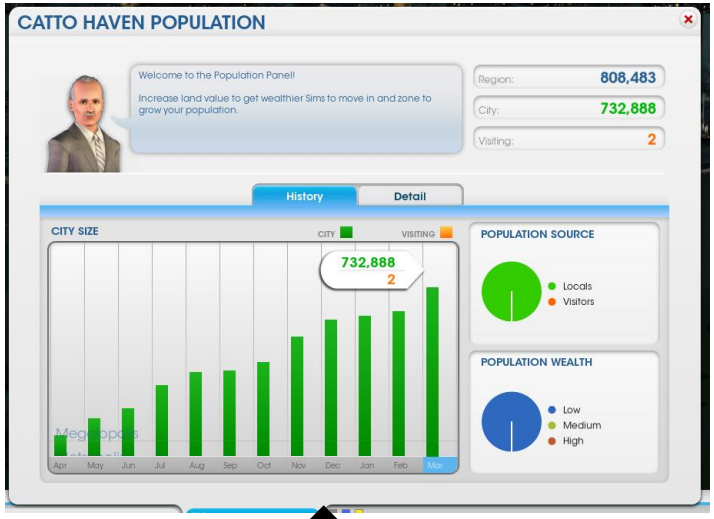
Who am I?

- 7 years at Maxis








CATTO HAVEN POPULATION
Welcome to the Population Panel!
Increase land value to get wealthier Sims to move in and zone to grow your population.
Region: 808,483
City: 732,888
Visiting: 2
History
Detail
CITY SIZE
CITY
VISITING
732,888
2
POPULATION SOURCE
Locals
Visitors
POPULATION WEALTH
Low
Medium
High
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar


PLAY
RESUME GAME
JOIN GAME
CREATE GAME
Viridian Woods
Name Region
Multiplay
Public Region
Private Region
Sandbox Mode
Sandbox Mode
BACK
NEXT


Buses
Streetcars
Trains
Boats
Planes

What this talk is about

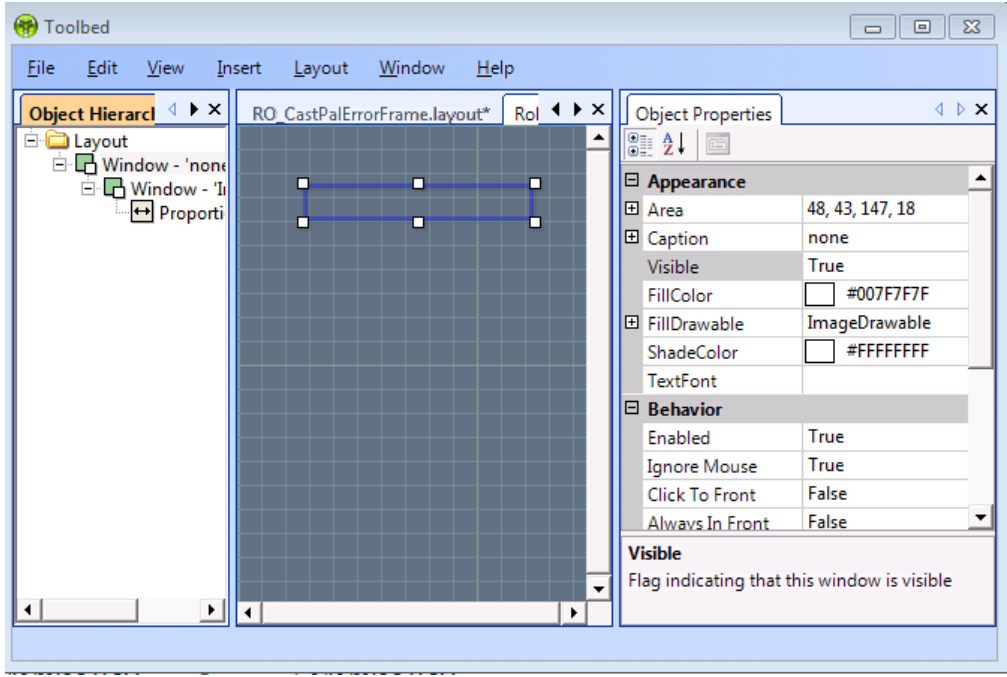
1. Why HTML*?
2. Maxis' UI, based on HTML. Shipped with SimCity
3. Tips and gotchas

What this talk is not about

- Building web games in HTML5. This is specific to using HTML to build just the UI component for a large native game.
- Emscripten / asm.js
- How to build generic web pages.

Maxis' journey in UI tech

- UTFWin
  - Custom built solution, didn't fit all our needs, hard to animate, hook up.



Maxis' journey in UI tech

- Scaleform / Flash
  - Relies on external license, reliant on Flash.
  - Non-mergeable binary data format
  - Potential issues with ActionScript performance
  - Flash as a general technology was slowly losing support from major players like Apple

SimCity and UI

- Start of SimCity development, needed a new UI system
- Vision of combined web and client interfaces, with shared components between web and in-game UI
- Easy to update and integrate web content

SimCity and UI

- Web-based UI using EA WebKit, and custom-built JavaScript layer.
- Investigated other options as well
  - For engine, at the time didn't find anything better, and EA WebKit was starting to get traction.
  - For building web pages, couldn't find good WYSIWYG editor, built it ourselves.


SIMCITY™
SINGLE-PLAYER
MULTIPLAYER
RESUME
SuperCircularCollider
PLAY
SIMCITY WORLD
Transfer your Multiplayer Regions and Cities to play in Single-Player!
AMUSEMENT PARK SET
LEARN MORE
CITYLOG

What this talk is about

1. **Why HTML*?**
2. Maxis' UI, based on HTML. Shipped with SimCity
3. Tips and gotchas

Why HTML?

- Existing tech
  - WebKit, Blink, Gecko, etc.
  - Inspector/Debugger

Why HTML?

- Fast reloading. Takes 3 seconds to reload the entire UI. No import/export process, or compilation/linking required.
- Most implementations have blazing fast JavaScript engines with JIT compilations.
- Easy to update over the web

Why HTML?

- You are probably going to need web content for leaderboards, etc. anyway. May as well go all the way!
- Large community. Easy to hire people or find knowledge
  - Caveat: Not everyone who has “web” background is suitable for game dev. Need to be performance-conscious.

Why HTML?

- Modding. If you want your PC game to be modded by your community then there's really nothing that beats HTML.
  - Most people know it, and it's easy to modify.
- Caveat: Remember people can and will read your code:
  - `gameCode.DoSomethingStupid= function(stupidity) {… /* ☹ */}`

Other EA games using HTML tech

- SimCity
- Skate 3
- Sims
- Most new console games for online features

What this talk is about

1. Why HTML*?
2. **Maxis' UI, based on HTML. Shipped with SimCity**
3. Tips and gotchas

SimCity's UI

- More complicated than action oriented games UI.
  - As a result both have more requirements (layouts, dynamic UI scaling, etc) and higher budget.
- EA WebKit
- MUiLE

EA WebKit

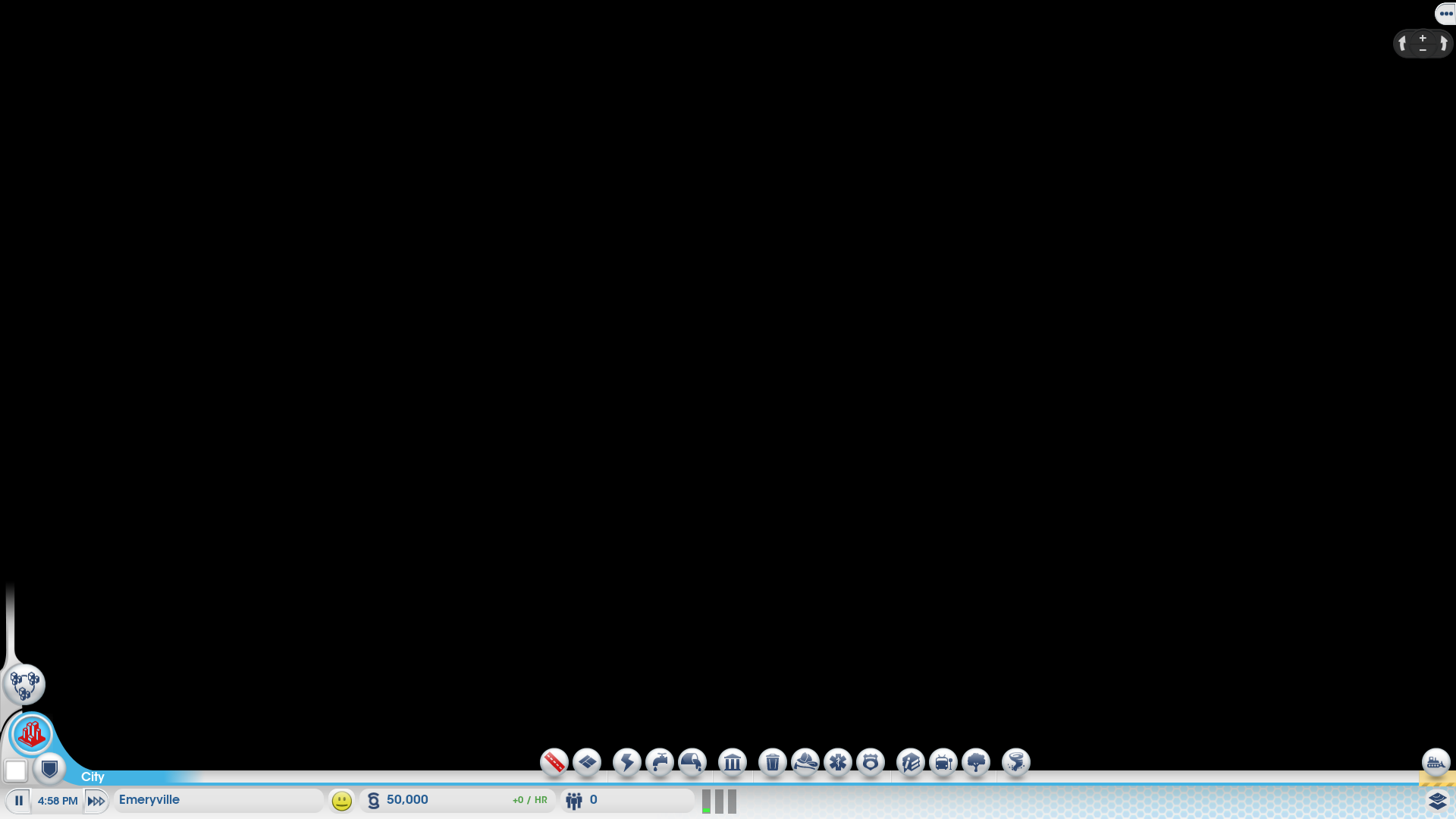
- Backend of our UI
- Fork of Apple's WebKit project, but designed to be embedded into games, while providing much more hooks such as custom memory allocator, profiler, JS/C++ bindings, network layer, etc.
- Open sourced: http://gpl.ea.com

EA WebKit

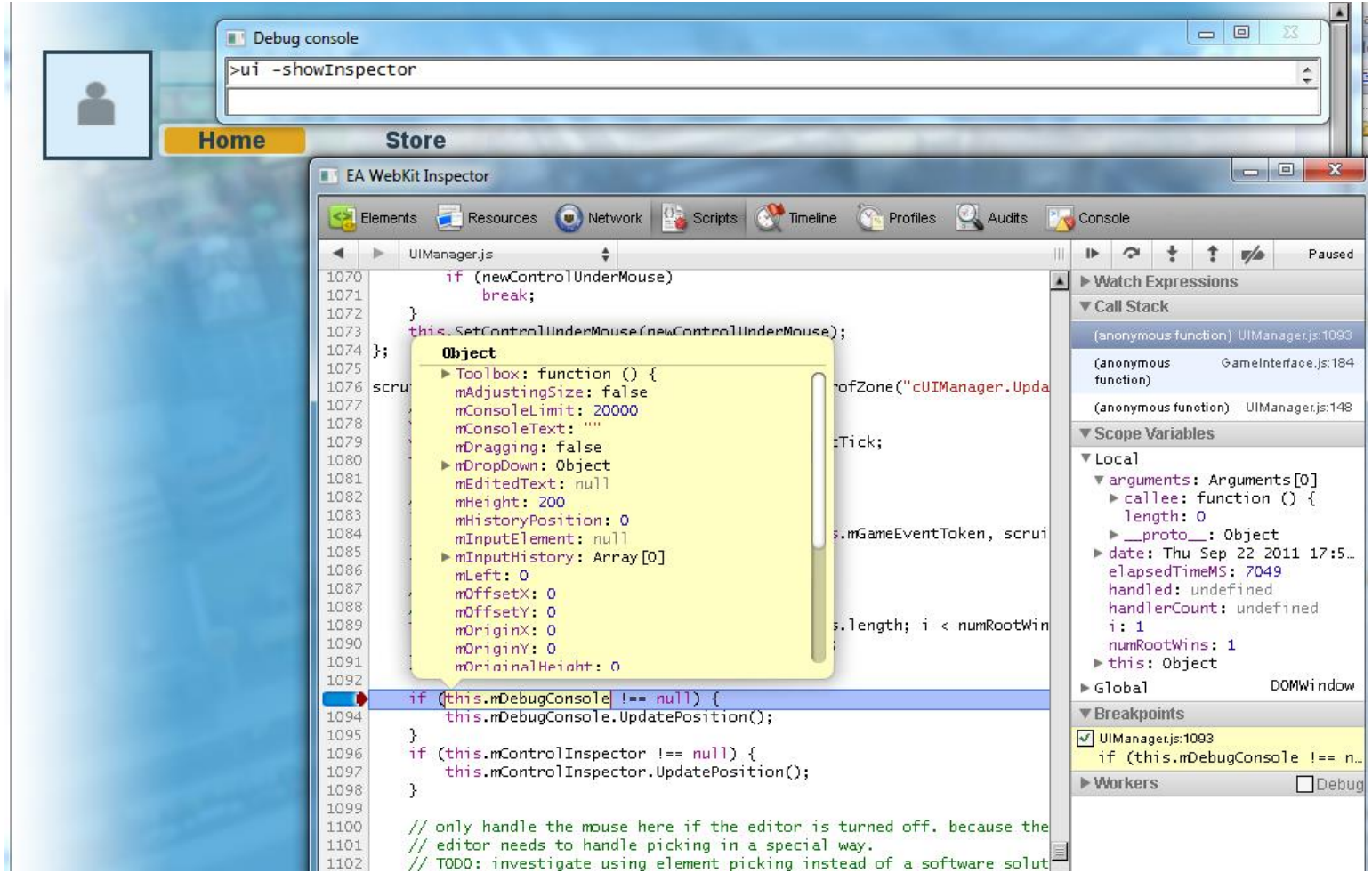
- Get all the benefits of active development (and drawbacks).
- WebKit's modular design helps adapt to other platforms.
- Good inspector for live inspection, JavaScript debugging
- Gotcha: Doesn't work on mobile (platform limitations)

EA WebKit

- Features:
  - Supports multiple views
  - Hardware compositing API
  - Efficient JavaScript bindings
  - Designed for games, support plugins for custom text renderer, memory allocators, etc
  - (First party support)


City
4:58 PM
Emeryville
50,000
+0 / HR
0


City
5:38 PM
Emeryville
50,000
+0 / HR
0


Debug console
>ui -showInspector
Home
Store
EA WebKit Inspector
Elements
Resources
Network
Scripts
Timeline
Profiles
Audits
Console
UIManager.js
Paused
Watch Expressions
Call Stack
(anonymous function) UIManager.js:1093
(anonymous function) GameInterface.js:184
(anonymous function) UIManager.js:148
Scope Variables
Local
arguments: Arguments[0]
callee: function () {
length: 0
__proto__: Object
date: Thu Sep 22 2011 17:5…
elapsedTimeMS: 7049
handled: undefined
handlerCount: undefined
i: 1
numRootWins: 1
this: Object
Global DOMWindow
Breakpoints
UIManager.js:1093
if (this.mDebugConsole !== n…
Workers
Debug
Object
Toolbox: function () {
mAdjustingSize: false
mConsoleLimit: 20000
mConsoleText: ""
mDragging: false
mDropDown: Object
mEditedText: null
mHeight: 200
mHistoryPosition: 0
mInputElement: null
mInputHistory: Array[0]
mLeft: 0
mOffsetX: 0
mOffsetY: 0
mOriginX: 0
mOriginY: 0
if (this.mDebugConsole !== null) {
this.mDebugConsole.UpdatePosition();
if (this.mControlInspector !== null) {
this.mControlInspector.UpdatePosition();
// only handle the mouse here if the editor is turned off. because the
// editor needs to handle picking in a special way.
// TODO: investigate using element picking instead of a software solut


Debug console
>ui -showInspector

MUiLE

- HTML/JavaScript-based UI layer.
- Custom to Maxis.
- Built most of the functionality from the ground up as we couldn’t find good alternatives at the time.
- Just implementing a button with the correct behaviors took some time…

MUiLE

- All UI 100% in HTML/JS/CSS.
- Component based, storing layouts in JSON files, which allow us to merge and allow concurrent edits.
  - Layout files then loaded in dynamically and the DOM is constructed from them.
- Layouts can link to other layouts, allowing reusability.


```
{
 "instanceID": 1,
 "left": 205,
 "top": 98,
 "width": 800,
 "height": 600,
 "visibility": true,
 "ignoreMouse": true,
 "children": [ {
   "instanceID": 2,
   "left": 10,
   "top": 10,
   "width": 176,
   "height": 45,
   "visibility": true,
   "drawable": {
    "type": 2,
    "images": [ "Graphics/textInputField.png" ]
   },
   "type": "cWindow"
  },
```

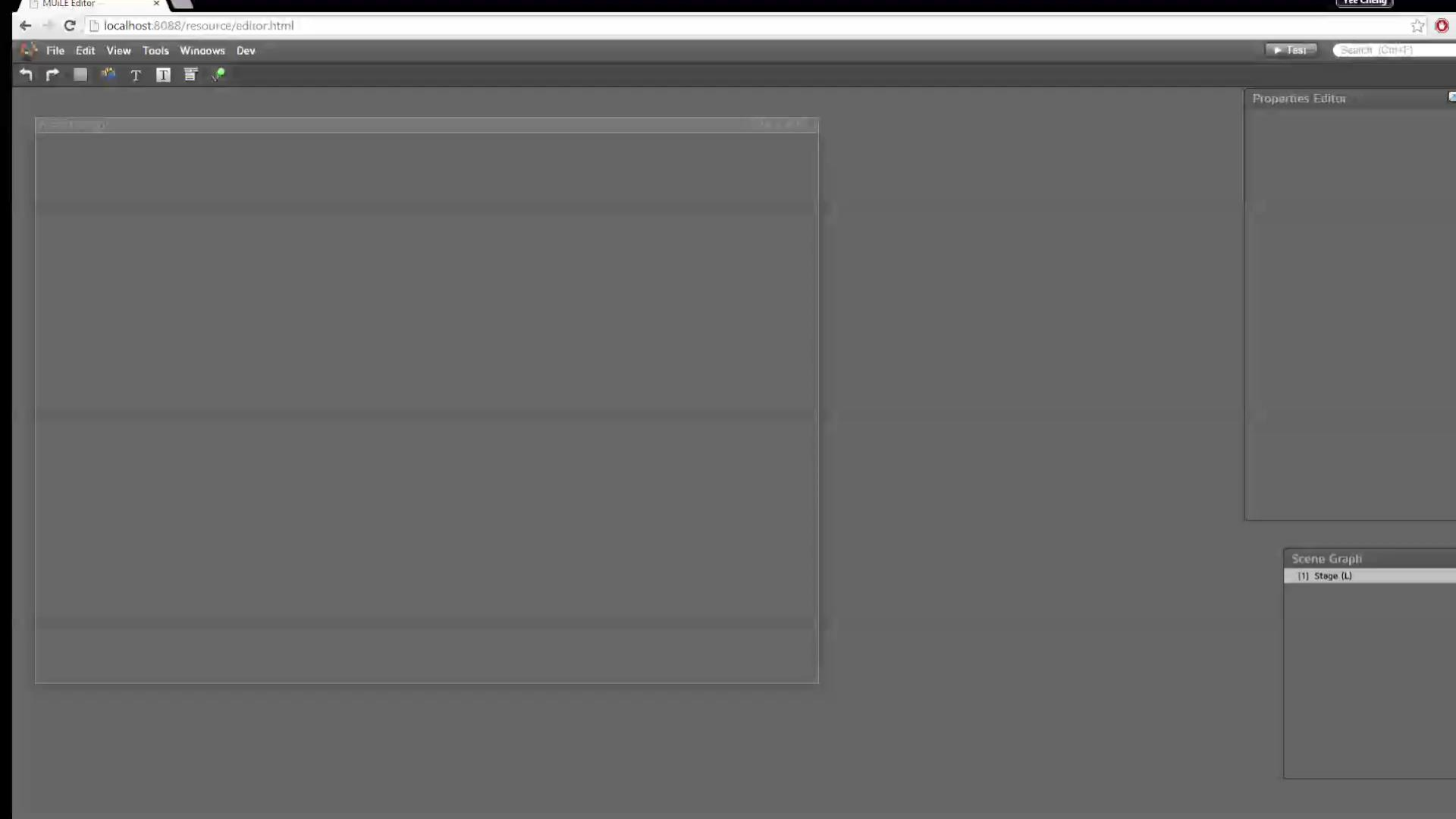
```
{
   "layoutPath": "Layouts/GlobalUI2.js",
   "instanceID": 3,
   "left": 0,
   "top": 0,
   "width": 800,
   "height": 600,
   "controlID": 174136993,
   "visibility": true
  } ],
 "type": "cLayout",
 "version": 1
}
```

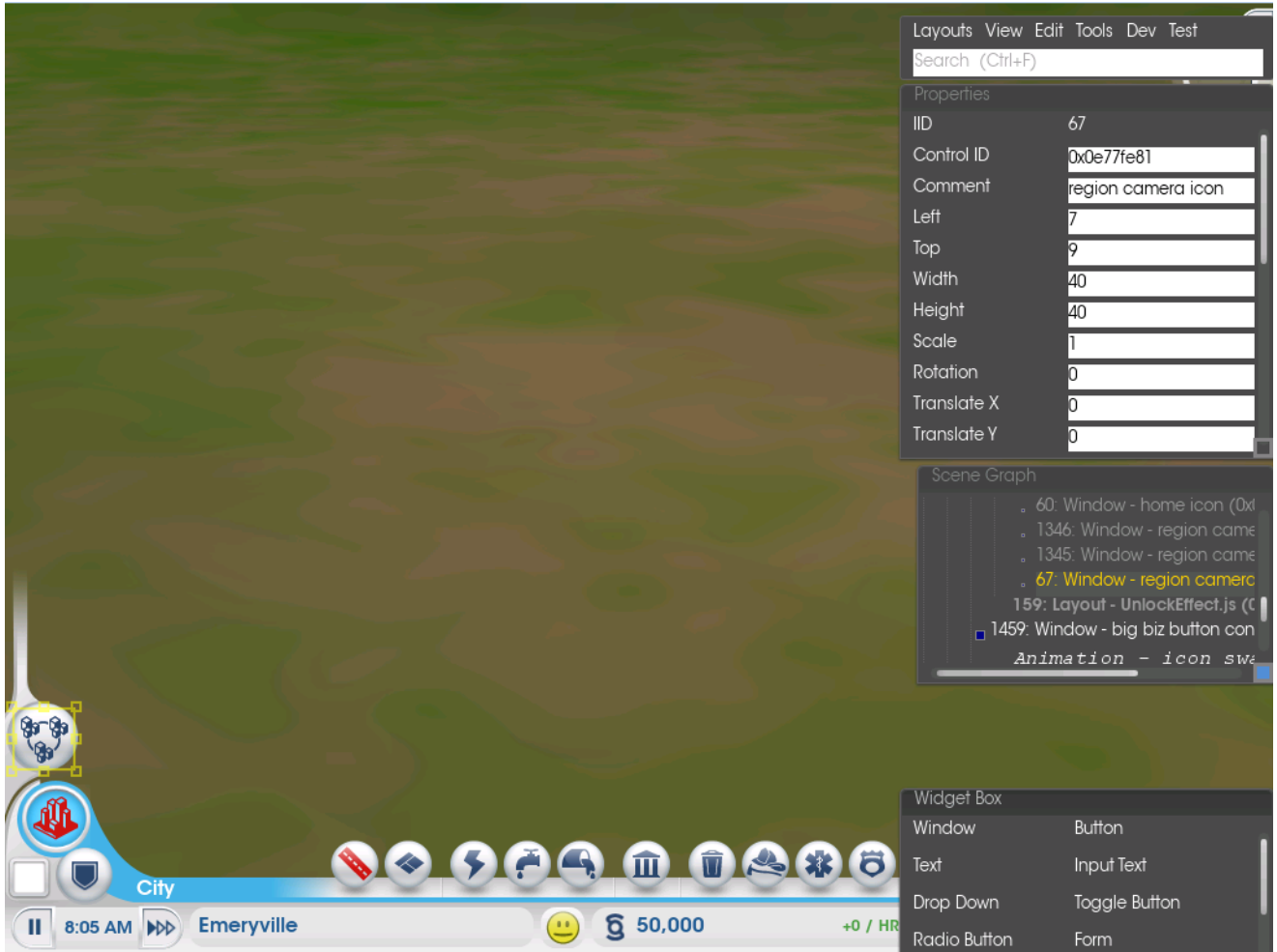
MUiLE

- In game communication is done through async callbacks, through game commands, game events, and game data callbacks.
  - Better for multithreading, and similar to the async nature of web interfaces
  - `PostGameCommand(kCmdDoSomething, someData, function(result) { /* got result! */});`
  - `RequestGameData(kDataPopulationCount, function(data) { /* process data */ });`

MUiLE Editor

- WYSIWYG editor, also built in HTML as part of the package itself, allowing it to be used in any browser.
  - No dependencies. Anyone with a debug version of the game can edit UI using a browser.
  - Can edit the UI in-game.


MUILE Editor
localhost:8088/resource/editor.html
File
Edit
View
Tools
Windows
Dev
Test
Properties Editor
Scene Graph
[1] Stage (L)


Layouts View Edit Tools Dev Test
Search (Ctrl+F)
Properties
IID 67
Control ID 0x0e77fe81
Comment region camera icon
Left 7
Top 9
Width 40
Height 40
Scale 1
Rotation 0
Translate X 0
Translate Y 0
Scene Graph
60: Window - home icon (0x
1346: Window - region came
1345: Window - region came
67: Window - region camera
159: Layout - UnlockEffect.js (
1459: Window - big biz button con
Animation - icon swa
Widget Box
Window
Button
Text
Input Text
Drop Down
Toggle Button
Radio Button
Form
City
8:05 AM
Emeryville
50,000
+0 / HR

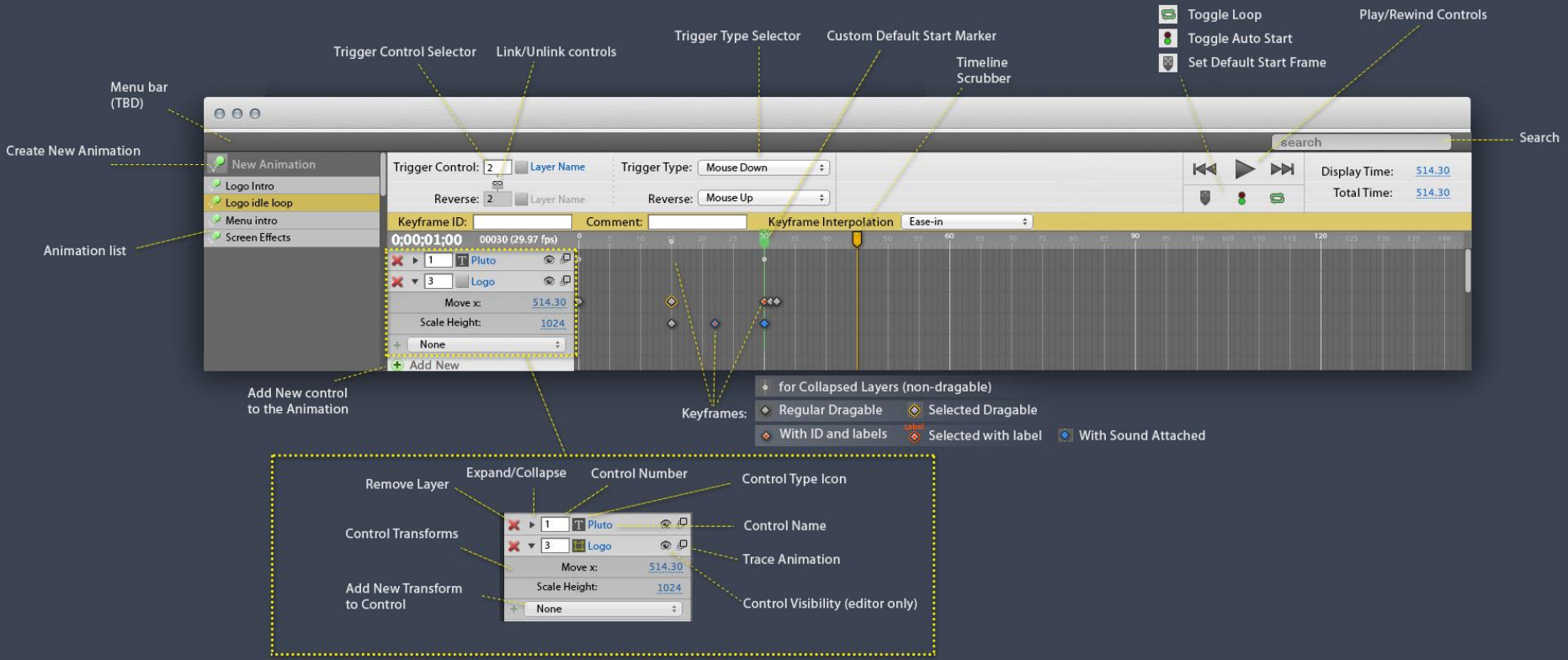
MUiLE Editor

- Communicates with game through a localhost server served by the game (we already use that for other debugging utilities)
  - Uses a REST-like API. When in game we expose a custom URL handler (game://localhost/) instead.
  - Editor: `http://localhost/resource/editor.html`
  - List all the layouts: `http://localhost/dir/layouts` (GET)
  - Save layout: `http://localhost/layout/mainMenu.json` (POST)
  - Load layout: `http://localhost/layout/mainMenu.json` (GET)

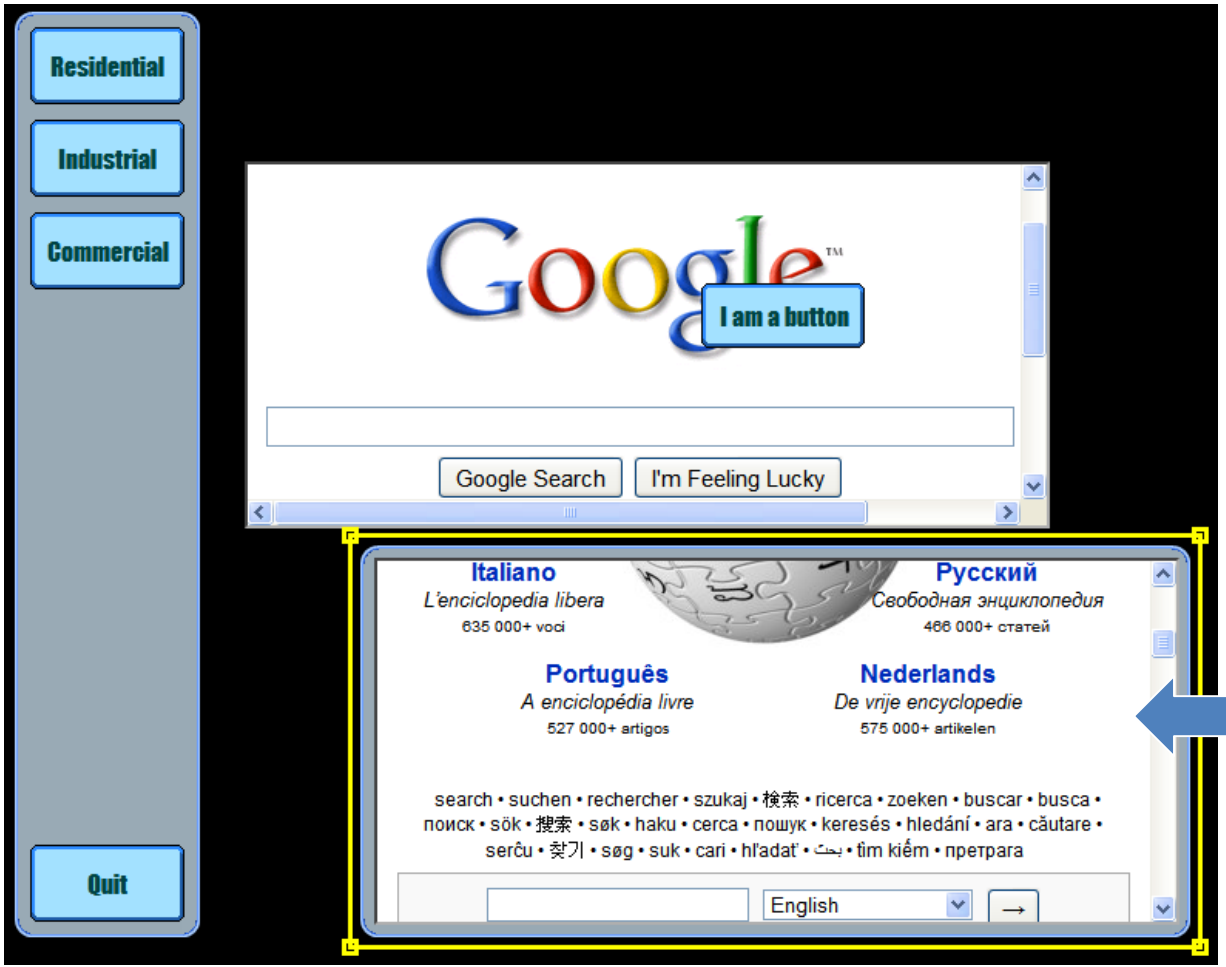
MUiLE Animations

- Recommended way is to use CSS
  - Fade in/out, transitions, keyframes etc.
  - When we started it wasn't as advanced, and we found out we also wanted more control.
- Implemented custom animation system
  - Full control over timeline, can scrub, stop, loop.
  - Controls exactly the parameters we need.
  - Probably increased load in the JS engine as they aren't natively animated like in CSS.
  - Each control has triggers to play/stop/loop animations. Each animation is a timeline of different controls' states such as positions, visibilities, rotations, or game events.

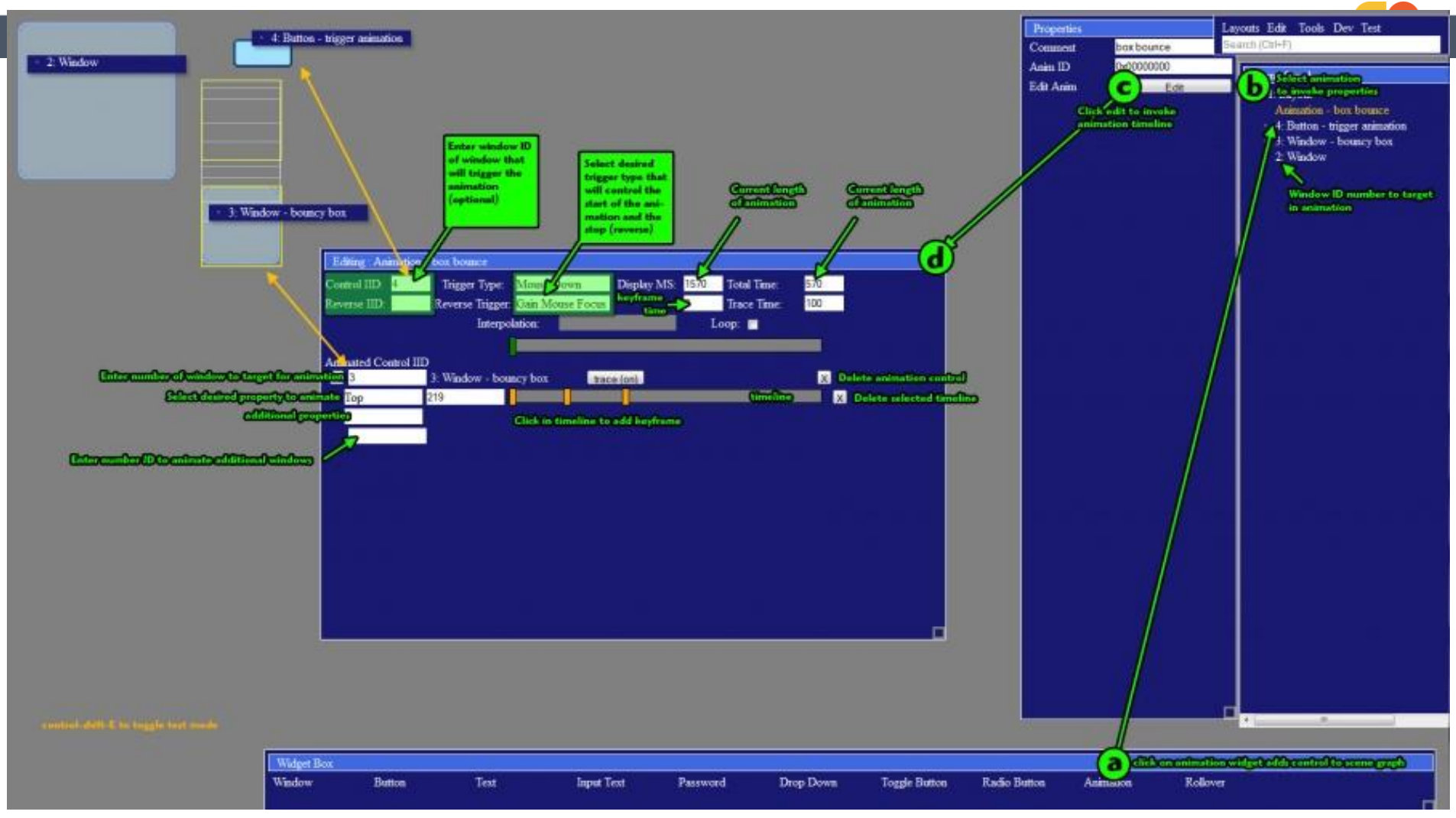
UI Tool Timeline

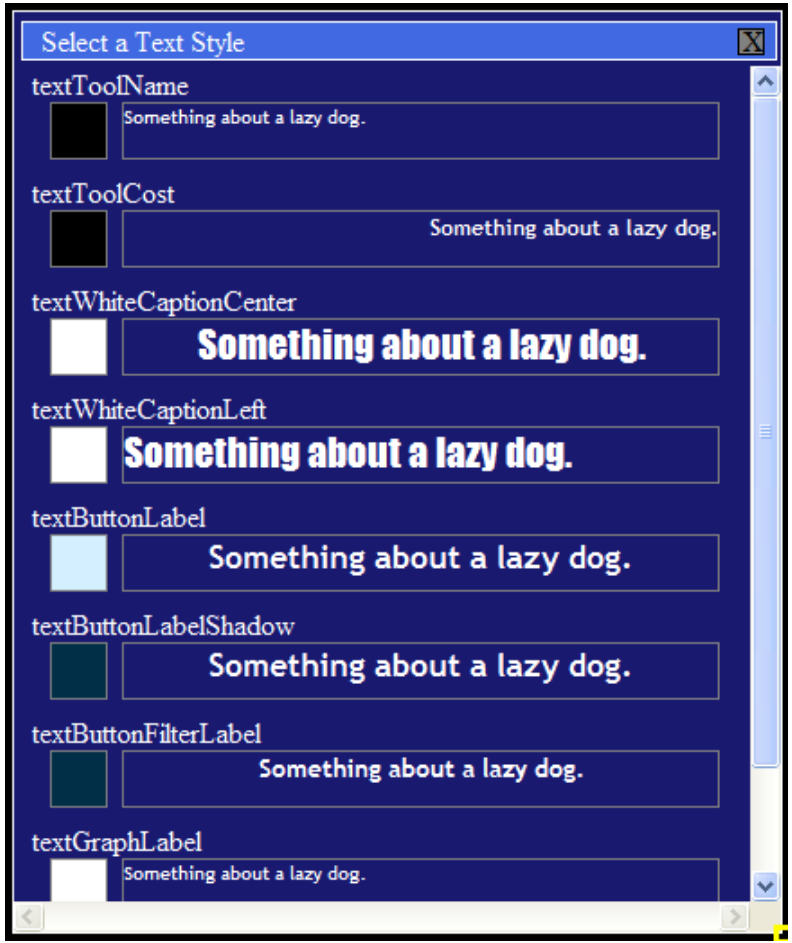


Early Sceenshots



Make sure to use sandbox flag for your iframes!

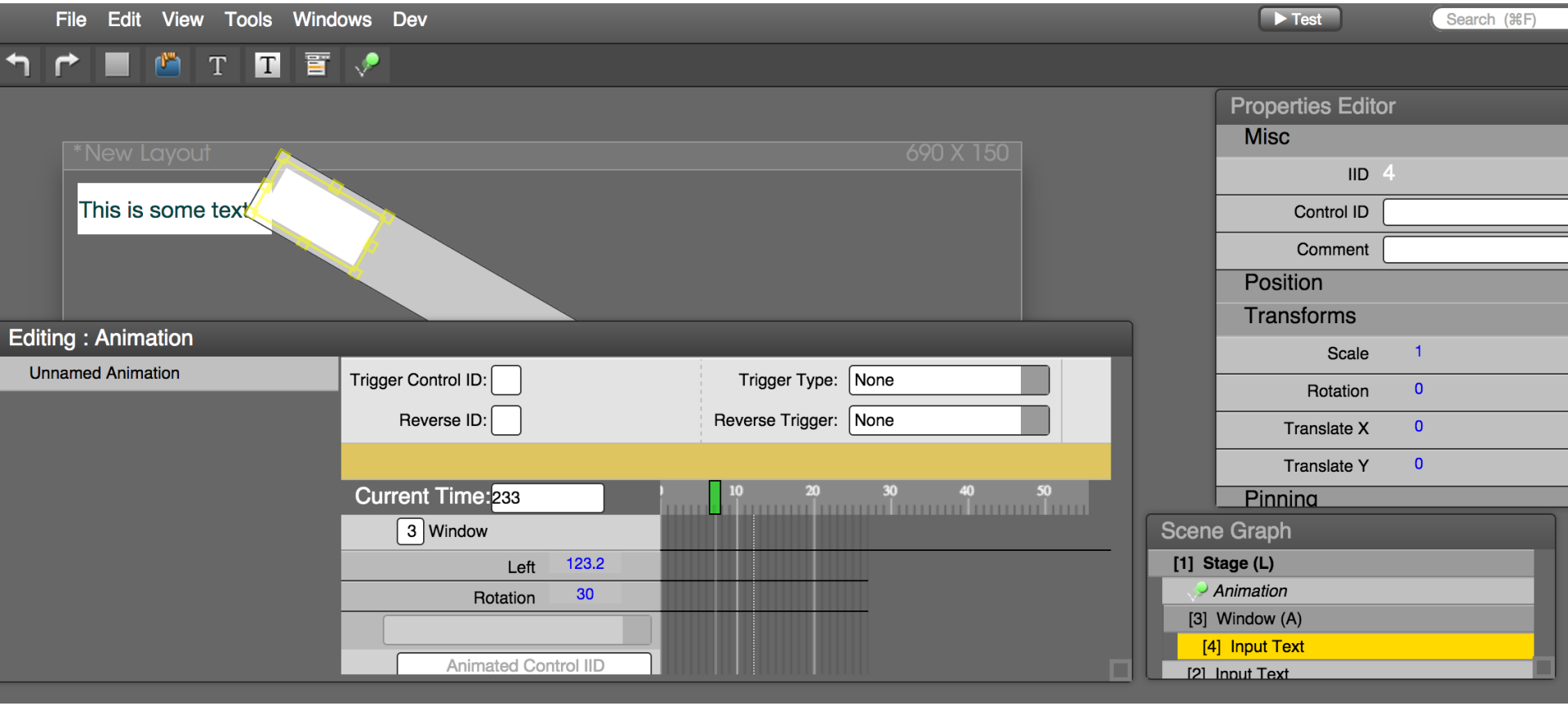

2: Window
4: Button - trigger animation
3: Window - bouncy box
Enter window ID of window that will trigger the animation (optional)
Select desired trigger type that will control the start of the animation and the stop (reverse)
Current length of animation
Current length of animation
Editing: Animation - box bounce
Control IID: 4
Trigger Type: Mouse Down
Display MS: 1570
Total Time: 570
Reverse IID:
Reverse Trigger: Gain Mouse Focus
keyframe time
Trace Time: 100
Interpolation:
Loop:
Animated Control IID
Enter number of window to target for animation
3
3: Window - bouncy box
track (on)
Delete animation control
Select desired property to animate
Top
219
timeline
Delete selected timeline
additional properties
Click in timeline to add keyframe
Enter number ID to animate additional windows
Properties
Comment
box bounce
Anim ID
0x00000000
Edit Anim
Edit
Click edit to invoke animation timeline
Layouts Edit Tools Dev Test
Search (Ctrl+F)
Select animation to invoke properties
Animation - box bounce
4: Button - trigger animation
3: Window - bouncy box
2: Window
Window ID number to target in animation
a
b
c
d
Widget Box
Window
Button
Text
Input Text
Password
Drop Down
Toggle Button
Radio Button
Animation
Rollover
click on animation widget adds control to scene graph

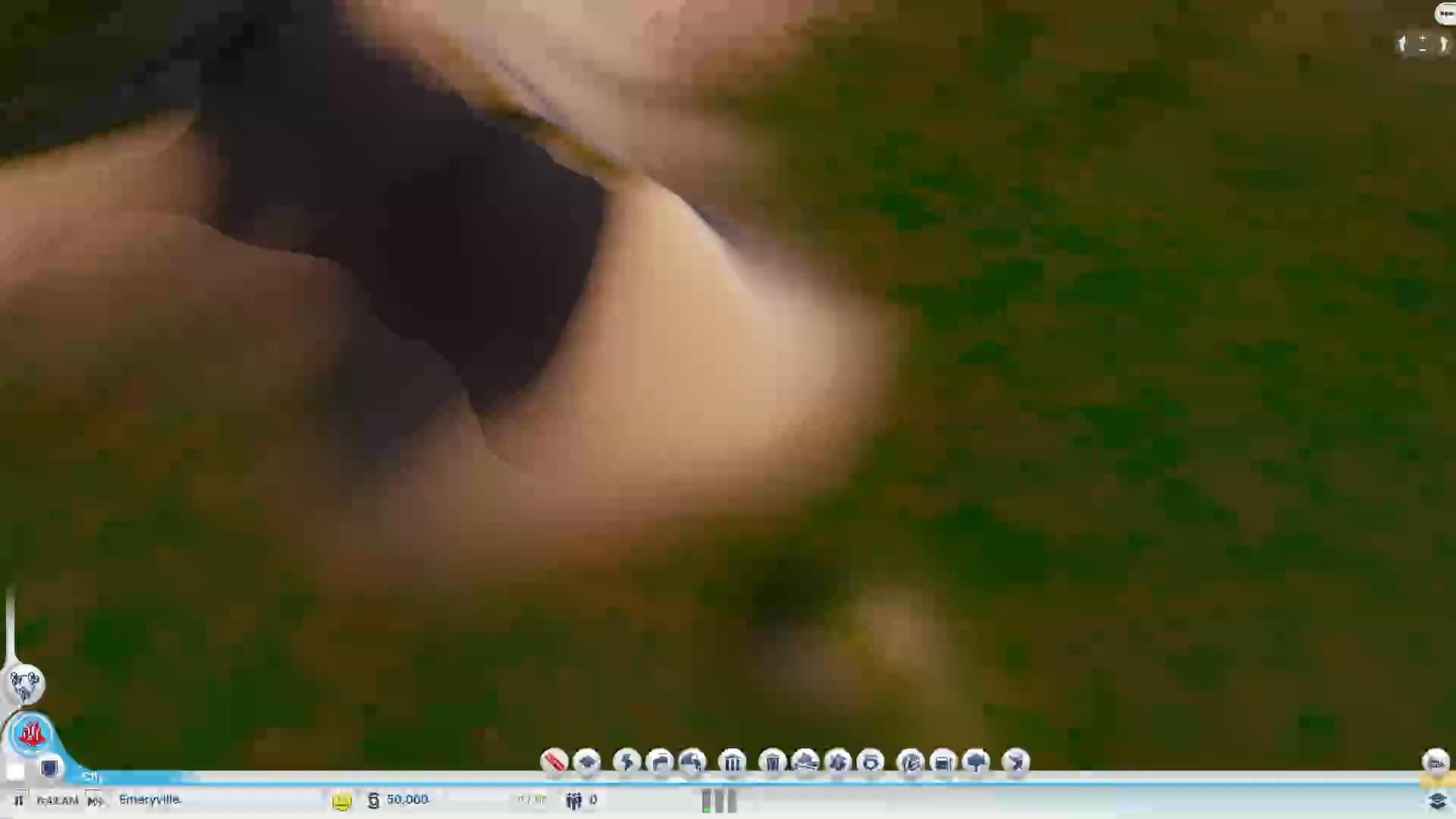

Select a Text Style
textToolName
Something about a lazy dog.
textToolCost
Something about a lazy dog.
textWhiteCaptionCenter
Something about a lazy dog.
textWhiteCaptionLeft
Something about a lazy dog.
textButtonLabel
Something about a lazy dog.
textButtonLabelShadow
Something about a lazy dog.
textButtonFilterLabel
Something about a lazy dog.
textGraphLabel
Something about a lazy dog.


Small Coal Power Plant
0 / 24
3 kWh
-55 / hour
3 gal/hour


Small Coal Power Plant
0 / 24
3 kWh
4,160 / hour
-55 / hour
3 gal/hour
On
Deliver
On

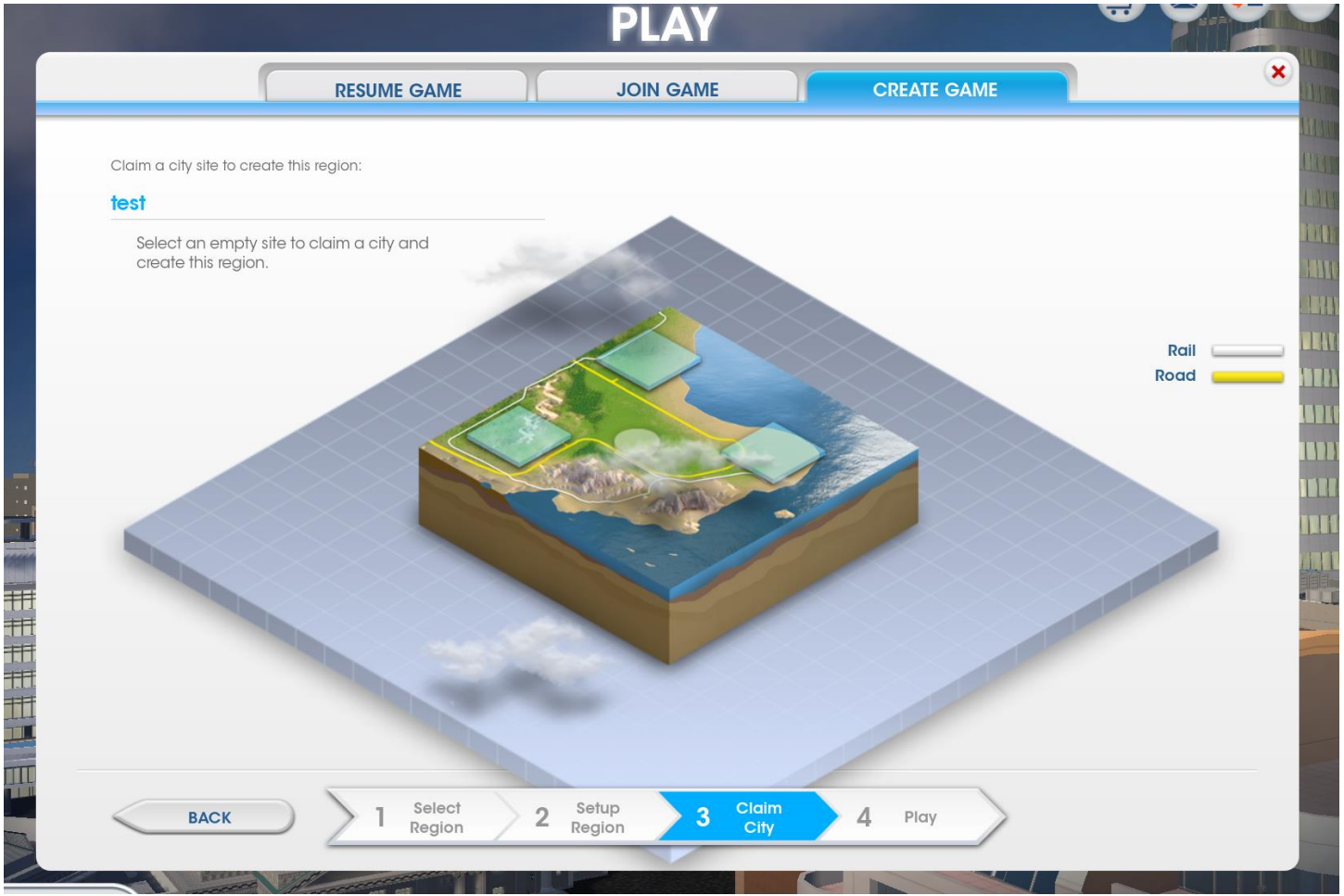

City
12:00 PM
Emeryville
50,000
0


File Edit View Tools Windows Dev
Test
Search (⌘F)
*New Layout
690 X 150
This is some text
Properties Editor
Misc
IID 4
Control ID
Comment
Position
Transforms
Scale 1
Rotation 0
Translate X 0
Translate Y 0
Pinning
Editing : Animation
Unnamed Animation
Trigger Control ID:
Trigger Type: None
Reverse ID:
Reverse Trigger: None
Current Time: 233
10 20 30 40 50
3 Window
Left 123.2
Rotation 30
Animated Control IID
Scene Graph
[1] Stage (L)
Animation
[3] Window (A)
[4] Input Text
[2] Input Text


9:43 AM
Emeryville
50,000
0

Final Shipped UI


Server: North America West 3
MULTIPLAYER MODE
SIMCITY™
SINGLE-PLAYER
MULTIPLAYER
RESUME
PLAY
SIMCITY WORLD
SimCity is better with friends!
Your Origin friends list is in game, filtered by Origin friends who own SimCity.
When you start the game your online status is visible to your friends. Click the button next to your avatar to change your status.
CityLog is a game event feed broadcasting events in SimCity World. Your CityLog feed is currently set to share with your friends. If you would like to change your settings, go to the Settings menu.
OKAY
OFFLINE IS HERE
LEARN MORE
CITYLOG


PLAY
RESUME GAME
JOIN GAME
CREATE GAME
Claim a city site to create this region:
test
Select an empty site to claim a city and create this region.
Rail
Road
BACK
1 Select Region
2 Setup Region
3 Claim City
4 Play


Breaking Ground
Draw a road from the highway
Zone residential and have 10 more houses
Roads
Guides
City
12:16 PM
Trinity Point
1,000,000
+0 / HR
0


VISITING
Giovanni Woods
MaxisScott
GIOVANNI WOODS POPULATION
Welcome to the Population Panel!
Increase land value to get wealthier Sims to move in and zone to grow your population.
Region: 2,051
City: 2,051
Visiting: 0
History
Detail
CITY SIZE
CITY
VISITING
Metropolis
5,415
0
May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr
POPULATION SOURCE
Locals
Visitors
POPULATION WEALTH
Low
Medium
High
6:58 PM
Giovanni Woods
406,587
-512 / HR
2,051

What this talk is about

1. Why HTML*?
2. Maxis' UI, based on HTML. Shipped with SimCity
3. **Tips and gotchas**

Content Creation

- Unless you are using this for limited use cases, we suggest having a good WYSIWYG editor. Your UI artists will thank you.
- We built our own but there should be a lot more choices now (Adobe Edge, etc)

Content Creation

- Traditional split of HTML/CSS/JS is such that they represent content, style, and logic. Good for representing documents.
- Essentially building a mini-application. Built most UI out of JSON modules that are loaded in dynamically through JS instead.

Pick a Good Engine

- Access to source code. You really need to be able to dig into it if things go wrong.
- Find one which you can get good support from.
- Allow custom memory allocator hooks, etc. You need the control.
- Has hardware rendering support and provides hooks for it.
- Comes with a standalone demo app for you to test pages on.
- Supports JIT compilation
  - JIT mode is at least twice as fast for us

JavaScript Organization

- Be careful with common libraries such as jQuery.
  - They may be great for web development with tons of features, they may not give the best per-frame performance or memory-use.
  - Make sure to profile before you commit!
- Read *JavaScript: The Good Parts*

JavaScript Organization

- We used the Google Closure Library
  - Library developed in a modular fashion, allowing you to selectively pull in only the necessary components.
  - Contains useful functions for matrix calculations, cryptography, basic utilities for inheritance, etc.
  - Open Sourced

JavaScript Organization

- Google Closure Library (continued)
  - Solves the issue of managing large amount of JS files. Other solutions usually involve just concatenating them all or modifying HTML files.
  - Allows you to specify dependencies among JS files and build up a manifest JS files that pull in all necessary dependencies.
    - May be less useful with advent of Common JS.

JavaScript Organization

- Standard JS annoyances

```
<html><body>
    <script src="ControlInspector.js"></script>
    <script src="UIAnimationEdito.js"></script>
    <script src="UIEditorDropDown.js"></script>
    <script src="UIEditor.js"></script>
…
```

JavaScript Organization

- Google Closure Library dependencies example

Project.js:

```
goog.provide('muile.editor.project');

goog.require('muile.project'); // pull in the general muile library

goog.require('muile.editor.ControlInspector');
goog.require('muile.editor.UIAnimationEditor');
goog.require('muile.editor.UIEditor');
goog.require('muile.editor.UIEditorDropDown');
goog.require('muile.editor.UIEditorProperties');
...
```

EditorControlInspector.js:

```
goog.provide('muile.editor.ControlInspector');
...
```

Editor.html:

```
<script src="Project.js"></script> <!- this automatically pulls in the other files -->
```

JavaScript Organization

- Google Closure Compiler
  - Designed to go with Closure library
  - Allows you to “compile” all JS files into one after analyzing dependencies.
    - Because of the way it works compiled and uncompiled code may work differently if dependencies weren’t correctly specified!
  - Generates source maps to allow debugging compiled files in debugger (similar to .pdb files)
  - 2 optimization modes: simple and advanced. Advanced mode requires much more aggressive changes to code but could lead to big gains

JavaScript Organization

- Google Closure Compiler (continued)
  - Examples of advanced mode compilation:

```
var DEBUG = false;
var counter = 1;
if (DEBUG) {
  console.log('Super secret output:' + counter++);
}
console.log('Generic boring output:' + counter);
```

  - *Compiled to:*

```
console.log("Generic boring output:1");
```

JavaScript Organization

- Communicating with the game
  - Use C++ bindings
  - You could try to be cute and use REST APIs
    - `game://localhost/Game/Commands/Sim/AddPopulation/1`
  - You are kind of adding unnecessary cost for string parsing etc. Just call the C++ function.
    - `Game.AddPopulation(1)`

Performance

- Rendering
  - We used software rendering for SimCity as we didn’t have good hardware compositing support yet.
  - Performance was mostly fine but animating stacked opacity killed performance. Easily created 10ms hitches without knowing why.
  - Switch to a hardware compositing model now used by some browsers.
    - http://www.chromium.org/developers/design-documents/gpu-accelerated-compositing-in-chrome

Performance / Rendering

- Make sure can do dirty rect visualization.
- If you are using WebKit or Blink based browsers, use translateZ(0) to force elements into another layer if using hardware compositing. Only do this for animating elements.


City
7:47 AM
Emeryville
50,000
+0 / HR
0
Network


City
8:20 AM
Emeryville
50,000
+0 / HR
0

Performance

- Memory use
  - Watch your memory use! We found that it’s not easy to profile memory used by the UI system as we would get a global heap using up to 100mb of memory, with no finer details. Blink-based browsers seem to have better control over this.
  - Try to use pools instead of dynamic allocation as much as possible

Performance

- C++ / JS bindings
  - Try to reduce communications between C++ and JS code. The bridge is not optimized.
  - You may have to cache some data on both sides to prevent back-and-forth communications.
  - Don't do something like this every time

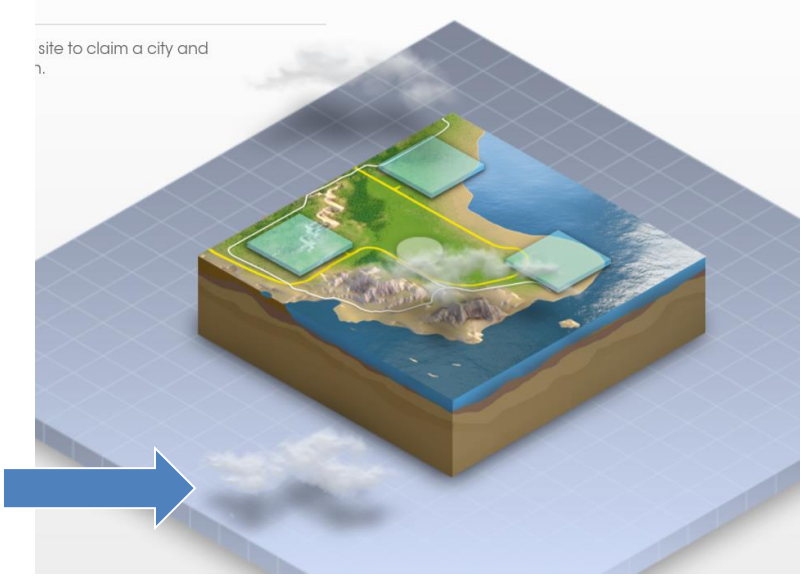
```
uiView->evalJS("someCharacter.ShowHealth()");
```

  - This requires a recompilation. Hopefully your engine of choice can cache JS functions so you can do this instead:

```
auto cacheFunction = uiView->evalJS("(function()  {
        someCharacter.ShowHealth();
})");
cacheFunction.Evaluate(); // this will be much more efficient!
```

Grab bag

- Scrolling text and images smoothly is surprisingly hard!



Non-PC platform issues

- Console
  - Mostly works, but you won’t get JIT-compiled JavaScript code. Reduce JS workload and budget accordingly
- Mobile
  - Especially on iOS it’s not possible to ship your own HTML/JS runtime, so need to use native web view.
  - iOS now supports JIT through WKWebView

Wrap up

- It’s possible to make quality UI using HTML
- Tools and libraries available.
- Building our own tools and editor was very time consuming.
- Performance *will* be less than native UI
- Improved iteration time and ease of development was worth it.

Thanks!

- **Brad Smith** and **Scott Clarke**, who did a lot of the actual work on this.
- **Renaud Ternynck** for his continuous bombardment of request for features and improvements.
- **EA WebKit team** for their support throughout.
- The entire **Maxis UI team** for making this all possible.

ychin@maxis.com

Q & A