Sixspeedmedia.com - Development history, features and notes

This document outlines the technical development and innovations of Sixspeedmedia.com, including its integration with IViD (Interactive Video Display) systems and their shared focus on interactive, scalable design. Key sections include:

1. Development Timeline:

- **2002 to 2003**: Tracks the launch and evolution of Sixspeedmedia.com, including foundational elements like elliptical icon layouts, scalable UI templates, and modular navigation systems.
- Integration with IViD prototype, which introduced motion-tracking and physics-based inputs for interactive systems.

2. Innovative Features:

- **Elliptical and Carousel Navigation**: Early designs implemented dynamic elliptical icon layouts, evolving into 3D and 2D carousels with inertia-based scrolling and parallax effects.
- **Blur and Transition Effects**: Developed real-time blur and animated transitions for enhanced UI interactivity and user focus.
- **Data-Driven UI**: Enabled dynamic content updates through external data sources, laying groundwork for modular, skinnable designs.

3. Technical Implementation:

- **Codebase**: Features elliptical math, scalable UI elements, and modular templates. IViD integration focused on hardware coordination between video projectors, cameras, and interaction mapping.
- **Media Integration**: Combined vector graphics, video, and dynamic imagery into seamless user experiences.
- **Physics-Driven Interactions**: Both IViD and carousels utilized physics principles, such as inertia and touch responsiveness, for smoother user engagement.

4. Visual Documentation:

- Includes screenshots and design diagrams of early prototypes, such as the elliptical carousel interface and IViD's interactive shadow-tracking technology.
- Demonstrates IViD applications in entertainment and commercial environments, including real-world examples like Rainforest Café's interactive rain effects.

5. Milestones:

2002, Aug 9 Domain created*, nameservers added

Nameservers
ns1.fastnameserv.com
ns2.fastnameserv.com

6speedmedia.com also purchased

From: billing@featureprice.com

Sent: Tuesday, November 26, 2002 1:08 AM

To: ayeckel@go2marine.com

Subject: ACCOUNT INFORMATION

Dear Andy,

Thank you very much for your web hosting order.

Below, you will find your account information as well as important instructions with helpful and mandatory information for operating your web hosting account. Please read this letter carefully and in full - many questions that you may have are answered immediately in this email.

1.) CLIENT ACCOUNT INFORMATION

==============================

Domain Name: www.sixspeedmedia.com

Dedicated IP Address: 64.127.195.77

Username: ________(removed for security)

Password: _______(removed for security)

Servername: UNIX239

1. **Data-Driven UI Specification [(May 2002 Document](#))**
   - Envisioned a dynamic interface capable of pulling in content (images, text) automatically, rather than embedding everything manually.
   - Laid out the concept of having multiple "categories" or "modules," each potentially skinnable or themed.

    - Proposed the idea of a "parent shell" that could host different content blocks/mini-sites without reloading the entire interface.
2. **Initial Elliptical Icon Layout (2D "Arc" Arrangement)**
    - Menu items arranged along a curved path (an ellipse or arc) rather than in a simple row or column.
    - Early versions scaled icons into the scene rather than animating them around the ellipse.
    - Served as the foundation for later 2D/3D carousel experiments.
3. **Simulated 3D Carousel for Navigation**
    - A progression from the static ellipse into a "carousel" effect that positions icons/items in (simulated) 3D space.
    - Depth and perspective created by scaling and layering 2D elements, giving the illusion of rotation around a center point.
    - Solved the problem of handling varying numbers of items by dynamically adjusting positions around a "virtual center."
4. **Flattened 2D Carousel (Tier-2 Navigation)**
    - A secondary navigation system that "flattened" the 3D effect, displayed vertically, though it was intended to be used horizontally as seen by later interfaces.
    - Often used for sub-pages or category-level content, where a big 3D rotation was overkill but the same "carousel mechanics" (sliding/inertia) still applied.
5. **Inertia/Scrolling Mechanics**
    - Applied a small amount of physics-based calculation so that when an icon was dragged or flicked, it continued moving briefly (slowing down) instead of stopping immediately.
    - Precursor to the "touch flick" or "momentum scroll" found in later multi-touch libraries.
6. **Blur Transition / Visual Effects Tests**
    - Prototype where the interface performed a channel-based blur (or alpha-blur) when moving or transitioning between states.
    - Used overlay or alpha masking to show a blurred version of an image behind the active element.
    - Demonstrated manual control of the blur radius based on mouse movement or "distance" from the center, creating a focus/defocus effect.
7. **"Template" System for Skins/Themes**
    - A set of base SWF (Flash) files that could be swapped to change the look/feel of specific categories (e.g., photography, print layout, demos).
    - Ensured that the underlying navigation logic stayed consistent while visuals changed per theme or project.
    -
8. **Scalable UI Elements / Single-Pixel Stretching**
    - Used "1 px–wide" or "1 px–tall" bitmaps (plus gradients) stretched horizontally/vertically to fill dynamic layouts at any resolution.
    - Aimed at letting the same site scale or adapt to different screen sizes without redesigning the graphics from scratch.
9. **Animated Transitions & Layering**
    - Pages were rarely static: items faded, slid, or scaled into view.

    - Consistent layering approach allowed background images or videos to remain behind navigational elements (a mild parallax effect at times).

10. **Multimedia Integration**

- Incorporated short videos, dynamic image galleries, or audio for interactive buttons.
- Early usage of embedded media demonstrating how a single SWF could load and swap in external files on demand.

11. **Hierarchical Navigation Concept**

- Primary (3D) carousel for top-level sections; flattened or simpler 2D carousels (or elliptical menus) for subcategories.
- Introduced the notion of infinite or "modular" expansions, each reusing the same fundamental carousel/ellipse logic.

12. **"Spec Document"–Style Roadmap**

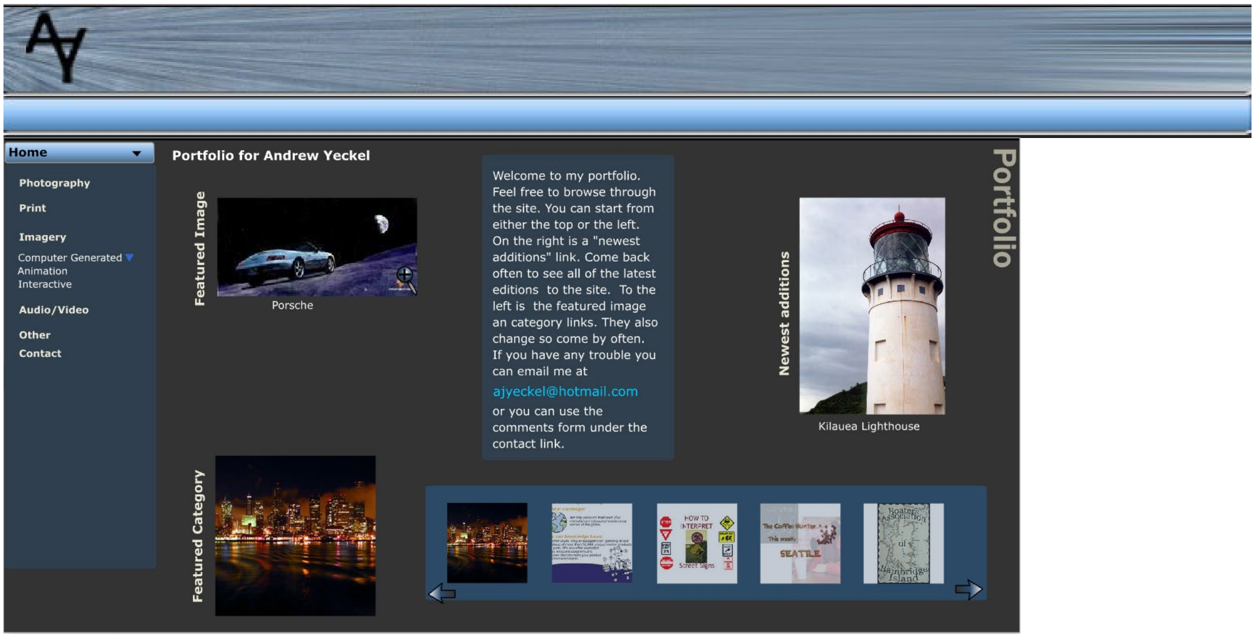
- Detailed how all these modules (carousels, blur filters, data binding, etc.) might eventually tie together into a full data-driven system (somewhat forward-looking given 2002 technology constraints).
- Provided a blueprint for future expansions: multi-tiered, skinnable UI

13. **RGB Transitions**
    A technique where an image was split into its red, green, and blue channels. Each channel could be briefly animated, offset, or faded in/out independently before recombining, creating a distinctive "ghosting" or glitch-like effect during transitions. Part of the project template

Timeline:

2002 spec beta flash version (pre sixspeedmedia.com);

First called AY then Carbon Media then Sixspeedmedia.

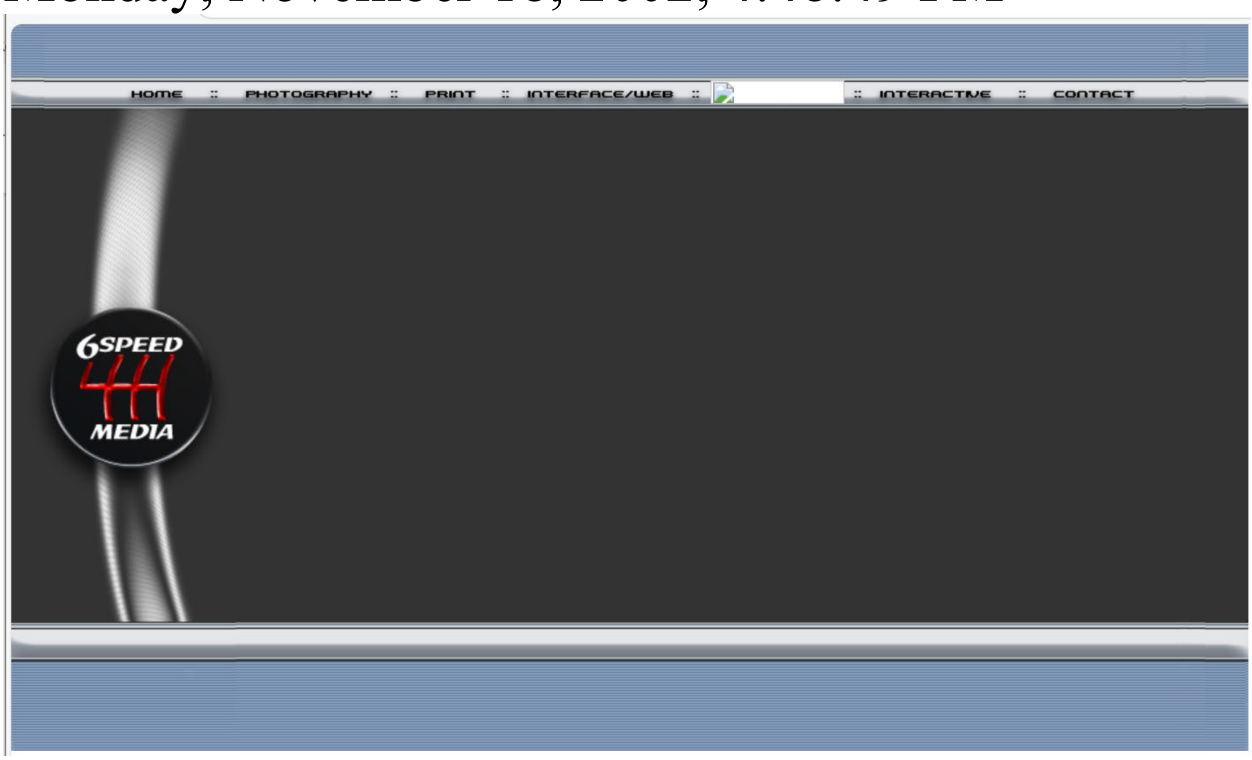

Home
Photography
Print
Imagery
Computer Generated
Animation
Interactive
Audio/Video
Other
Contact
Portfolio for Andrew Yeckel
Featured Image
Porsche
Welcome to my portfolio. Feel free to browse through the site. You can start from either the top or the left. On the right is a "newest additions" link. Come back often to see all of the latest editions to the site. To the left is the featured image an category links. They also change so come by often. If you have any trouble you can email me at
ajyeckel@hotmail.com
or you can use the comments form under the contact link.
Newest additions
Kilauea Lighthouse
Portfolio
Featured Category


CARBON
media

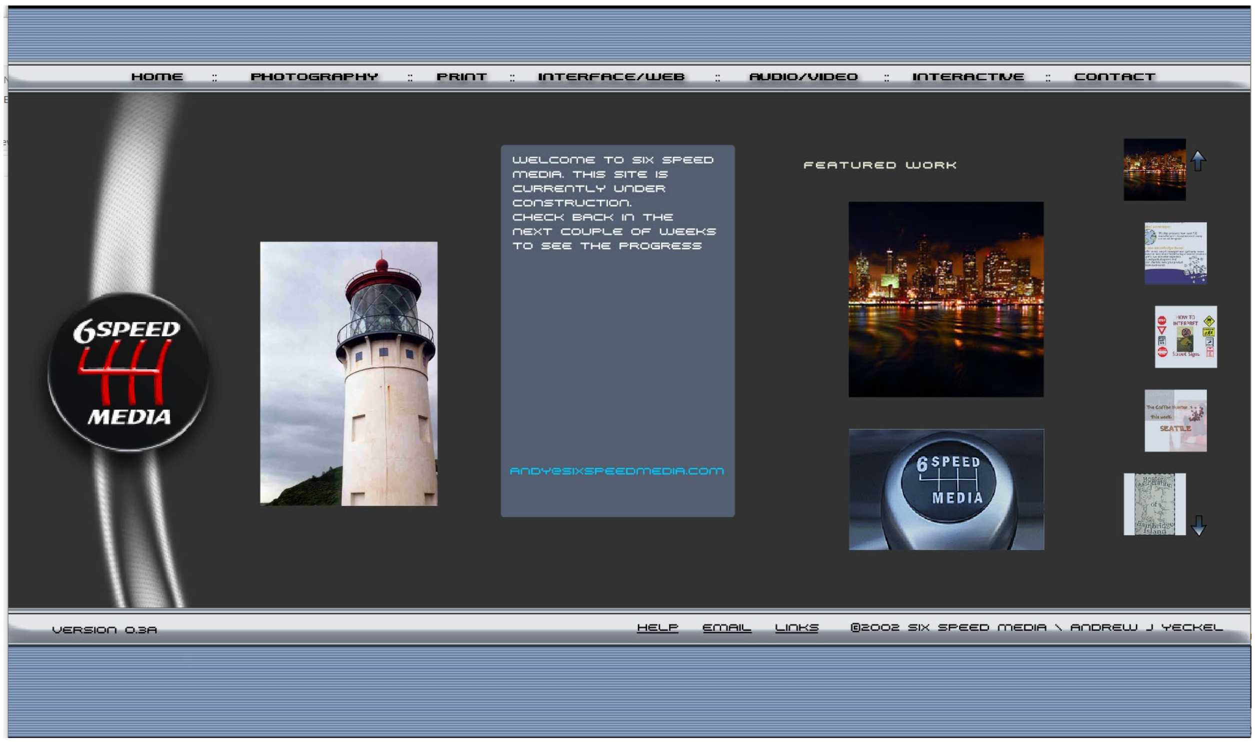

SixSpeed Media

Sixspeedmedia.com .1 HTML Version

Monday, November 18, 2002, 4:48:49 PM



Sixspeedmedia.com v.03 Flash version

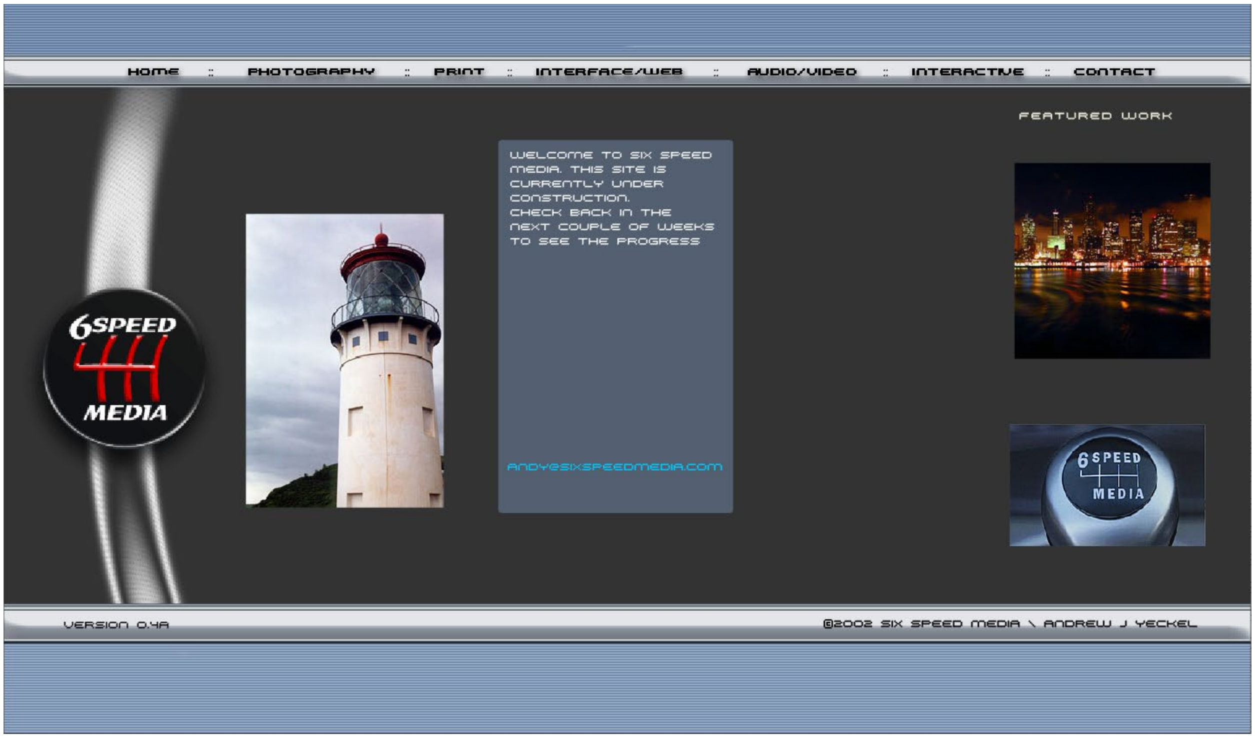


First functional "home page" template

Live 2002.11.26 16:24 B C:\Portfolio Files\build\index.html --> 64.127.195.77 /httpdocs index.html (WS_FTP log)

Single pixel high gradient image stretched horizontally for super low file sizes and scalability. New logo. The domain 6speedmedia.com also pointed to this website. Interactive was the section where these animated interfaces were to be showcased, first layout including icons laid on an ellipse with arrows indicating additional content

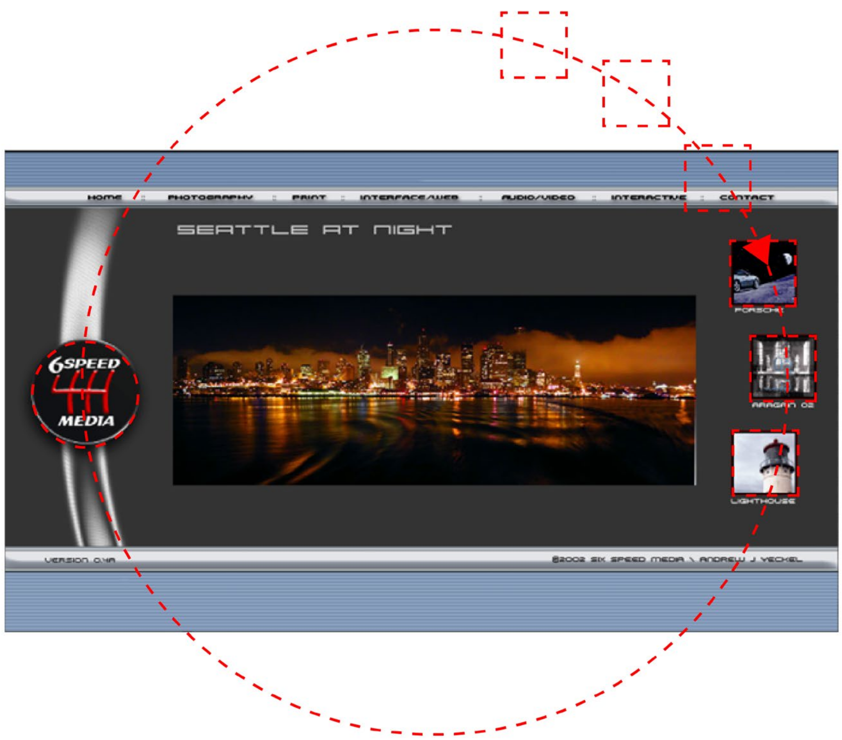
Sixspeedmedia.com v.04a (12–2–2002)
Below is a concise overview of the core features and associated files of the **Sixspeedmedia.com v0.4a** release, dated **December 2, 2002**. This build showcases several foundational design ideas and UI innovations that predate much of the later carousel and parallax work.



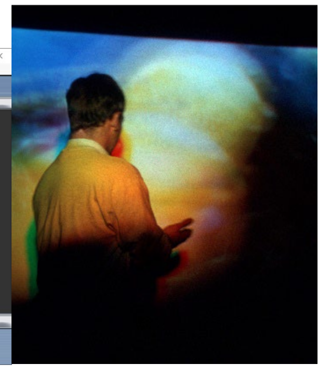
Key Features

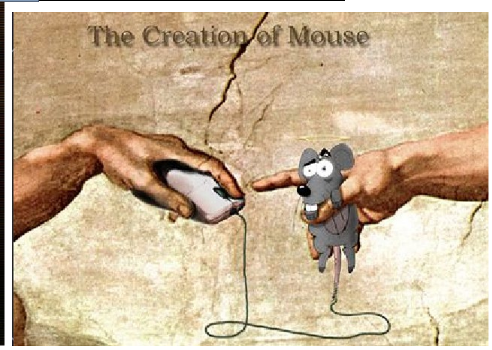
1. **Standardized Templates for Multiple Media Types**
    - Implemented uniform layout structures for documents, videos, websites, photographs, and vector-based graphics.
    - Provided consistent navigation elements (pre–carousel version) to unify user experience.
2. **Elliptical Icon Layout (Early Carousel Concept)**
    - Introduced an elliptical layout of icons and placeholders, demonstrating the first iteration of what would shortly evolve into a more advanced carousel system.
3. The icons are arranged around an invisible “center” on an elliptical path, placing a logo at that center and positioning icons toward the right. Initially, these icons animated dynamically into place based on the number of items, requiring a more flexible solution. That led to a 3D carousel, with the logo as its central reference point and the right-hand elliptical layout eventually evolving into the 2d carousel.

4.  V 0.3 ellipse carousel.



5. **Basic Data-Driven Elements**
   - Beginnings of using external sources (e.g., partial text or references from a local "database" approach) to populate certain SWFs and HTML pages.
   - Although not fully dynamic, the foundation for a more robust data-binding system was laid.
6. **Mouse–Based Motion and Animations**
   - Demonstrated prototypes of "mouse follow" and interactive transitions (e.g., **interactive.swf**, **mouse.swf**), foreshadowing future touch gestures.
   - 





7. **Integrated Media: Vector + Photo + Video**
   - Showcased combining vector art, raster images, and limited video clips (e.g., **trailer.swf**) to demonstrate the power of a single user experience unifying multiple media types.
8. **Design & Branding Elements**
   - Branding references to "© 2002 Andrew J. Yeckel" and "Sixspeedmedia.com" on main screens and associated SWFs for consistent ownership markers.
   - Introduced placeholders for potential client "theming" (e.g., boat shows, automotive shoots, city–scape presentations).

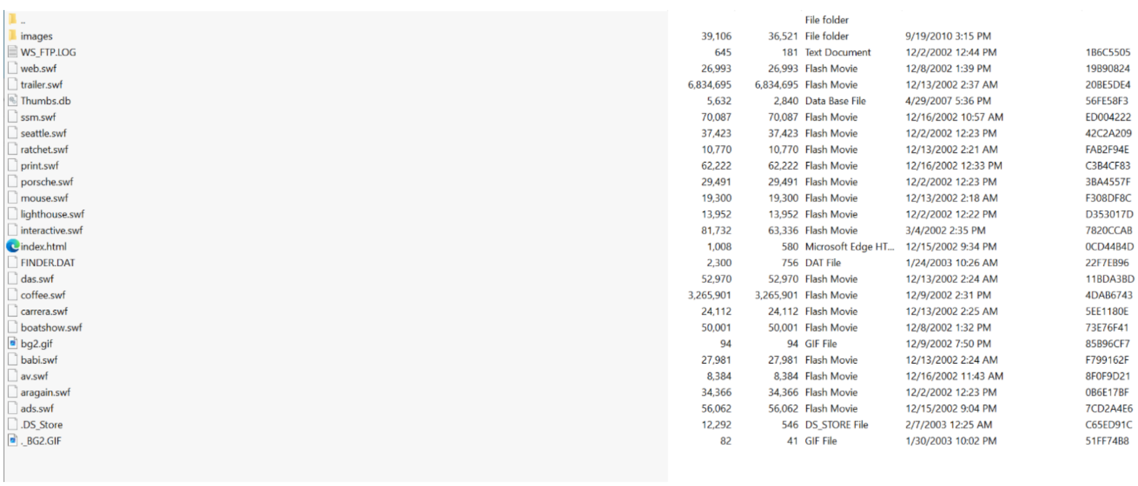
File Metadata and Descriptions

File Name	File Type	Size (bytes)	Last Modified	Description
images	Folder	39,106	9/19/2010 3:15 PM	Contains referenced images for the SWF files

File Name	File Type	Size (bytes)	Last Modified	Description
WS_FTP.LOG	Text Document	645	12/2/2002 12:44 PM	Log file indicating files uploaded on 12/2/02
web.swf	Flash Movie	26,993	12/8/2002 1:39 PM	Demonstrates examples of website designs
trailer.swf	Flash Movie	6,834,695	12/13/2002 2:37 AM	Video clip prototype, accessed from a static 2-item carousel
Thumbs.db	Data Base File	5,632	4/29/2007 5:36 PM	Windows thumbnail cache (auto-generated by Windows)
ssm.swf	Flash Movie	70,087	12/16/2002 10:57 AM	Labeled “Version 0.4a © 2002 Andrew J Yeckel.” Main SWF with placeholder homepage; includes logo on an ellipse & horizontally
seattle.swf	Flash Movie	37,423	12/2/2002 12:23 PM	Template showcasing a photograph of Seattle (taken from the ferry to Bainbridge Island)
ratchet.swf	Flash Movie	10,770	12/13/2002 2:21 AM	Project template featuring vector art of a ratchet
print.swf	Flash Movie	62,222	12/16/2002 2:21 AM	Category-page template with elliptical menu (3 items on right); precursor to an animated menu
porsche.swf	Flash Movie	29,491	12/13/2002 2:18 AM	Photo/design project template featuring a Porsche “on the moon” (Photoshop)
mouse.swf	Flash Movie	19,300	12/2/2002 12:22 PM	Illustration / prototype for moving from mouse to touch manipulation; comedic “Michelangelo Creation of Man” parody
lighthouse.swf	Flash Movie	81,732	3/4/2002 2:35 PM	Photo template involving a lighthouse scene
interactive.swf	Flash Movie	1,008	12/15/2002 9:34 PM	Early button-interaction prototype (animations & sounds for “sign/button” concept)
index.html	HTML / Edge Doc	580	12/15/2002 9:04 PM	Basic HTML wrapper embedding the Flash files; minimal styling
FINDER.DAT	DAT File	2,300	1/24/2003 10:26 AM	Likely leftover from Mac file system (Finder data)
das.swf	Flash Movie	52,970	12/2/2002 2:24 AM	Final for a web design class; attempts early skeuomorphic design
coffee.swf	Flash Movie	3,265,901	12/9/2002 2:31 PM	Short video project for a video-production class

File Name	File Type	Size (bytes)	Last Modified	Description
carrera.swf	Flash Movie	24,112	12/13/2002 2:25 AM	Another photo/design project featuring a car (Carrera), presumably used as a template
boatshow.swf	Flash Movie	50,001	12/8/2002 1:32 PM	Advertisement for go2marine.com boat show; uses site DB data to populate a product template (made in Illustrator)
bg2.gif	GIF File	94	12/9/2002 7:50 PM	Repeating background tile for HTML background
babi.swf	Flash Movie	27,981	12/13/2002 2:24 AM	Project template with logo for Bainbridge Island Boater Association
av.swf	Flash Movie	8,384	12/16/2002 11:43 AM	Incomplete video page with just 2 items (no carousel needed)
aragain.swf	Flash Movie	34,366	12/2/2002 12:23 PM	Photo template for an album-cover design ("Aragain" project)
ads.swf	Flash Movie	56,062	12/15/2002 9:04 PM	Magazine print ads referencing a website DB
.DS_Store	DS_STORE File	12,292	2/7/2003 12:25 AM	macOS system file storing folder preferences
._BG2.GIF	GIF File	82	1/30/2003 10:02 PM	Another macOS resource fork for bg2.gif

---



..			File folder		
images	39,106	36,521	File folder	9/19/2010 3:15 PM	
WS_FTP.LOG	645	181	Text Document	12/2/2002 12:44 PM	1B6C5505
web.swf	26,993	26,993	Flash Movie	12/8/2002 1:39 PM	19890824
trailer.swf	6,834,695	6,834,695	Flash Movie	12/13/2002 2:37 AM	20BE5DE4
Thumbs.db	5,632	2,840	Data Base File	4/29/2007 5:36 PM	56FE58F3
ssm.swf	70,087	70,087	Flash Movie	12/16/2002 10:57 AM	ED004222
seattle.swf	37,423	37,423	Flash Movie	12/2/2002 12:23 PM	42C2A209
ratchet.swf	10,770	10,770	Flash Movie	12/13/2002 2:21 AM	FAB2F94E
print.swf	62,222	62,222	Flash Movie	12/16/2002 12:33 PM	C3B4CF83
porsche.swf	29,491	29,491	Flash Movie	12/2/2002 12:23 PM	3BA4557F
mouse.swf	19,300	19,300	Flash Movie	12/13/2002 2:18 AM	F308DF8C
lighthouse.swf	13,952	13,952	Flash Movie	12/2/2002 12:22 PM	D353017D
interactive.swf	81,732	63,336	Flash Movie	3/4/2002 2:35 PM	7820CCAB
index.html	1,008	580	Microsoft Edge HT...	12/15/2002 9:34 PM	0CD44B4D
FINDER.DAT	2,300	756	DAT File	1/24/2003 10:26 AM	22F7EB96
das.swf	52,970	52,970	Flash Movie	12/13/2002 2:24 AM	11BDA3BD
coffee.swf	3,265,901	3,265,901	Flash Movie	12/9/2002 2:31 PM	4DAB6743
carrera.swf	24,112	24,112	Flash Movie	12/13/2002 2:25 AM	5EE1180E
boatshow.swf	50,001	50,001	Flash Movie	12/8/2002 1:32 PM	73E76F41
bg2.gif	94	94	GIF File	12/9/2002 7:50 PM	85B96CF7
babi.swf	27,981	27,981	Flash Movie	12/13/2002 2:24 AM	F799162F
av.swf	8,384	8,384	Flash Movie	12/16/2002 11:43 AM	8F0F9D21
aragain.swf	34,366	34,366	Flash Movie	12/2/2002 12:23 PM	0B6E17BF
ads.swf	56,062	56,062	Flash Movie	12/15/2002 9:04 PM	7CD2A4E6
.DS_Store	12,292	546	DS_STORE File	2/7/2003 12:25 AM	C65ED91C
._BG2.GIF	82	41	GIF File	1/30/2003 10:02 PM	51FF74B8

Mockup of .5 (not listed) Modified Friday, November 29, 2002

Sixspeedmedia.com Version .71

Sixspeedmedia.com – Version 0.7 (12-16-2002)
Summary of Files & Features

1. **index.html**
   - **Purpose/Notes**: Carries over from v.04a as the primary HTML container embedding the Flash content.
   - **Key Feature**: Basic HTML wrapper/embedding code for the updated "ssm.swf."
2. **ssm.fla**
   - **Purpose/Notes**: Flash source file introducing a **scalable UI animation**.
   - **Key Feature**: Uses overlapping graphical elements & "single-pixel" stretch images to allow infinite scaling with gradients/raster effects. Demonstrates flexible layout based on screen size.

3. **ssm.swf**
   - **Purpose/Notes**: Compiled output from "ssm.fla."
   - **Key Feature**: Showcases the new "scalable UI intro," visually demonstrating how UI elements adapt to various screen dimensions.
4. **template.fla**
   - **Purpose/Notes**: Flash source used as a **data-binding** template (precursor to fully dynamic content).
   - **Key Feature**: Lays groundwork for pulling external data or changing text/images in a re-skinnable architecture.
5. **web.swf**
   - **Purpose/Notes**: Placeholder or partial "web" UI demonstration.
   - **Key Feature**: Possibly a minimal concept or skeleton for linking multiple site sections.
6. **seattle.fla / seattle.swf**
   - **Purpose/Notes**: Reused or slightly updated from earlier versions (v.04a).
   - **Key Feature**: Contains a photographic backdrop of Seattle (ferry vantage), used as a thematic site section.
7. **print (Fla & Swf)**
   - **Purpose/Notes**: "Print" project showcasing **elliptical menu** + a main content panel.
   - **Key Feature**: Expands the elliptical navigation concept introduced in prior versions; navigable sub-categories.
8. **ssm.html**
   - **Purpose/Notes**: Default Flash output HTML for "ssm.swf."
   - **Key Feature**: Typically auto-generated by Flash, allowing quick local testing or direct publication.
9. **ssm Report**
   - **Purpose/Notes**: A text or partial "Movie Report" exported from Flash. (Frame data, shape bytes, embedded objects, etc.)

    - **Key Feature**: Confirms the internal breakdown of frames, embedded graphics, images, and fonts.

10. **center.png / right.png** (if "right" is a PNG)
    - **Purpose/Notes**: **Scalable UI** elements that align or move to illustrate dynamic layout.
    - **Key Feature**: The "center" portion is a key anchor in the layout; "right" portion slides or animates for partial expansions.
11. **av.swf**
    - **Purpose/Notes**: Placeholder page for video or A/V content.
    - **Key Feature**: Could be a minimal example of linking embedded video or external media in the final site structure.

High-Level Feature Notes (v.07)

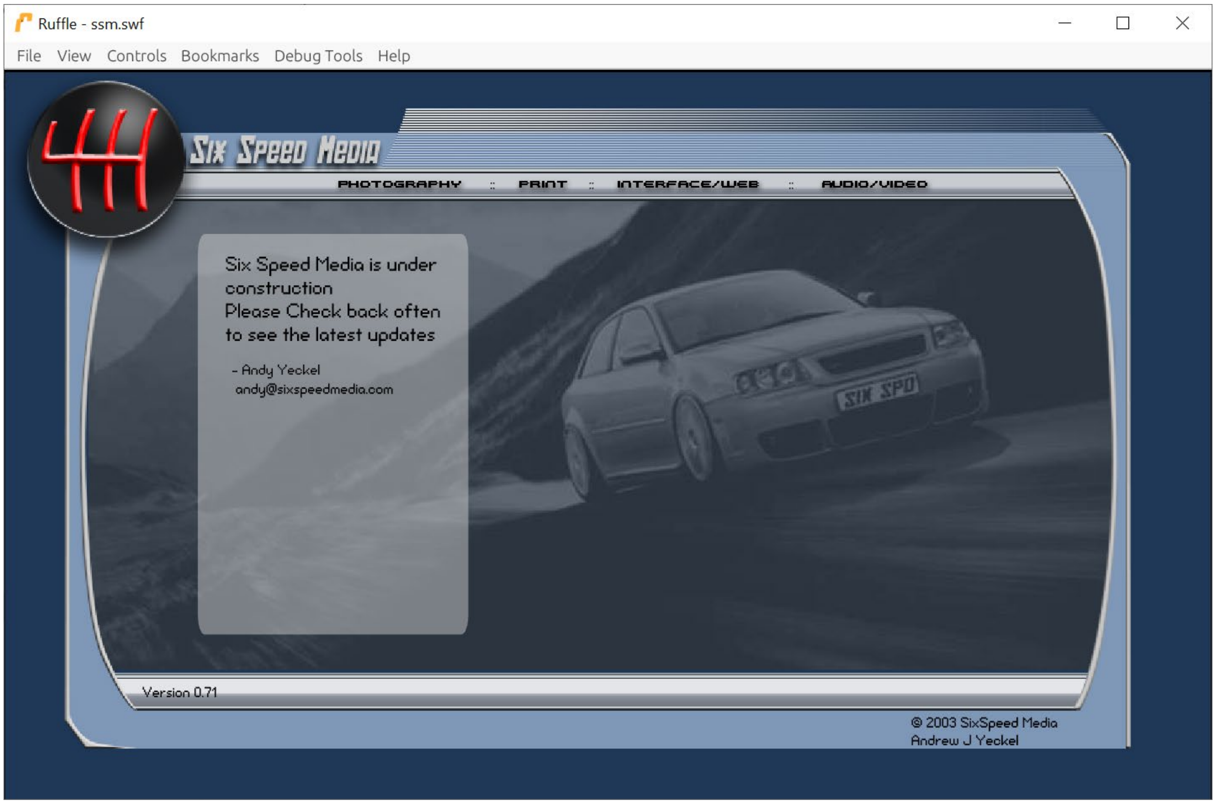
- **Scalable UI / Intro Animation**
  Demonstrates use of single-pixel stretch images to create infinitely scalable backgrounds or bars, plus overlapping layers for fluid resizing.
- **Elliptical Navigation & Content Panel**
  "print.swf" highlights a more refined elliptical menu (first introduced in earlier versions), forming a ring of icons or items around a central content area.
- **Template for Data Binding**
  "template.fla" hints at hooking external data to dynamically populate text or images—a stepping stone for a fully modular "carousel" or "slide" system.
- **Carry-Forward Visuals**
  Selections (like "seattle.swf," "print.swf," "web.swf") confirm incremental improvement from earlier versions (v.04a) rather than a complete rebuild.
- **Reports & Internal Summaries**
  "ssm Report" and any "Movie Report" text confirm detailed breakdowns of the SWF structure, used for debugging or verifying frame usage.

Table showing **only the new or substantially changed files** in *Sixspeedmedia.com v.07* versus the previous version (v.04). Each entry highlights the approximate date (if known) and what changed or was newly introduced.

File Name	Approx. Date	Description / Changes from v.04
ssm.fla	2/11/2003	*New Flash source* focusing on a **scalable UI animation**. Demonstrates overlapping layers, single-pixel image "stretch" techniques, and dynamic resizing that was *not* present in v.04.
ssm.swf	2/11/2003	Compiled output of `ssm.fla`. Showcases the **intro animation** highlighting flexible screen-size adaptation.
template.fla	2/11/2003	*Introduced in v.07.* A **data-binding**-ready template, paving the way for external text/image updates (modular content loading).
ssm.html	2/11/2003	Auto-generated HTML wrapper for `ssm.swf`. Different from prior "index.html" in v.04, used primarily for local testing or direct publishing.
ssm Report	2/11/2003	A "Movie Report" detailing the SWF's internal frames/bytes/objects. *New in v.07* to track resource usage and confirm refined animation assets.
center.png / right.png	2/11/2003	*New UI images* integral to the flexible/scalable layout. "center" is the anchor; "right" slides or resizes. Not used in v.04's simpler layout.
av.swf	12/13/2002	Placeholder for video/A-V content. Distinct from older "multimedia" placeholders in v.04, indicating expanded focus on integrated video sections.

Note:

- Dates shown are typical "Last Modified" times from the directory listing (some round to 2/11/2003).
- Files like `index.html`, `web.swf`, `seattle.fla`, or `print.swf` are *largely carried over* from v.04 with only minor edits.
- These new/changed files emphasize *scalable UI concepts*, refined elliptical navigation, and early steps toward data-driven templates.



Shows early scalable ui element concepts. The ui scales open to show how the UI works at different widths.

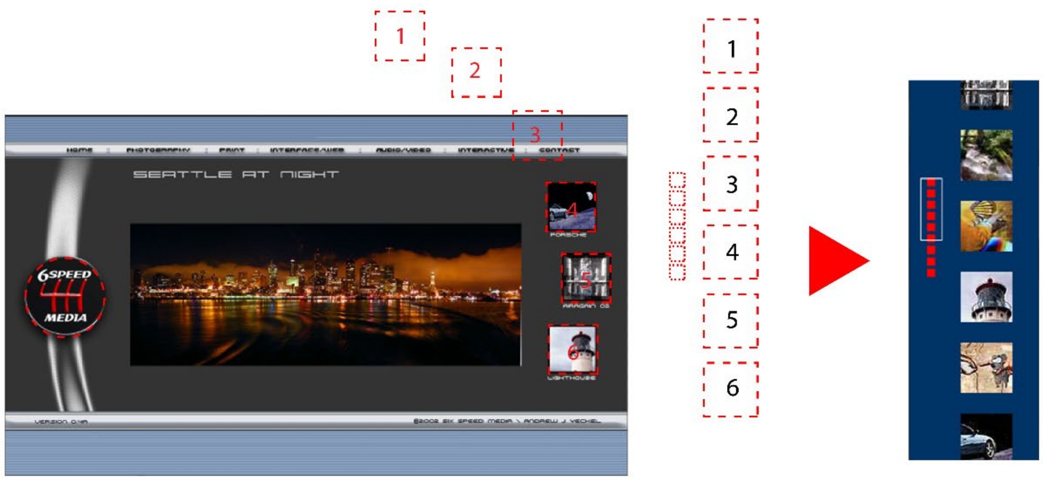
Version 0.8 Summary (March 2003)

- **Objective and Scope**
  Version 0.8 introduced multiple **2D carousel** variants, each employing **inertia-based scrolling**, a **parallax effect**, and a "crop area" to accommodate off-screen items. Although data binding remained manual, the structural foundation for a modular, data-driven interface was established. This version also continued using distinct SWF files for varied content categories (e.g., print layouts versus photographic galleries), demonstrating how a single carousel framework could be reskinned or adapted.
- **Key Files and Features**
  - **`homecontent.swf`**: A 3D carousel intended specifically for home-page use. It can be swapped in or out without changing the main application shell.
  - **`printnav.swf`** / **`webnav.swf`**: Skins that adapt the core 2D carousel for print materials or conventional websites, both relying on the same elliptical navigation logic.
  - **`nav.swf`**: A refined carousel prototype featuring velocity-driven "flick" gestures, parallax layering, and continued motion when thumbnails are "thrown."
  - **`gallery.swf`**: Describes the JavaScript-based approach (originally designed for the ArtsCorps image gallery) to show how external data could be used to automatically generate the gallery based on filenames.

- **Duplication and Placeholder Items**: Files such as `Copy of porsche.swf` or `Copy of aragain.swf` served as backups during color/transition testing. In addition, some placeholders like `dreamage.swf` were carried over from earlier builds for color or minor layout changes.

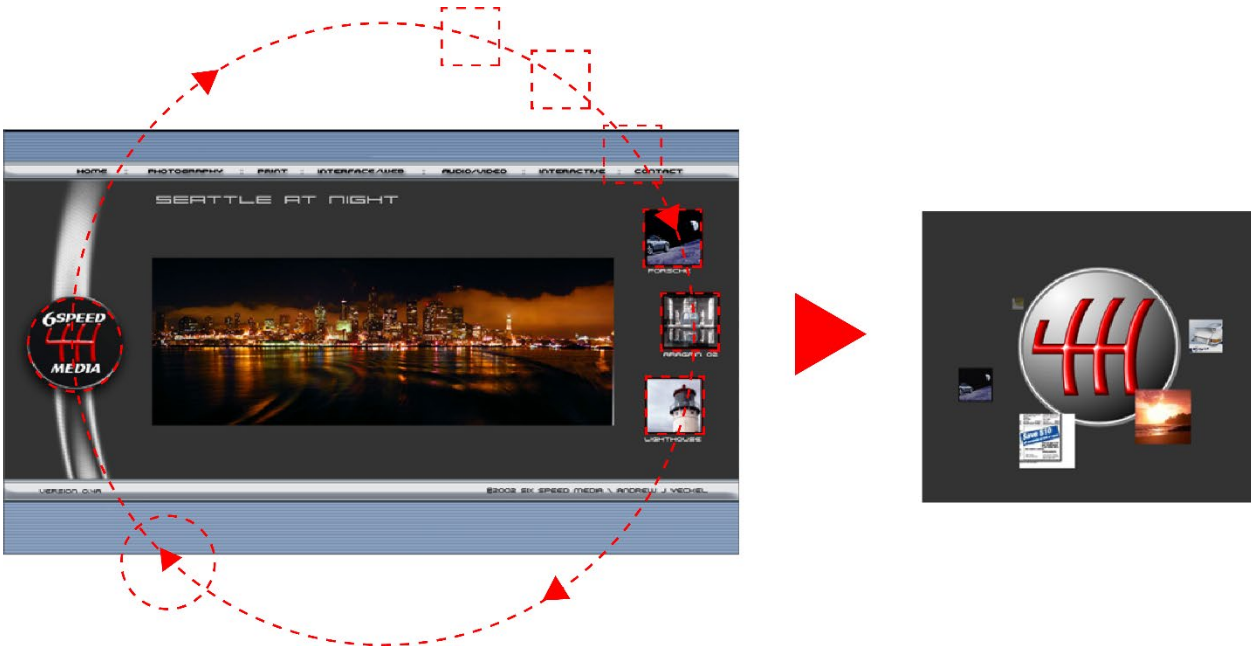
- **Comparison to Earlier Builds**
  Whereas Version 0.7 and prior releases focused on proof-of-concept for elliptical menus and 3D/2D transformations, Version 0.8 consolidated these ideas into a more cohesive system. The emphasis on **inertia-scrolling** and **skinnable carousels** demonstrated how different visual requirements (e.g., photo galleries vs. print previews) could be met by a single underlying framework. Though still lacking automated data-binding, Version 0.8 finalized the core architectural approach that would allow future dynamic or XML-driven content.

Nav.swf 3-11-2003 Shows the 2D carousel and parallax crop area as a stand-alone prototype.

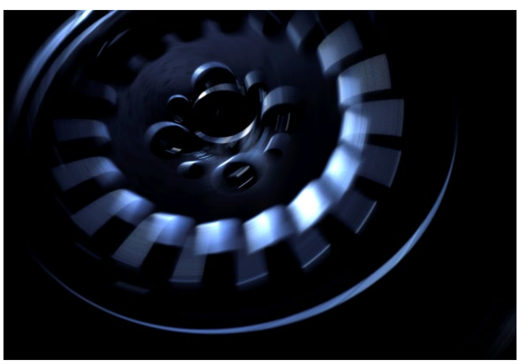


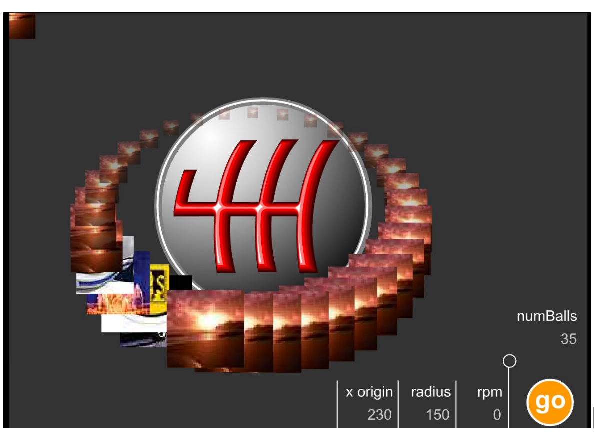
2d carousel was developed by flattening the 3d carousel concept and reusing the same motion and inertia as the 3d carousel. This had crop area to show off screen content.

homecontent.swf



Attempt at animating icons around a circle led to adopting existing ellipse math to create a carousel that moved. The background at the time was a BMW Bottlecap wheel which inspired the tilt. Ellipse as a brand element.

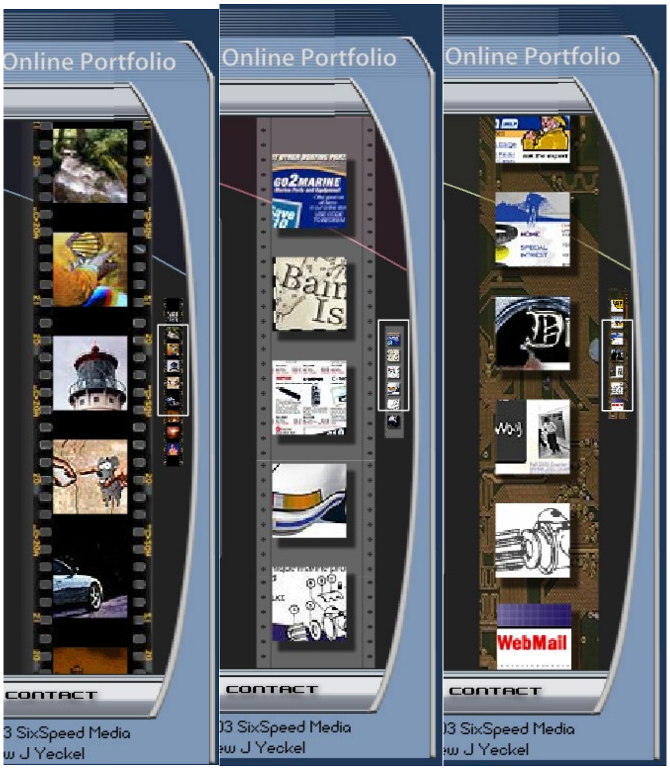


Modified ellipse Prototype with mouse control, and icons/links. The math allows for the number of items to change, here shows 35.

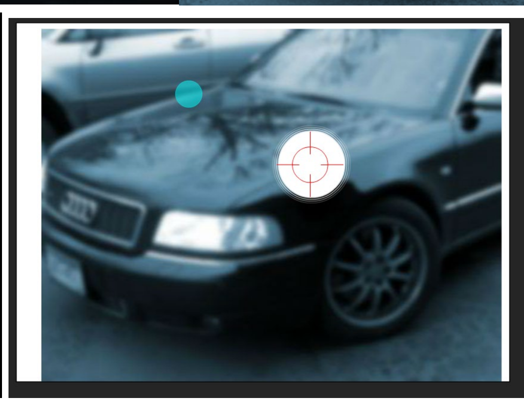
File Name	Approx. Date & Size	Notes / Description
homecontent.swf	3/20/2003 12:17 AM (17 KB)	Shows a 3D carousel UI embedded on the homepage. "homecontent" can be swapped to update just the landing content.
babi.swf	3/20/2003 1:06 PM (18 KB)	A relocation/re-export of the earlier "babi" file (e.g. brand or small site).
Copy of porsche.swf	3/19/2003 11:19 PM (23 KB)	Backup of the original "porsche" SWF, containing the RGB image transition.
webnav.swf	3/19/2003 11:04 PM (34 KB)	A 2D carousel ("nav"-style) themed for website integration, with elliptical or horizontal transitions.
iq.swf	3/13/2003 ~1:37 AM (29 KB)	Conceptual image of an interactive-wall prototype, showcasing a Photoshop mockup for physical or gesture control.
go2marine.swf	3/13/2003 ~1:36 AM (38 KB)	A placeholder "RGB project template" referencing Go2Marine content (possibly parallax or multi-item "boatshow").
dreamage.swf	3/13/2003 ~1:35 AM (39 KB)	Another placeholder or conceptual parallax demo, akin to go2marine.
gallery.swf	3/13/2003 ~1:33 AM (37 KB)	A "gallery view," presumably image-based 2D carousel or scroller, following earlier "photonav."
printnav.swf	3/11/2003 10:42 PM (11 KB)	2D carousel "skinned" for printed documents; evolved from older "print.swf" with added navigation for multiple items.
photonav.swf	3/11/2003 10:25 PM (21 KB)	Another 2D carousel with different item counts to demonstrate scaling for variable content.
chatterbox.swf	3/11/2003 7:57 PM (18 KB)	Displays a photo of "Chatterbox Falls"; primarily an image placeholder.
nav.swf	3/11/2003 4:29 PM (72 KB)	A refined 2D carousel prototype implementing inertia (thumbnails keep moving after flick/drag). Red squares are placeholders.

Skinnable



Blur Test Prototype (Jan-Feb 2003)



Purpose

- Demonstrate blur effects in a real-time interactive context.
- Experiment with layering techniques in Flash to produce a “soft-focus” or “defocused” region while the user interacted with the scene.

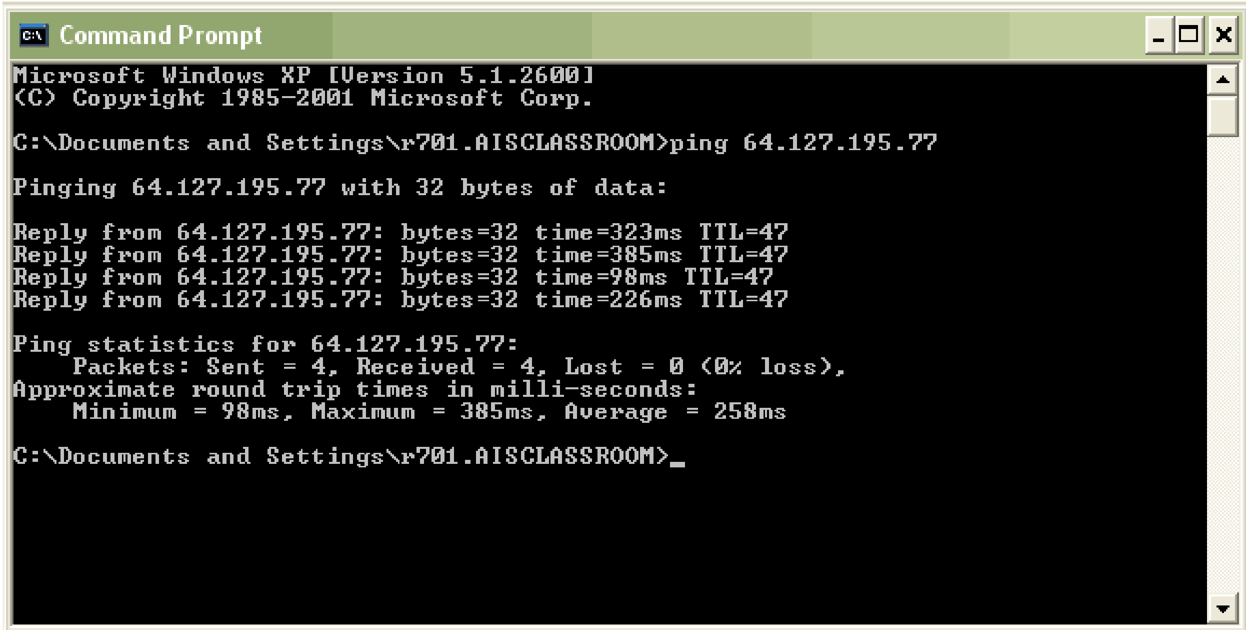
Implementation

- Employed a second, pre-blurred layer of the main graphic, alpha-masked so that only part of the image appeared blurred at any time.
- Used custom ActionScript to track user input (e.g., mouse movement or simple “aim” actions) and reveal the blurred layer in localized areas—creating the illusion of a selectively blurred “scope” or “reticle” region.
- Integrated simple sound effects (click or shot sound) to confirm user interactions and highlight the technical viability of multi-sensory feedback.

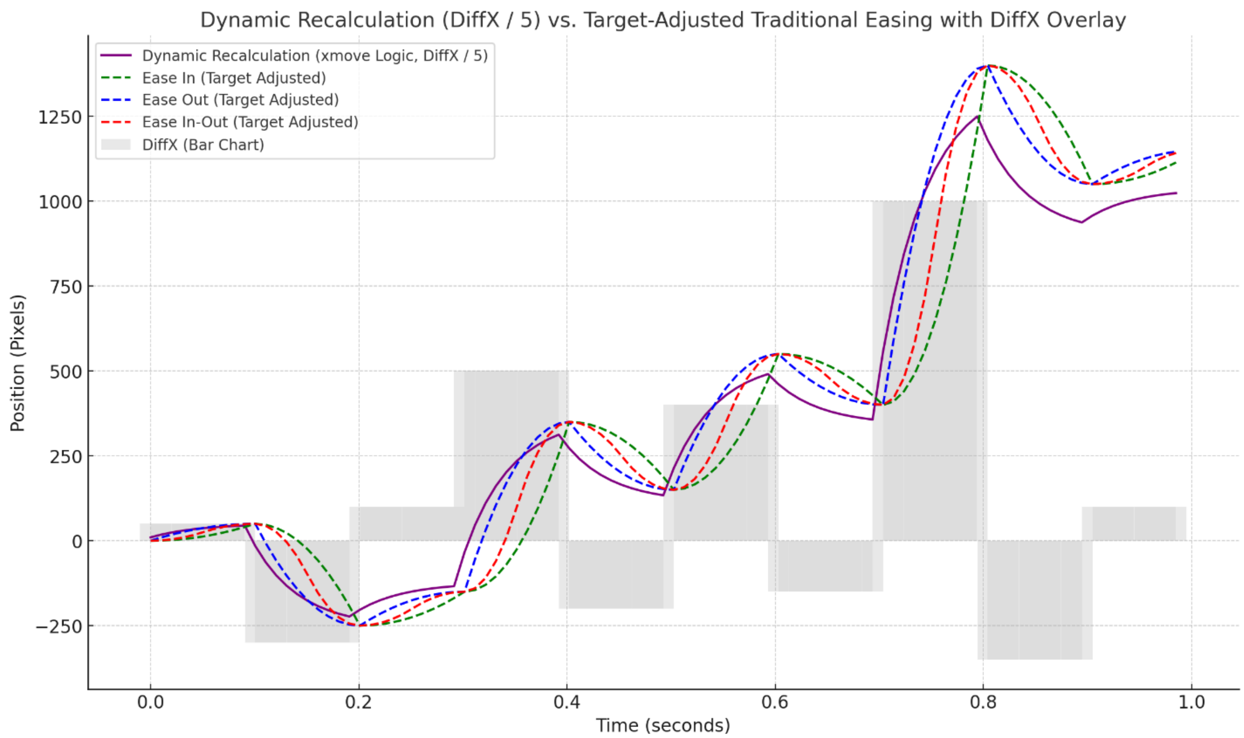
Key Takeaways

- Showed that real-time blur in a Flash-based interface could be done without external plugins or heavy performance hits (for that era).
- Proved that dynamic visual effects (such as animated transitions from sharp to blurred states) could respond to user-driven events—laying groundwork for later UI animations and transitional concepts found in the 2003–2005 timeframe.

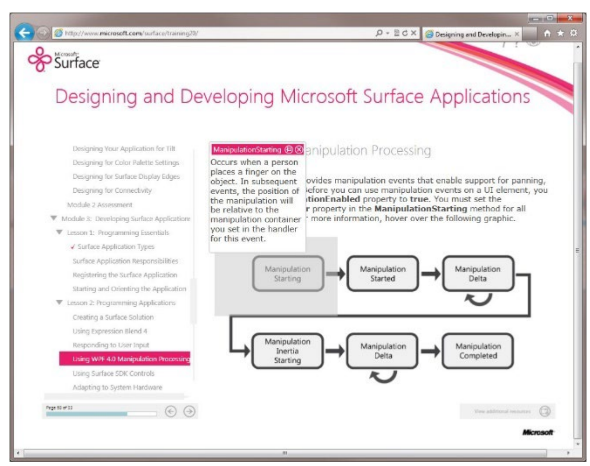
Although the blur test was a relatively small-scale demo, it exemplified the early drive toward interactive, visually rich interfaces that reacted fluidly to user input—well before such techniques became standard practice in modern frameworks.



Screenshot from the first Ping test of the live sixspeedmedia.com web server showing public accessibility, the time and which room it was accessed from (Room 701 at the Art Institute of Seattle, Wednesday, January 15, 2003, 10:23:48 AM) IP Matches WS_FTP Logs showing files were uploaded.



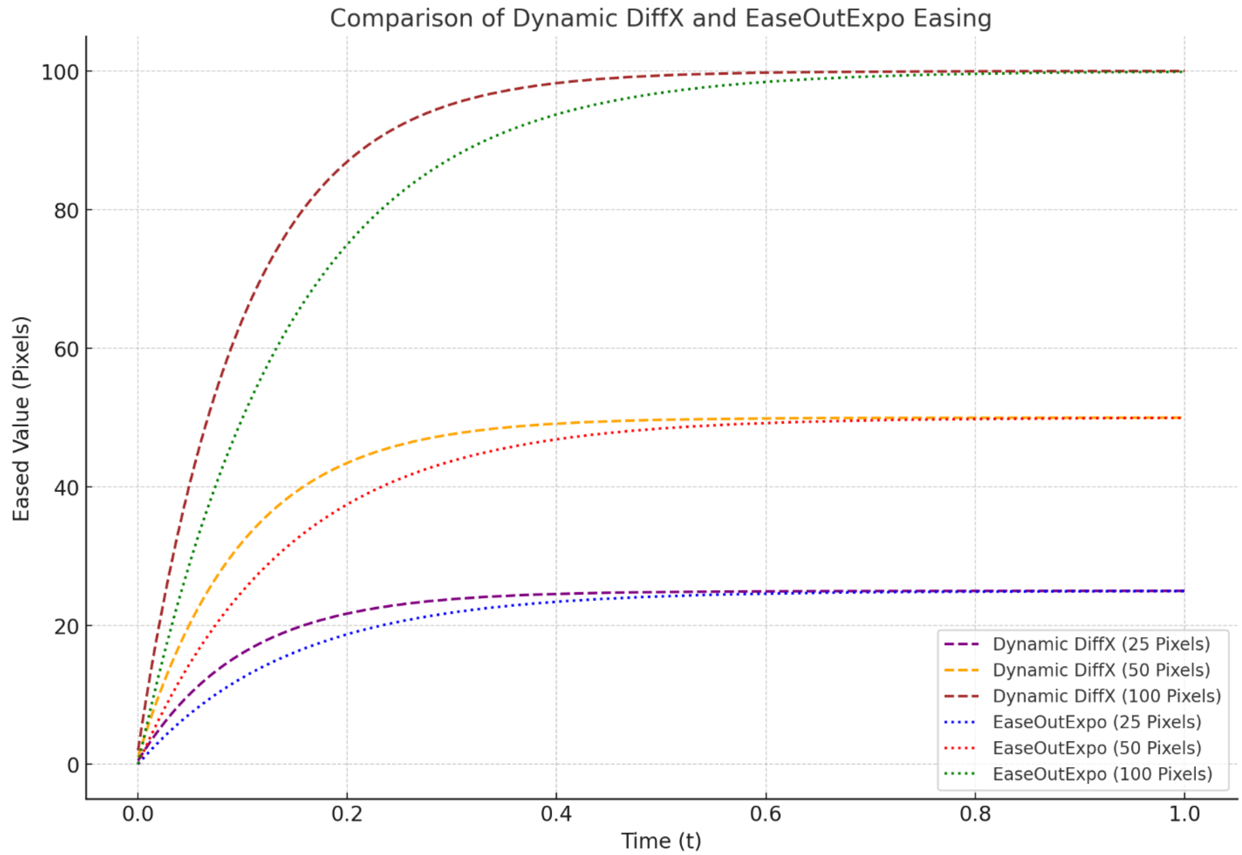
This graph shows the responsiveness of the manipulation using the older DiffX method. It is faster and smoother than basic easing equations, simulating physics without complex and fixed calculations. The gray bar shows the distance the point is moved and the response based on different methods. It is more limited in how it can be programmed later, though changing the ratio causes responsiveness and snappiness to decrease/increase. The purple line responds faster and limits overshoots as the math is constantly recalculated.



Touch screens mostly use a 1:1 (drag) position when the object is being manipulated and then switches to easing over time once the delta is calculated.

This graphic shows the current methods where the first position is calculated, the delta(difference) between start and finish and then calculating inertia based on time and whatever easing or physics calculations are chosen (easing or real physics) then a second delta transform, this one controlled by the inertia calculations and then it calls complete.

I have drag (grab) code demonstrated in 2001 as part of a demo for mouse. The concept is essentially identical 1:1 movement upon a grab event.



This shows the difference in actual values vs more modern EaseOutExpo like used in Manipulation processing is minimal when the end point is stationary. Essentially performing identical or snappier than more modern and powerful methods. The earlier photonav prototype has more visible inertia. This was to increase responsiveness of the ui while still having some life.

There are a number of easing choices in modern system including ones that have bounce as an option.

Ellipse UI Code

```
/////////////////////////////////////////////////////////////////////////////////
//////////////////////////////

////// NOTE: This is COMMENTED FOR CLARITY, ORIGINAL CODE AND LOGIC RETAINED and
archived in original document /////

/////////////////////////////////////////////////////////////////////////////////
//////////////////////////////



// disable yellow box around buttons accessed via the keyboard
_focusrect = 0;

// Clip scale at smallest part of ellipse in %
var minScaleInput = 50;
// Clip scale at Largest part of ellipse in %
var maxScaleInput = 250;

//Initalize rotation position in degrees
var rotationinput = 30;

//Set the ellipse -1/2 the height
var ellipseHeightInput = -.5;

// Set width to 100%
var ellipseWidthInput = 1;

// Center reference point Y position
var centerYinput = 250;
// Center reference point X position
var centerXinput = 500;

// Set the size of the ellipse in pixels
var radiusInput = 300;

// Set the initial rotation speed
var speed = .2;

// Number of Clips to create
var numClipsInput = 4;
```

```
// Initialize movement speed and start time.
function init() {
    // First, clear the stage of any previous Clips.
    removeClips();
    // Now make a new set of Clips as Int.
    makeClips(parseInt(numClipsInput));
    // Done initializing, now start the event loop to move the object
    this.attachMovie("processDoMove", "processDoMove", 10);

    //origin is center of logo clips orbit convert to float for precision
    origin._x = parseFloat(centerXInput);
    origin._y = parseFloat(centerYInput);

}

// Function to move a clip with each passing frame.
function moveClipAroundCircle(theClip) {
    // Determine how much time has passed instead of fixed to framerate for
smoother animations
    theClip.then = theClip.now;
    theClip.now = getTimer();
    var elapsed = theClip.now - theClip.then;
    var numSeconds = elapsed / 1000;

   // Calculate the speed of the animation based on the horizontal mouse position
(_xmouse).
// This determines the base value for mouseX, influenced by an external data-
driven speed variable (_root.speed).
mouseX = (_xmouse - 500) * _root.speed;

// Check for extreme conditions in the horizontal mouse position to adjust mouseX
further.
// If the mouse is near the left edge of the screen (_xmouse < 100), slow down
the animation significantly.
if (_xmouse < 100) {
    mouseX = mouseX / 20;
}
// If the mouse is far to the right (_xmouse > 1000), apply a complex adjustment
to reduce the speed dynamically.
// This uses a nested calculation based on _xmouse for diminishing returns as
rudimentary ease out.
// Mimics easing functions by reducing the rate of change as the mouse moves
farther to the right.
else if (_xmouse > 1000) {
```

```
    mouseX = mouseX / (_xmouse / (_xmouse / 3));
}

// Override previous adjustments if the mouse is near the far right edge (_xmouse
> 900),
// set idle speed
if (_xmouse > 900) {
    mouseX = 10;
}

// If the mouse is positioned near the top of the screen (_ymouse < 1),
// set idle speed based on y mouse value
if (_ymouse < 1) {
    mouseX = 10;
}

// Calculate the increment in degrees for this frame based on the adjusted mouseX
value.
// Multiply by the number of seconds elapsed to ensure smooth, time-based
animation.
var degreeIncrement = mouseX * numSeconds;

    // Update the angle
    theClip.theta += degreeIncrement;
    // Keep the angle between -360 and 360 degrees
    theClip.theta %= 360;
    // Convert the ellipse's rotation angle (rotationTheta) from degrees to
radians
    // for use with trigonometric functions (sin, cos) in the following
calculations.
    var radians = degreesToRadians(theClip.theta);
    // Figure out where the clip resides on the ellipse around which it's
travelling.
    // Adjust the ellipse tilt dynamically based on Y mouse position (commented
out).
    //*((_ymouse - 1100) / -700);
    var xOnEllipse = theClip.ellipseWidth * (Math.cos(radians) * theClip.radius);
    var yOnEllipse = theClip.ellipseHeight * (Math.sin(radians) *
theClip.radius);

    // Adjust the calculated position on the ellipse to account for the ellipse's
overall rotation.
    // The raw position (xOnEllipse, yOnEllipse) is rotated by the ellipse's
rotation angle (rotationTheta).
```

```
    // This transformation ensures that the ellipse can be tilted or oriented in
any direction, rather than remaining axis-aligned.
    var rotationRadian = degreesToRadians(theClip.rotationTheta);
    var rotatedX = xOnEllipse * Math.cos(rotationRadian) - yOnEllipse *
Math.sin(rotationRadian);
    var rotatedY = xOnEllipse * Math.sin(rotationRadian) + yOnEllipse *
Math.cos(rotationRadian);

    // Finalize the clip's position by adding the rotated X and Y offsets to the
ellipse's center point.
    // Adjust the Y position with a subtraction because Flash's coordinate system
has an inverted Y-axis.
    theClip._x = theClip.centerX + rotatedX;
    theClip._y = theClip.centerY - rotatedY;

// Simulates a 3d effect

function addDepthEffect(theClip) {
    // Calculate a value using sin(theta), which oscillates between -1 and 1.
    // Adding 1 shifts the range to [0, 2], ensuring all values are positive.
    // These positive values are used for smoothly scaling and fading the clip.
    var degreeSin = Math.sin(degreesToRadians(theClip.theta));
    // Adjust the clip's horizontal and vertical scale.
    // The size oscillates smoothly between minScale and maxScale based on
degreeSin.
    theClip._xscale = theClip.minScale + ((degreeSin + 1) / 2) *
(theClip.maxScale - theClip.minScale);
    theClip._yscale = theClip.minScale + ((degreeSin + 1) / 2) *
(theClip.maxScale - theClip.minScale);
    // Adjust the clip's transparency (alpha).
    // Transparency also oscillates between minScale and maxScale for a fading
effect.
    theClip._alpha = (theClip.minScale + ((degreeSin + 1) / 2) *
(theClip.maxScale - theClip.minScale));
}

// Makes a series of Clips
function makeClips(numClips) {
    // Create an array to store references to all the Clips created.
    ClipList = new Array();

    // Loop through the number of Clips to create and configure each one.
    for (var i = 0; i < numClips; i++) {
        // Attach a new Clip to the stage. The instance name is dynamically
generated as "Clip" + i.
```

```
        // The same movie is attached twice, likely an oversight or placeholder
for further logic.
        this.attachMovie("Clip", "Clip" + i, i);
        this.attachMovie("Clip" + i, "Clip" + i, i);

        // Store a reference to the newly attached Clip for easy access.
        var thisClip = this["Clip" + i];

        // Define properties for the Clip based on user input and default
programmatic values.
        thisClip.radius = parseFloat(radiusInput);         // Radius of the
Clip's motion
        thisClip.centerX = parseFloat(centerXinput);       // Center X position
of the ellipse
        thisClip.centerY = parseFloat(centerYinput);       // Center Y position
of the ellipse
        thisClip.ellipseWidth = parseFloat(ellipseWidthInput);   // Horizontal
scaling of the ellipse
        thisClip.ellipseHeight = parseFloat(ellipseHeightInput); // Vertical
scaling of the ellipse
        thisClip.rotationTheta = parseFloat(rotationInput);      // Initial
rotation angle of the ellipse
        thisClip.maxScale = parseFloat(maxScaleInput);           // Maximum size
of the Clip
        thisClip.minScale = parseFloat(minScaleInput);           // Minimum size
of the Clip

        // Calculate the initial angle (theta) for this Clip, spacing them evenly
around the ellipse.
        thisClip.theta = 0 + (i * (360 / numClips));

        // Record the current time for motion calculations.
        thisClip.now = getTimer();

        // Temporarily position the Clip off-stage to hide it until properly
animated.
        thisClip._x = -1000;
        thisClip._y = -1000;

        // Add the configured Clip to the ClipList array for group management.
        ClipList[i] = thisClip;
    }
}

// Cleans the stage of all Clip clips.
```

```
function removeClips() {
    for (var i = 0; i < ClipList.length; i++) {
        ClipList[i].removeMovieClip();
    }
}
// Updates the position and appearance of all Clips in ClipList.
// Each Clip is moved along the ellipse and its size/transparency is adjusted for
a depth effect.
function moveClips() {
    for (var i = 0; i < ClipList.length; i++) {
        moveClipAroundCircle(ClipList[i]); // Calculate and update the Clip's new
position on the ellipse.
        addDepthEffect(ClipList[i]);      // Adjust the Clip's size and
transparency for a 3D effect.
    }
}
// Converts degrees to radians.
function degreesToRadians(degrees) {
    return (degrees / 180) * Math.PI;
}
stop();
```

2D Carousel Code

```
var startscale = 250;
var endscale = 8;
var distance = 350;
var centerpoint = 500;
var numSlides = 6; // Number of slides
var slideList = []; // Array to hold slide references

// Initialize slideList with slide references
for (var i = 1; i <= numSlides; i++) {
    slideList.push(this["move" + i]);
}

function xmove() {
    homeX = (-_root._xmouse * 1.3) + 300;
    thisX = slideList[0]._x;
    diffX = homeX - thisX;
    moveX = diffX / 50;

    for (var i = 0; i < slideList.length; i++) {
        slideList[i]._x = thisX + moveX + distance * i;
```

```
    }

    _root.centerpoint = _xmouse;
    textbox = textval;
}

setInterval(xmove, 10);
```

May 2002 Website Project Goals

Andy Yeckel

Interface Design

Wednesday eve

Portfolio Architecture Plan

Site Goals

The primary goal of the site is to show the work that is in my portfolio in a simple and easy to use interface as well as define what style of art I produce.

The long-term goal(s) is to get a job.

The site will be easy to use and navigate, be compatible on all of the major browsers. All of the primary content sections will be accessible through a single menu structure that is on every page. The site will contain all of my major pieces for people to see. Speed and ease of use are important as well as standing out without being annoying.

Audience

The audience will consist on a number of people. First and foremost is the ideal employer. Second is the portfolio review teachers and students. Third is the person who is just surfing the internet for ideas and someone to help them with their own work

The first target is a 30-40 year old male who is a manager or owns his own company. He is married and has kids, Enjoys sitcoms and Star Trek. Likes to participate in the more youth oriented events or at least think young. Likes power tools. He is humorous about life but serious about work. He has ideas about design but doesn’t have the skill to refine and produce them. He is looking for someone to work with him in building a medium sized business.

Second are the people at portfolio review who are there because they can miss class. They don’t usually want to waste their time with the multimedia people because culinary has food. They want to look at a piece and be able to find what they want very quickly. They are average of 27 years old and have a pet. They are looking for ideas to use in their portfolio. The majority of the people will give it about 30 seconds of their time, if I’m lucky. To please them it needs to be quick and easy to use as well as stand out.

Third is the person who is looking through a series of online sites for ideas. They already have a job working for a large company and there is an upcoming website project. Their boss said that they need to either find someone or do the work themselves. They have great programming skills but are lacking on design skills. They will need to find someone who can do design and artwork but who understands the programming of the internet.

Theme

The primary theme will be a light industrial. There will be a combination of water and stainless steel. The Idea is to create the feeling of creativity while retaining the productive and real-world looks. The colors of steel will represent a refined and timeless style where as the water will show that the design will be creative and dynamic. The industrial feel that the steel brings will show productivity. The mood is bright and clear but not a dream or imaginary. It will all be rooted in the real world with real life applications.

The colors that are to be used are silver (grey) and blues with red highlights.

An example of the colors can be seen at the end of the document.

Visual Design

The home page will have a scaleable menu that stretches across the top of the page. In the main content area below the menu will sit visual links to the corresponding menu items providing two ways of site navigation. All of the major categories will sit in a drop-down style menu that provides instant access to all the content categories from the top scaleable menu. A logo will sit on the upper left with the link for contact just below that the link will be repeated on the scaleable menu. When an area is selected through the menu or on the visual representations, the selected link and corresponding sub categories on the drop-down menu will remain, acting as a form of bread crumbs. When a category is selected the sub category components will also appear to the left below the logo so that navigation within a category will be quicker. Each category will have large thumbnails and a descriptive sentence. The category pages will only be there as an intermediate page in the event that someone wants just a category. The primary sub categories will have a series of thumbnails that represent the piece. The thumbnails will act as links to the content. The thumbnails will be large enough that clicking on them is not required to see the detail

Below each thumbnail will be a magnifying glass icon and a icon that has an arrow. The magnifying glass icon will show the full size image and the arrow will bring the user to a page will all of the information.

Competitive Analysis

Many of the other portfolio sites tend to get distracted from the goal of the site. It seams that people just want to show that they can use the programs, but never really look any deeper. The assumption that people will be able to navigate and understand the site is a common one. Looking deeper into the layout of the site would help all of them. It doesn’t matter how good ones work might be, if it cannot be found or is too small then the goal is lost.

One site that I feel works well is Setpixel.com(http://www.setpixel.com) it uses some of the basic principals that I wish to incorporate within my portfolio.

The following is a list of observed likes and dislikes from other portfolio sites. Only the major items in their respective categories are listed.

Dislikes:

Menu structures that change from page to page

Thumbnails that are too small or nonexistent

Full size images that don't properly display the needed visual information

Any visual toy that cannot be turned off I.E. mouse trails (if they are used for the purpose of showing the technology than they are fine)

Misleading and non intuitive organization and structure

Likes:

Simple and easy to follow interfaces

Good use of graphics

An attempt to show new and emergent technologies without overdoing it

Visual feedback of links

Use of common themes and styles

Quick access to all content

Content and structure:

Ideally the site will be database driven and will allow for expandability of the site with little reprogramming.

Functionality requirements

Consistent navigation with sub-navigation listed

Standardized project display layout with corresponding project file

Expandability without major redesign (The ability to add content as needed to stay updated with current projects database driven if needed)

User activated animation within menu structure

Some form of followed link and current location display (bread crumbs)

Access to all major content categories from a single menu

Ability to show images, sound, video and other media content

Object oriented layout (Dynamic in both size and location for differing content as well as browsers)

Scaleable to accommodate users with newer systems as well as allowing for the internet standard)

Content requirements:

Images

Video

Audio

Animation

Interactive content

Contact information form page

Resume

Information presentation styles

The home page will be a static page with a dynamic link to a random page of content. The link will be a “take me to random project” link and will randomly pick from the database of projects. There will also be a “Featured project list” that will either be chosen by me or will use a database of page views to link to the most popular pieces” Beside each piece will be a comment form that will create a comment to appear in a section near the piece or send an email to myself.

Page layout for all pieces will contain an image or representation of the piece with an image or example of the piece that the user can download or view in greater detail.

There will be a section listing the “story” behind the piece describing the creative process in further detail.

Sample site map:

Home

Photography

B&W

Thumbnails and names of black and white photos

Color

Thumbnails and names color photos

Manipulation

Thumbnails of Manipulated images

Each will contain source images that made up the final piece

Print

Brochures and Flyers

Thumbnails of brochures and flyers

Magazine Advertisements

Thumbnails of advertisements

Imagery

Computer generated

2d

Thumbnails of 2d images

3d

Thumbnails of 3d images

Animation

Thumbnails of animation pieces

Interactive

Thumbnails of interactive programs and images

Video

Traditional

Thumbnails of videos

Short video clips

Special effects

Thumbnails of computer generated video clips and effects

Clips

Audio

Music

Representational thumbnail of piece along with name and length

MP3 or Wav

Voice

Representational thumbnail of piece along with name and length

MP3 or Wav

Other

Fun projects

Pieces that were done for friends and family or for charity

Resume

Resume

Contact

Form to write to me and other contact information

Class Details for May 2002 document.

Portfolio Design MM308: Lesson Plan 4

Instructor: Carrie Grey

mailto:punk@coolgrey.com

http://www.coolgrey.com

206.726.6614

Introduction: Role

Discuss Brett Polansky's class expectations

Handouts:

Lecture 1:

http://ais.coolgrey.com

Web Site Hosting

Reference Sheet – Web hosting and domain name registration

HTML Resource Files > Code Handouts

Class Critiques - Home / Section pages

Collect HW 3: 1) Revised Sitemap / Navigation System due

2) One to two paragraph describing and defining your target audience

3) First Drafts of Home Page and Section Page due (print out)

HW 4:

Break:

Lecture 2:

Week 5:

Next week, Instructor will verify the following for mid-term grades

1)

Where are you hosting your portfolio (verify acct is set-up / print out)

2)

3)

What is your domain name (verify with "whois" / print out)

Bring compiled project CD's to class

If not previously checked off:

4)

Project Inventory (and Content Categorization if completed)

5)

Sitemap or wireframe structure

Second Round Portfolio Designs

1)

Home Page PSD’s (Tier 1)

2)

Section Page PSD’s (Tier 2)

Optional: Instructor works one on one with students

One-on-one critiques of portfolio site (work in class on designs)

Finalized Site designs due week 6, so we can start development

10-13-2003 Sixspeedmedia Mobile

Uploaded First Mobile Version of Sixpeedmedia

This basic but live WML file is strong evidence that I was actively pursuing mobile development and applications as part of my broader work. It demonstrates clear intent to design for mobile platforms at a time when mobile optimization was not mainstream. It did not feature a carousel.

(IViD logo, showing water simulations)

IViD loosely connects to the Sixspeedmedia carousels through their shared focus on creating dynamic, interactive systems. While it was a rough prototype, IViD explored motion tracking and physics-based inputs, the carousels focused on modular, scalable UI elements, such as skinnable templates and inertia-based scrolling. Together, they represent a progression toward integrating physical interactions with adaptable digital interfaces.

When combined with early mobile development efforts, these projects demonstrate significant overlap with modern design and patents. The motion tracking and physics concepts from IViD, paired with the modular and touch-friendly designs of Sixspeedmedia, align closely with features seen in multi-touch systems, gesture-based interfaces, and modular UIs. These innovations clearly predate and align with many foundational elements of modern interaction design.

Original Artwork 12-15-2002 9:16 A.M.