

GAME ENGINES ON THE WEB: PAST, PRESENT & FUTURE

LITERATURE STUDY

LEONIDAS DIAMANTIS
SUPERVISED BY PROF. ANTON ELIENS

Approach

- Many aspects of an extremely wide topic, many challenges to research upon
- Main focus on technological building blocks per “era”
- Reflection on the gaming experience and implications about the community, whenever possible

Questions

- How did the development of Web-based games evolve throughout the years?
- How much was the gaming experience affected by this evolution?
- What can we expect from Web games in the future?

THE PAST

Text-Based Adventures



Text-Based Adventures

- Non-existent graphics
- Poor interaction & game control
- Rich back story

THE GRAPHICS ENGINE OF THAT TIME
WAS WITHOUT A DOUBT THE
IMAGINATION OF THE PLAYER

Text-Based Adventures

- Minimal front-end: plain HTML with trivial client-side scripting
- Rich back-end coded in solid server-side languages (e.g PHP, JSP), where all the logic happens
- Game control limited to the absolutely necessary: keyboard for typing, mouse for clicking

Text-Based Adventures



Java Applets & Embedded Code



Java Applets & Embedded Code

- Embedded and launched within a Web Page
- Executed in the Java Virtual Machine
- Highly performant execution of Java byte-code
- Everything can be located client-side, a back-end infrastructure is not required

Java Applets & Embedded Code

- Introduction of **real-time multi-player** gaming, requiring a centralized server to dictate communication
- Introduction of **cross-browser** gaming, given that all players have the Java Virtual Machine installed in their systems

Java Applets & Embedded Code

- Minimal graphics, whatever would the native Java graphics libraries provide
- Simple animations that took the gaming experience a bit further
- Simple player interaction with game objects, game control took a small step forward
- No need to refresh the browser

The Domination Of Flash



The Domination Of Flash

- Brought Web-based gaming before a paradigm-shifting situation
- Took over where Java had left from, and enhanced the whole process
- Created new standards for Web gaming, allowing for development of complex and information-sensitive games
- Deserves the credit for setting the fundamentals for what we have today

The Domination Of Flash

- Similar architecture with Applets: an application is built in an offline environment, then embedded to a Web Page and executed via a Flash runtime environment
- Re-defined cross-browser, since the runtime environment for flash became a browser plug-in instead of a separate program in the players' system

The Domination Of Flash

- Games written in ActionScript, a powerful language targeted for developing Web applications
- Proper implementation of physics, game object interaction, animations and game control

The Domination Of Flash

- Really good rendering of 2D graphics, easy and smooth integration of images and sprites, various levels of abstraction for more intuitive programming
- Incorporation of both mouse and keyboard, and extension of the devices so that they can perform more meaningful operations within a gaming context

The Domination Of Flash

WEB-GAME DEVELOPMENT IN JAVA WAS
ABOUT BRINGING ALL THE BOARD
GAMES TO THE WEB.

WEB-GAME DEVELOPMENT IN FLASH
BROUGHT ALL THE GOOD OLD ARCADE
GAMES TO THE WEB!

The Domination Of Flash

- Gradual decline of appraisal for Flash started when browsers became more advanced and more demanding from the third-party software that were running under them
- The Flash Player began to struggle in order to stay compatible, causing various instabilities and reducing the gaming experience

The Domination Of Flash

THE DESIRE FOR SOMETHING NEW AND FRESH WAS EXPRESSED FROM ALL SIDES IN THE WEB COMMUNITY. AND THAT SOMETHING NEW WAS IMMINENT.

THE

PRESENT

HTML5: The Canvas



HTML5: The Canvas

- Similar to Flash Stage: a rectangular piece of real estate to manipulate with code
- Can be drawn directly with images & paths, and transformed on the fly with the native Canvas JavaScript API
- Game control limited to the absolutely necessary: keyboard for typing, mouse for clicking

HTML5: The Canvas

- Native graphics on the browser (finally!)
- Game development with JavaScript more powerful than ever before
- Many third-party libraries to exploit the Canvas capabilities (e.g Processing.js)

Advanced Graphics In The Browser



Advanced Graphics In The Browser

- Very powerful API incorporated in (high profile) Web browsers
- JavaScript bindings to OpenGL ES for implementing hardware accelerated 3D graphics on the Web
- Takes advantage of the 3D context of the Canvas

Advanced Graphics In The Browser

- A big contender for Flash in the high performance, “fancy” graphics and animations world
- Main disadvantage of Flash: need for a third-party plugin in the browser, whereas WebGL is **native & multi-platform**
- Might not be fully supported yet, but it's getting there

Real-Time Communication With WebSockets



Real-Time Communication With WebSockets

- A full-duplex communication channel through a single socket over the Web
- Takes conventional HTTP communications to the next level
- Introduces the notion of **real-time communication**, in a **native** manner

Real-Time Communication With WebSockets

- A whole different set of Web-game types opens up for development; **multi-player** is now an option, but without any further pre-requisites: **cross-browser**
- Libraries which take care of all the groundwork offer WebSocket APIs which are simple and intuitive to use
- Very promising high-scale, real-time, multi-player games ahead

Smooth Back-End Systems



Smooth Back-End Systems

- A platform built on Google's JavaScript Runtime (V8 Engine) for building fast & scalable network applications
- Non-blocking, event-driven I/O model, suitable for data-intensive, real-time & distributed Web-apps
- Does not use multi-threading for concurrent computations; it uses the event-driven model instead

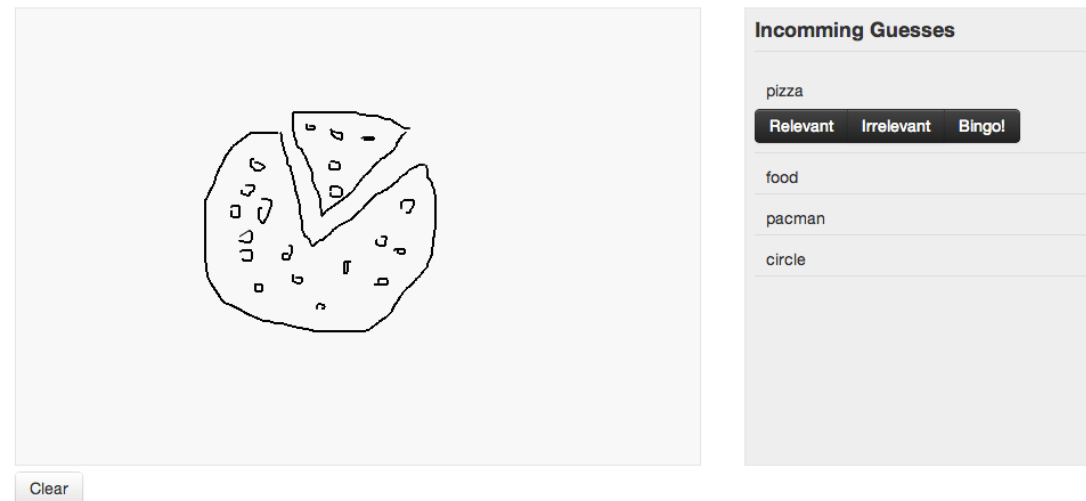
Smooth Back-End Systems

- The big shift: **server-side JavaScript**
- A single programming language throughout the whole Web game: coherence, homogeneity and maintainability

Smooth Back-End Systems

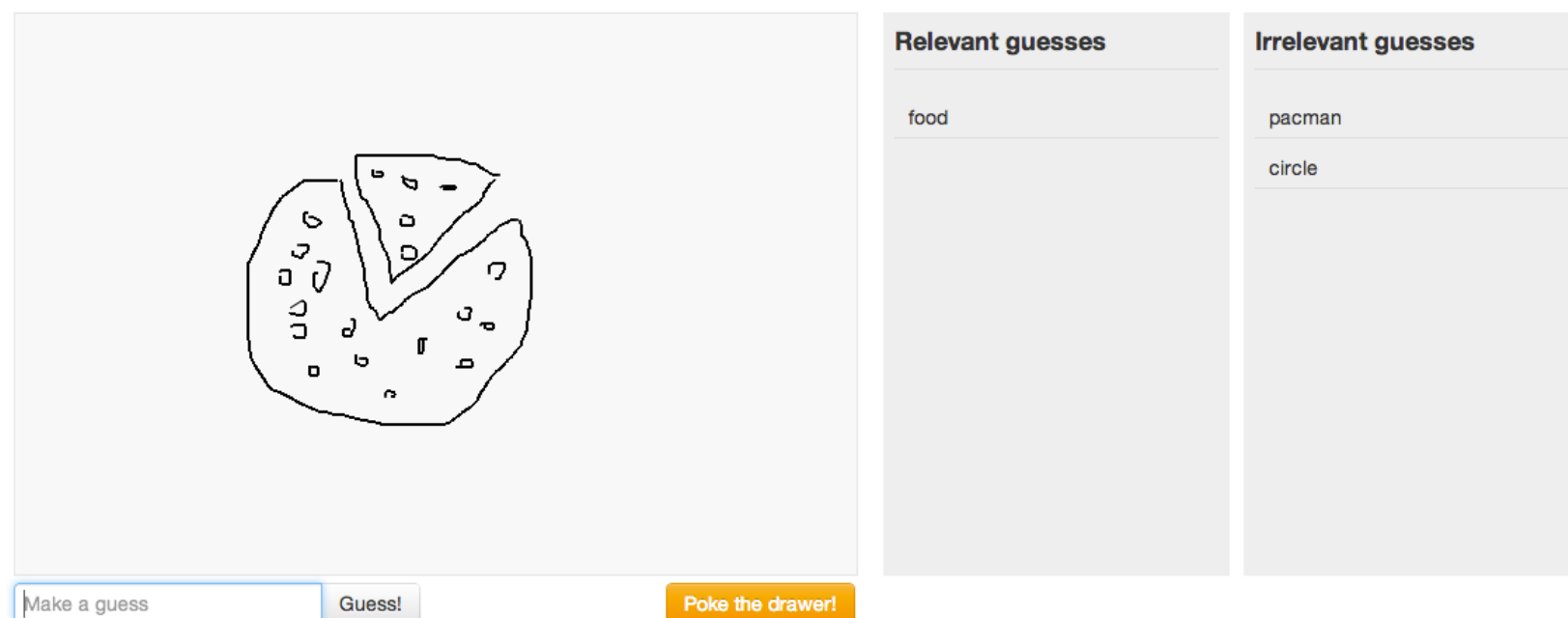
- Blazing fast I/O and game logic computation/communication over a distributed system
- Scalable online games with big amounts of concurrent players
- Massive development community & support for cutting edge features

Case Study: HTTPICTIONARY



You are looking for:

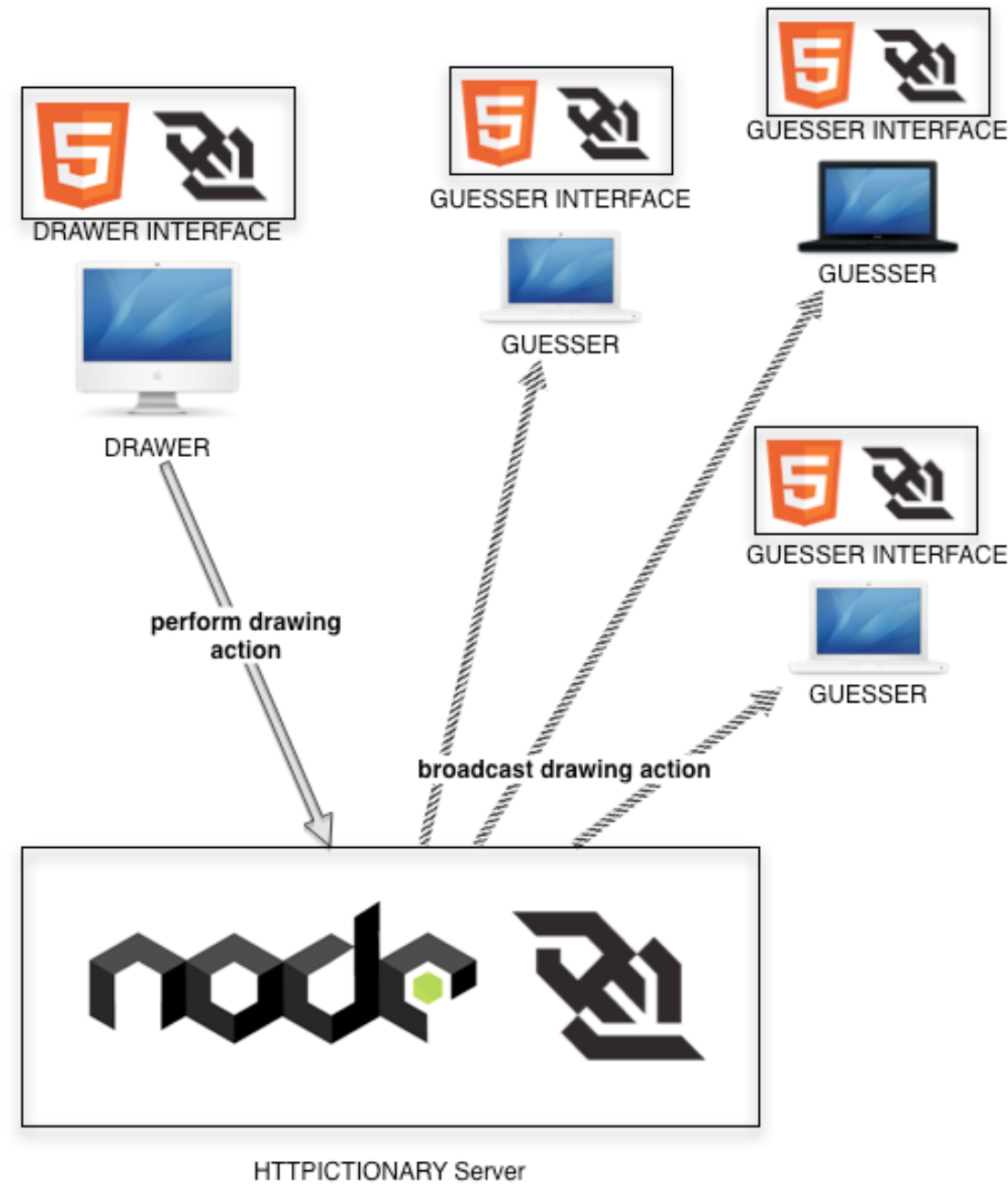
Pizza



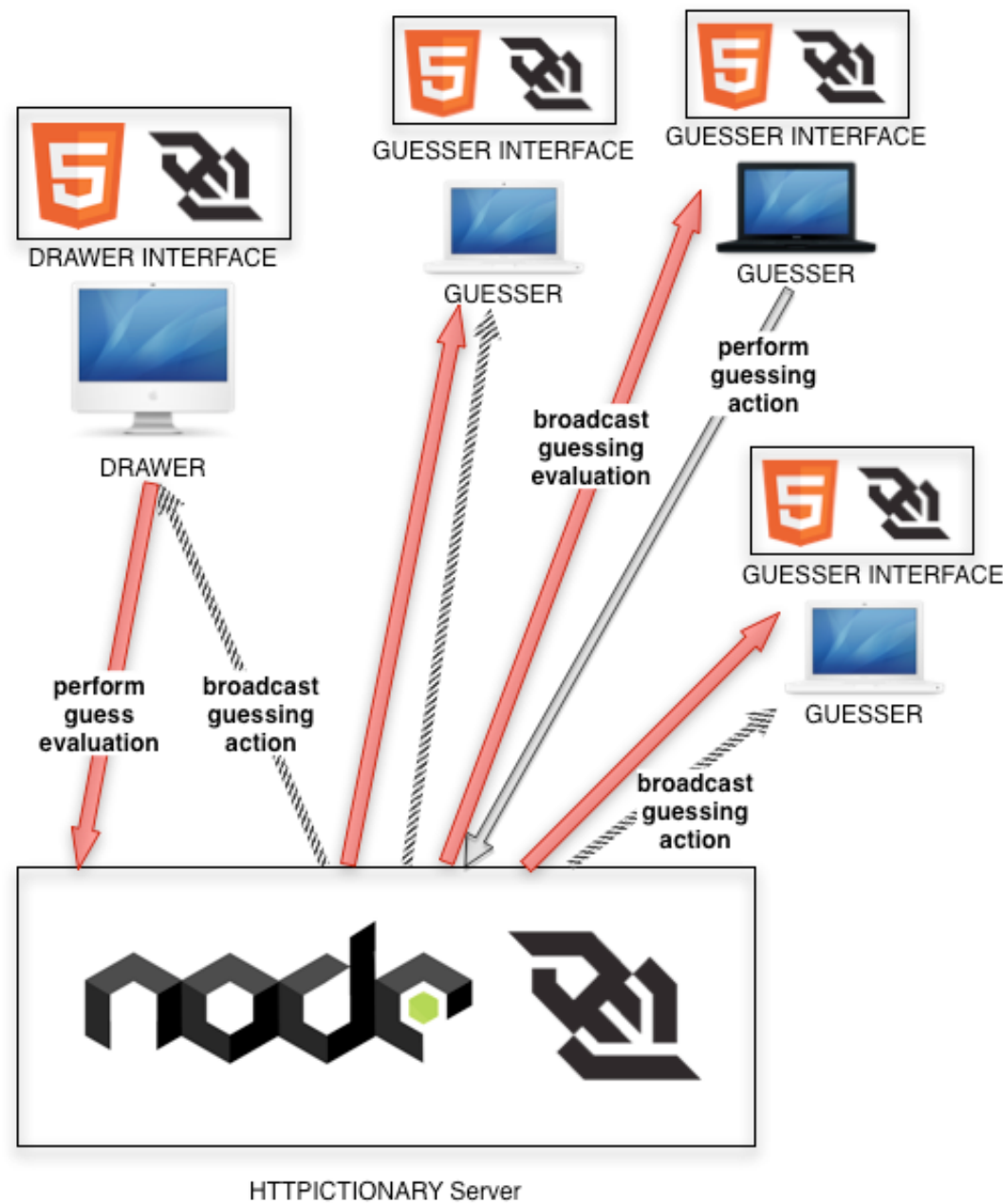
Case Study: HTTPICITIONARY

- Two player roles: the drawer & the guesser
- Guessers see what the drawer is drawing in real time and guess what it may be
- Drawer receives the guessers' guesses in real time and evaluates them as either relevant, irrelevant & bingo (found)

Case Study: HTTPICITIONARY



Case Study: HTTPICITIONARY



More Advanced Control



KINECT™
for  XBOX 360.

More Advanced Control

- Need to give credit to Wii-Mote, for revolutionizing the way of game control
- Microsoft's Kinect sensor though, took the full body control to the next level

More Advanced Control

- Depth sensor
- Color camera
- Four-microphone array for full-body 3D motion capture
- Facial recognition
- Voice recognition

More Advanced Control

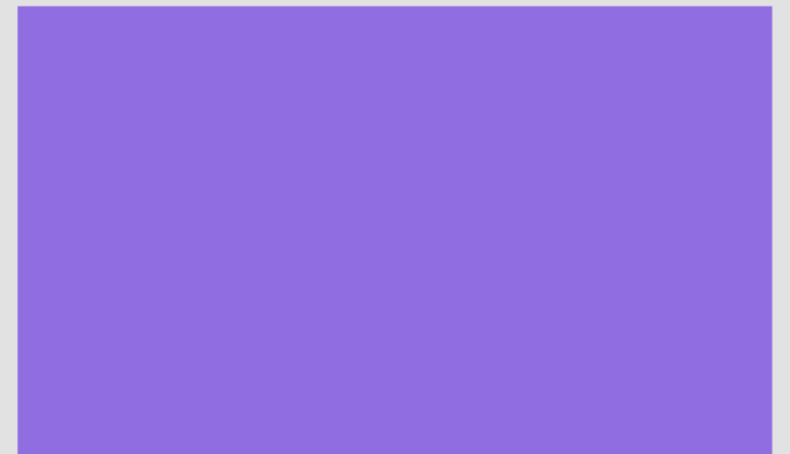
- Deeply programmable: official MS Kinect SDK for C++, C#, VB
- Unity3D engine took big advantage of the Kinect, made handlers for it (C++) in its games and the results can be deployed on the Web with Unity Web Player: Flash-like
- Client-side scripting APIs also available for Kinect: Zigfu

More Advanced Control

KINECT COULD TURN THE BROWSER
INTO A XBOX, AND IT PROBABLY WILL!

Case Study: Find The Color!

Hits: 0



Case Study: Find The Color!

- Game asks from the player to hit a color from the grid
- Player has to point their cursor and push on the color they are asked to find
- Moving and pushing the cursor is done by tracking the player's arm gestures with a Kinect sensor
- Exclusively implemented on the client-side

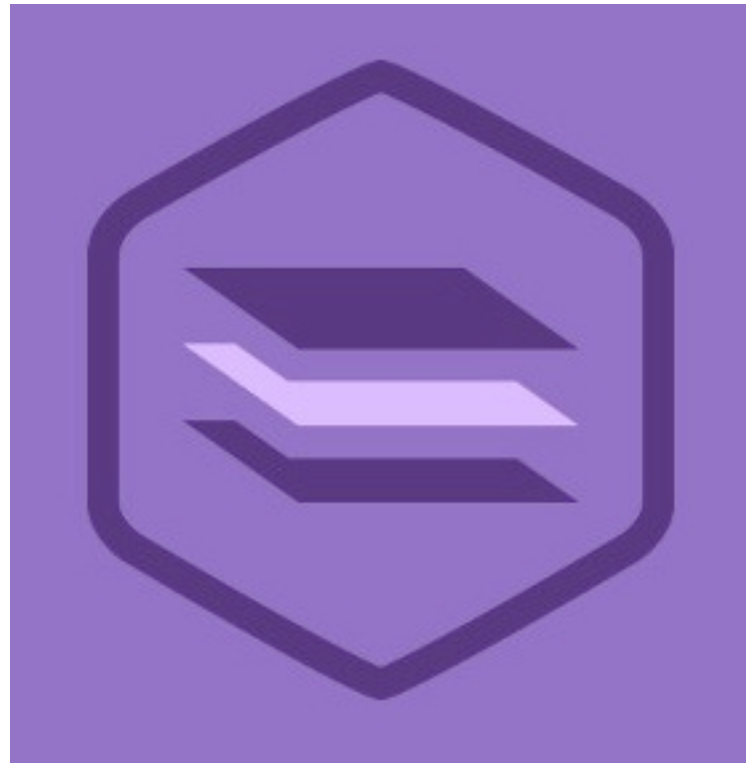
Case Study: Find The Color!



THE

FUTURE

Web Game Frameworks



Web Game Frameworks

- Distributed Systems Technology Software Engineering have been moving things forward (MVC, notion of Middle-ware etc)
- Numerous libraries out there to make the Web-game developers' life easier
- Open-source software expose technologies which create a new world order in game development in general

Web Game Frameworks

DEVELOPING (WEB) GAMES CAN BE A
ONE-MAN-JOB!

Web Game Frameworks

- Taking care of the underlying infrastructure
- Standardized protocols and layers of abstraction for communication between different pieces of hardware & software
- Communal development, knowledge sharing, creating more cutting-edge features and more impressive games
- Setting new standards for gaming experience

Intelligent Control



Intelligent Control

- Game controllers have evolved radically in the last decade
- An intelligent and customizable game controller can provide with ideas for new games
- Collaborative game control, combining multiple controllers (full-body tracking, sound, facial expressions, attention span?, brainwaves?..)

Smart-phones



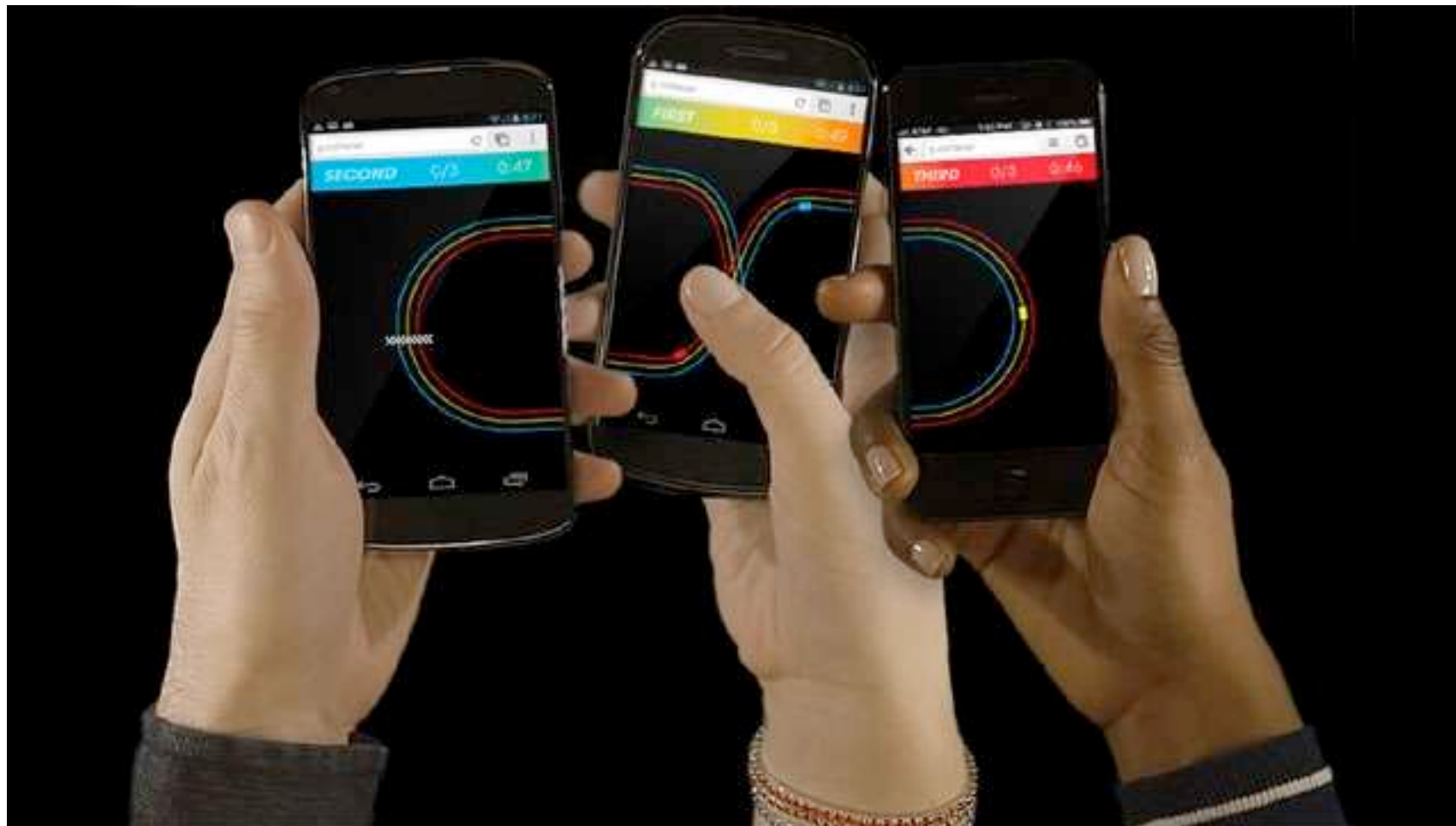
Smart-phones

- Extension of each-and-everyone's arm nowadays
- Powerful devices (small-scale computers), could act as a gaming platform and a controller at the same time
- Big attention to smart-phone browser features: setting new standards for feature support in the Web: if Mobile Safari supports WebGL, Desktop Safari should absolutely support it as well

Smart-phones

SMART-PHONES COULD BECOME THE
NEXT GENERATION'S "NINTENDO
GAMEBOY"!

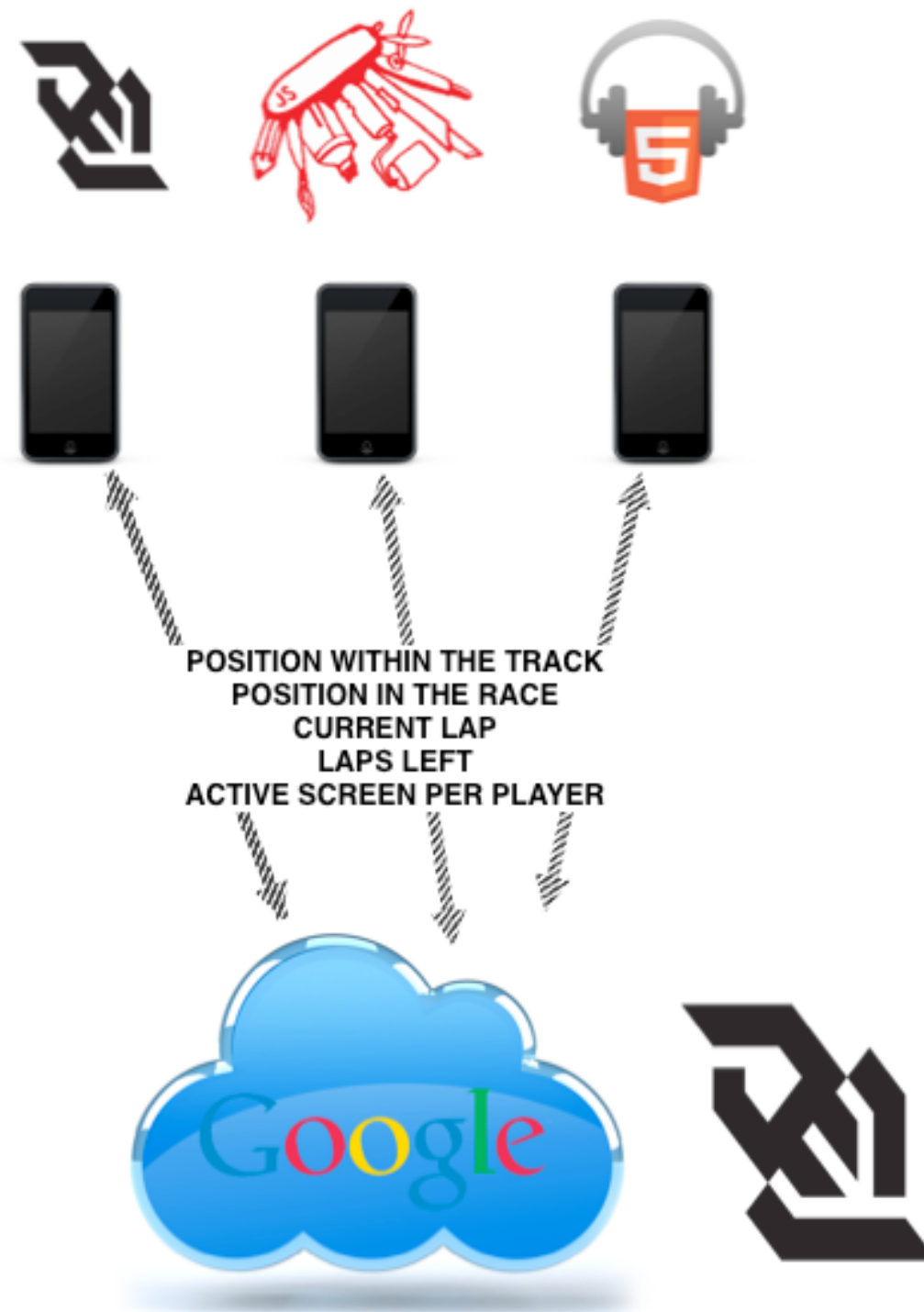
Case Study: Racer



Case Study: Racer

- A Chrome Experiment by Google, playable only on smart-phones
- Real-time, multi-player racing game
- The race track is divided by the number of players, and each one has a portion of it on their screen
- Speeding up achieved by pressing on the screen

Case Study: Racer



Conclusions

- Gaming is one of the primary usages of personal computers since the early days, and so is the Web today
- Big evolution since the past, strong combination between the two brings new flavors to gaming experience
- The future looks (at least) promising for Web-based gaming

THANK YOU FOR YOUR
ATTENTION!

ANY QUESTIONS?