Feel the ‘Vibe’: Enhancing haptics for accessibility and more fun

Harry Wang, harryhwang@tencent.com
Selen Xu, selenxu@tencent.com
Diverse development style, Different framework, architecture, toolchain, language

30+ 05, 10+ Browser, 500+ app store, customization

Diverse hardware platform, screen size, vendor specific driver

Full support by Tencent games

Dynamic Resource Scheduling

Dynamic tracking of performance

Unified adaptation

MTGPA Immersive gaming experience

Technical support for operations

GAME Client

Unified performance optimization

Pre-download

adaptation

MTGPA-SDK

MTGPA Cloud Control System

ROM

Strategy Callback

Safely control

Resource Scheduling

MTGPA-Integration
Background
Haptics amend human machine interaction

- The haptics term comes from the Greek “haptikos” meaning “concerning the sense of touch”.
- In the tech world haptics identifies all the technologies that provide the sensation of digital touch feedback, also called haptic feedback.

"The feel of touch is about five times faster than vision."
-- Ambient Touch: Designing Tactile Interfaces for Handheld Devices, I. Poupyrev, S. Maruyama, J. Rekimoto

"All devices [will] need to interact.
If a thing does not interact, it will be considered broken.
"
-- The Inevitable, by Kevin Kelly
Haptics Taxonomy

Source: Haptics Industry Forum
Simulate Haptics Feedback

- Button is perceived as a function of the applied pressure, then a click happens, and the force suddenly drops down.

- Texture: Periodic space based skin compressing
- Stiffness: Non-Periodic space based skin compressing
- Vibration: Periodic Time based skin compressing
- Force: Non-periodic Time based skin compressing
Haptics for Game Developers

Devices that require haptics

- Sense of reality
  - Visible & Tangible
- Immersive experience
  - Audiovisual, force-haptic multisensory stimulation
- Interactive experience
  - Direct control of virtual objects, real-time sensory feedback

Console Games

Haptic feedback is the core experience

Mobile Games

Lack of feedback, apart from sound and visuals,
The sense of touch provides a powerful channel for communication and engagement.

**Improved usability**
- make interfaces and controls easier and more intuitive

**Increased accessibility**
- providing an additional sensory channel for information

**Enhanced user experience**
- mimicking real-world tactile sensations, make digital interfaces feel more natural

**Differentiation**
- make a product stand out through its unique and engaging feedback

**Improved perception of quality**
- make products feel higher quality and more precision engineered

**Subtle notifications**
- allow for private, subtle communication to the user that is not audible or visible to others

**Increased emotional connection**
- foster an emotional connection and feeling of intimacy with a product.
Solutions
Diverse mobile devices and components make it difficult to achieve a consistent experience.

Mobile phone model distribution among gamers

Mobile devices & Actuators

- Bar ERM
- Coin ERM
- Z-axis LRA
- X-axis LRA
Opportunities and Challenges

Opportunities:
- Growing demand for more complex and immersive user experiences.
- The extensive use of linear resonant actuators (LRAs).
- Highly precise and responsive actuators for rendering vibrations.

Challenges:
- The easy-to-use editing tool which can be adapted to the game engine.
- The unified standard interface between application, operation system and hardware.
- Rendering smooth and unified vibration effect.
Start designing a Haptic Effect

**Design a Haptics Effect**

**Independent design**
Take some system vibrations built into mobile devices, such as incoming calls and notifications, as an example. Directly establish vibration rules to express the occurrence of specific events.

**Auditory-based design**
Haptic effect is designed to be used in scenarios that can be standalone, combined with auditory/audio, or combined with visual/animation.

**Audio filtering to generate HE effect scheme**
According to the audio waveform, the audio filter is transferred to the HE Event algorithm to generate the best HE description of the motor vibration.

Sound wave of a shotgun *from Peacekeeper Elite (Good for Peace)*
Versatility and flexibility are the core design concepts throughout the entire solution.
Optimize the Performance

Optimization Solution

1. Asynchronous call for the time consuming tasks
2. Binary encoding of the haptic data, and reducing the data volume by 80%
3. Improving the caching for haptic data and decreasing the IO operation

Performance test index

1. Time consumption of calls to the underlying vibration API
2. Memory consumption
3. HE impact on APP frame rate
4. Impact on device power consumption
## Vibration playback mechanism

### Static Mode
- ✓ read the vibration effect resources designed in advance at runtime all at once
- ✓ Audio, video and vibration play independently
- ✓ more suitable for shorter duration vibration effects
- ✓ Long video/audio design and playback defects

### Streaming Mode
- ✓ In the unit of vibration event, read and play according to the time stamp in HE
- ✓ Good for long vibration effects
- ✓ Synchronize with audio and video at the granularity of vibration Event

## Vibration Synchronization Scheme

### Synchronize with audio
- ✓ suitable for music player
- ✓ Audio segmentation, prefabricated multiple HE resources
- ✓ Synchronize by track, play regularly

### Synchronize based on video and animation effects
- ✓ Suitable for games, video display
- ✓ Video segmentation, prefabrication of multiple HE resources
- ✓ Timed playback according to video timestamp
Game Haptics Practice

Function

17 custom configurations

Core elements

4 core elements, 200+ effects

Refinement effect

Application range

5 million+
Users open the function

20k+
models support

200 million+
devices support
Implementation with MTGPA SDK

For Android

- Support two playback mode: HQ and basic;

```java
HapticPlayer hapticPlayer = HapticPlayer.create(context, HapticPlayer.STANDARD_HAPTIC_PLAYER);
hapticPlayer.play(heJson);
```

```java
HapticPlayer hapticPlayer = HapticPlayer.create(context, HapticPlayer.ANDROID_PATTERN_PLAYER);
hapticPlayer.play(heJson);
```

- Configurable playback parameters;

```java
int loop = 10;
int interval = 100;
int amplitude = 255/2;
hapticPlayer.play(heJson, loop, interval, amplitude);
```
• **JSON format**
• **Pattern**
  - describe the haptics effect, composed by one or more events
• **Event**
  - Describe a basic haptics segment
  - ‘continuous’ or ‘transient’ type
  - Start timing
  - Duration
  - Fine control of intensity and frequency
  - Haptics parameters
• **Curve**
  - Describe parameters for ‘continuous’ haptics event
Accessibility
In 2020, the World Health Organization estimates that 596 million visually impaired people worldwide.

• We’ve talked to 100+ visually impaired users

• Haptics can help visually impaired users to gain more information and understanding.
What do visually impaired people use their mobile phones for when accessing the internet?

Here we are surprised to see that this part of the user’s information needs for the Internet involves a very large number of scenarios, and here are listed their specific needs for travel, transactions, social and other areas.
Taking intensity, frequency and duration as variables, several groups of experiments were conducted to determine the accuracy rate.

The compared results of the three are as follows:

<table>
<thead>
<tr>
<th>Variables</th>
<th>Highest correct rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>intensity</td>
<td>0.978</td>
</tr>
<tr>
<td>frequency</td>
<td>0.91</td>
</tr>
<tr>
<td>duration</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Coding methods of Six-point Braille:

- Take the intensity as the independent variable:
The selected intensity levels are 30 and 70.

- Take the frequency as the independent variable:
The selected frequency levels are 40 and 70.

- Take the duration as the independent variable:
The frequency level is 40, the intensity level is 70.
Haptics Solution for Input Method

Inconvenient to use audio
Introduce different haptic effects for specific characters on the input method keyboard. When audio feedback is inconvenient to use.

Protect privacy
Recognize letters and numbers on the keyboard by haptics, while protecting user privacy to increase input efficiency and accuracy.
**Walk Navigation Accessibility Mode**

When using walking navigation, if the user veers off the designated route, they will be alerted through vibrations and an audio announcement, guiding them back to the right direction.

**Transit Navigation Accessibility Mode**

Through customized vibration effects, users are reminded in four different scenarios: waiting at the station, about to arrive at the station, arriving at the station, and arriving at the destination.
Promote Haptics as a Common Human Language

- Integration with Web contents, support by Web platform, adoption in Accessibility guideline

Common understanding, standard API and data, unified user experience, evaluation and metrics;

“human language:

language that is spoken, written

or signed (through visual or
tactile means) to communicate

with humans

-- WCAG 2.2
4 Standardization
Need for Haptics Standards

Service & Content

App & Platform

OS

Hardware + Device

Developers

Users
Need for Haptics Standards

**Without Standards**

- Market fragmentation - Walled Gardens
  - Proprietary APIs
  - Incompatible HW-to-SW interfaces
  - Different motor characteristics
- Content creators and application developers are de-incentivized to include haptics in their applications
  - Changes to lower levels of the haptic stack typically necessitate changes to the upper layers
- Adverse impact on end-user adoption

**With Standards**

- Standards harmonize haptics offerings from vendors
  - Without compromising vendor differentiation
- Content creators and application developers incentivized to incorporate richer haptic experiences
  - Leverage standard interfaces throughout the haptic stack
- Walled gardens give way to a flourishing haptic ecosystem that benefits all stakeholders.
Haptics Stack & Standard Activities

Service & Content
- Haptics Design tool
- ISO MPEG-I
- IETF Haptics Media
- Audio-to-Haptics tool

App & Platform
- Web API & runtime
- IEEE P2861.3/Haptics Industry Forum
- Khronos OpenXR

OS
- USB-IF Haptic Controller HID

Hardware + Device
- Hardware interface

Developers

Users

Standard Activities

Description
MPEG ISOBMFF (Haptics in ISO Base Media File Format)
- Establish haptics as a first-order media type, along with audio and video, in all ISO media files (e.g., MP4)
- Next step: flesh out the haptic codec-specific header ‘boxes’ in ISOBMFF

MPEG-DASH (Haptics in media streaming)
- Leverages the ISOBMFF standard above
- Initial proposal made; waiting for haptic codec to be finalized
- Will enable haptics to be part of all media streaming (Netflix, Hulu, etc.)

MPEG-I (Haptics in Immersive Media – AR/VR/Gaming)
- Phase 1: Develop a basic haptic codec standard – in CD ballot November 2022
- Phase 2A: Add support for spatial haptics, scene description – CfP 1H 2023
- Phase 2B: Add support for interactivity, avatars, XR experiences (metaverse) – CfP 1H 2024
• Avatar: body (or part of body) representation
• Perception: haptic perception containing channels of a specific modality
• Device: physical system having one or more actuators configured to render haptic sensation corresponding with a given signal
• Channel: component in a perception containing one or more bands rendered on a device at a specific body location
• Band: component in a channel for containing effects for a specific range of frequencies
• Haptic Effects: component of a band for defining a signal, consist of a haptic waveform, or one or more haptic keyframes
• Key frames: component of an effect mapping a position in time or space to an effect parameter such as amplitude or frequency
Haptics in MPEG - Haptic File Format

Interchange file format
File type: .hjis, .ivs, .ahap
A JSON implementation of the data model human-readable and can be manually edited

Binary file format
File type: .hmpg
One-to-one correspondence with the interchange format data compression is applied for the data at the band level

Streaming
File type: mihs
Streaming format to transport the MPEG-I haptic data (MIHS) Sequence of MIHS Units containing MIHS packets which includes metadata and haptic effect data
Two types of input:
- Descriptive content (.ivs, .ahap, .hjif)
- PCM content

Three types of output:
- interchange file format (.hjif) encoded in a JSON readable form
- compressed format encoded as a binary file (.hmpg)
- a streaming format defining a packetized bitstream
IEEE P2861.3 Haptic Enhancement for Mobile Game

- Standard architecture for mobile haptics platform: interface, data, evaluation

**Haptic System Architecture**

<table>
<thead>
<tr>
<th>APP</th>
<th>OS</th>
<th>Vibratory Equipment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customize HE</td>
<td>Haptic Core</td>
<td>Actuator Driver</td>
</tr>
<tr>
<td>Customize HE Library</td>
<td>HE Decode</td>
<td>X-axis Actuator</td>
</tr>
<tr>
<td>Haptic SDK</td>
<td>HE Algorithm</td>
<td></td>
</tr>
<tr>
<td>Play Back</td>
<td>Control Algorithm</td>
<td></td>
</tr>
<tr>
<td>Protocol Conversion</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Interfaces**

- HEI: Haptic Effect Interface
- HRI: Haptic Runtime Interface

**Functional Requirements**

- Haptic SDK: data conversion and playback
- Haptic Core: data processing and control

**Data format**

- Event based JSON format

**Processing flow**

- Initialization and playback

**Evaluation**

- Evaluation procedure and Performance metrics
Haptics in Khronos OpenXR

- Haptic extensions to the OpenXR API will provide developer-facing haptic interfaces transportable across a variety of XR platforms.
- OpenXR is a platform API intended to standardize the interface between app developers/game engines and XR hardware devices.
- Input and Haptics API: Applications define actions, the runtime binds actions to input devices. Action sets are application-defined collections of actions.
Haptics in IETF: Haptics as a TOP-LEVEL Media Type

• IETF – Internet Engineering Task Force
  • Internet standards body, developing open standards through open processes
  • Top-level media types: application, audio, font, image, message, model, multipart, text, video

• Proposal to register haptics as a new top-level media type
  • Latest Internet Draft: https://datatracker.ietf.org/doc/draft-ietf-mediaman-haptics/
  • Now on WG lastcall

• How does it help the haptics community?
  • An IETF haptics RFC will be a foundational, enabling standard
  • Will enable proliferation of haptics media in a standardized manner in all internet media and communications
  • Will enable IANA registration of multiple haptic sub-types under the ‘haptics’ top-level type.

• Aligns with other haptics standardization activities (MPEG, OpenXR, etc.)
• Will facilitate adoption of haptics by the industry – benefitting all stakeholders in the haptics value chain
Way Forward
Some Thoughts on W3C Vibration API

W3C Vibration API (https://www.w3.org/TR/vibration/), implemented on Chrome and Firefox for Android.

Possible Amendment

- Align with system API
- Support vibration waveforms
- Define requirements for data format
- Adapt to device capabilities
- Support the integrated playback control

EXAMPLE 1

```javascript
// vibrate for 1000 ms
navigator.vibrate(1000);

// or alternatively
navigator.vibrate([1000]);
```

In the following example the pattern will cause the device to vibrate for 50 ms, be still for 100 ms, and then vibrate for 150 ms:

EXAMPLE 2

```javascript
navigator.vibrate([50, 100, 150]);
```

The following example cancels any existing vibrations:

EXAMPLE 3

```javascript
// cancel any existing vibrations
navigator.vibrate(0);

// or alternatively
navigator.vibrate([]);
```
Possible Amendment

- Extend support to more actuator types
- Add support for multiple actuators
- Align with system API
- Support vibration waveforms
- Define requirements for data format
- Support the integrated playback control

W3C vibration extension for gamepads (https://w3c.github.io/gamepad/extensions.html).

```webidl
[Exposed=Window]
interface GamepadHapticActuator {
  readonly attribute GamepadHapticActuatorType type;
  boolean canPlayEffectType(GamepadHapticEffectType type);
  Promise<GamepadHapticResult> playEffect(
    GamepadHapticEffectType type,
    optional GamepadEffectParameters params = {});
  Promise<boolean> pulse(double value, double duration);
  Promise<GamepadHapticResult> reset();
};
```

```
enum GamepadHapticEffectType {
  "dual-rumble"
};
```

"dual-rumble" effect type

"dual-rumble" describes a haptics configuration with an eccentric rotating mass (ERM) vibration motor in each handle of a standard gamepad. In this configuration, either motor is capable of vibrating the whole gamepad. The vibration effects created by each motor are unequal so that the effects of each can be combined to create more complex haptic effects.

A "dual-rumble" effect is a fixed-length, constant-intensity vibration effect intended for an actuator of this type. "dual-rumble" effects are defined by `startDelay`, `duration`, `strongMagnitude`, and `weakMagnitude`, none of which are required.
Some Thoughts on Microsoft Proposal of HapticsDevice API

Microsoft Proposal

• Tied to USB-HID use case
• Pre-defined waveform ID
• Trigger and re-trigger mechanism
• Enhance the gamepad extension API

Possible Amendment

• Align with system API
• Define requirements for data format
• Adapt to device capabilities
• Support the integrated playback control
• Add support for multiple actuators

Microsoft Proposal

```javascript
partial interface PointerEvent {
    [SameObject] readonly attribute HapticsDevice? haptics;
}

interface HapticsDevice {
    void play(HapticsPredefinedWaveform predefinedWaveform);
    void stop();
}

dictionary HapticsPredefinedWaveformInit {
    required unsigned short waveformId;
    float intensity = 1.0;
    DOMString vendorId = "";
    sequence<HapticsPredefinedWaveform> alternates = {};
}

interface HapticsPredefinedWaveform {
    constructor(HapticsPredefinedWaveformInit predefinedWaveformInit);
    readonly attribute unsigned short waveformId;
    readonly attribute float intensity;
    readonly attribute DOMString vendorId;
    readonly attribute FrozenArray<HapticsPredefinedWaveform> alternates;
};
```
### With pre-defined JSON format

```java
partial interface HapticPlayer {
    boolean isAvailable();
    void play(DOMString pattern, int loop, int interval, int amplitude, int freq);
    void stop();
};
```

### Generate on-demand

```java
interface HapticDevice {
    boolean isAvailable();
    void play(HapticEffect effect);
    void play(DOMString effect);
    void stop();
}
```

```java
interface HapticEffect {
    constructor(HapticsPredefinedWaveformInit predefinedWaveformInit);
    readonly attribute unsigned short version;
    readonly attribute DOMString description;
    readonly attribute DOMString createTime;
    readonly attribute FrozenArray<HapticEvent> alternates;
};
```

```java
interface HapticEvent {
    readonly attribute unsigned short eventType;
    readonly attribute long relativeTime;
    readonly attribute long duration;
    readonly attribute long intensity;
    readonly attribute long frequency;
    readonly attribute FrozenArray<HapticCurvePoint> alternates;
};
```

```java
interface HapticCurvePoint {
    readonly attribute long time;
    readonly attribute long intensity;
    readonly attribute long frequency;
};
```
Summary & Discussion

• Haptics goes beyond just vibration. As the actuator performance and algorithm improves, it makes the digital experience feel more tangible and real.

• Tencent MTGPA Haptics suite is being widely supported by smartphones, and it enables more sophisticated tactile experiences for games and accessibility use cases (e.g. mobile map, input method).

• Technical know-how on mobile haptics development is reviewed, and the global standard efforts by the community help accelerate adoption of haptics as a common human-machine interface.

• Some preliminary thoughts on the relevant W3C APIs and proposals are meant as a starting point for further discussion.

Open Questions

• Should web API align with the native system API?
• Should haptics effect (waveform) be integrated with web contents?
• How to enable haptics to the extensive use cases, especially for accessibility?
• Should there be a guideline for haptics design to carry specific information as an independent sensory channel?
THANKS