Are All Games Equally Cloud-Gaming-Friendly?

-- An Electromyography Approach

Yeng-Ting Lee  National Taiwan University
Kuan-Ta Chen  Academia Sinica
Han-I Su  Academia Sinica
Chin-Laung Lei  National Taiwan University
Long-Lasting Problems of Gamers

- Game software becomes more complex
  - The overhead of setting up a game is significant
  - Locked by a certain computer

- Games have software and hardware compatibility with computers

- Computer hardware upgrade demands
Cloud Gaming

- Real-time game playing via thin clients
Benefits of cloud gaming

- Frees players from the need of indefinitely upgrading their computers
- No setting up overhead and compatibility issues for trying out a game as everything needed will be settled in data centers by game operators
- People can play games anywhere, anytime
Cloud Gaming Services
Unavoidable Delays

- Network delay: \((t_1 - t_0) + (t_3 - t_2)\)
- Processing delay: \(t_2 - t_1\)
- Playout delay: \(t_4 - t_3\)
A Comparison with “Traditional” Online Games

- Unavoidable extra delays
  - Video encoding at the server
  - Video decoding and playout buffering at the client

- Less effective delay compensation
  - Game states (e.g., game objects’ positions and velocity) are not available at the client side

Cloud games are intrinsically more susceptible to latency than online games.
Our motivations

- Q1: How the response latency in cloud gaming affects users’ experience?
- Q2: Are different games equally sensitive to latency?

Searching for opportunities to enhance cloud gaming QoE
Definition

- **Real-time strictness (RS)**
  - How much a game’s QoE degrades when the latency is higher?

- **Cloud-gaming friendliness**
  - A cloud game’s susceptibility to latency in terms of its QoE
Roadmap

- Cloud-gaming friendliness
- inversely proportional to
- Real-time strictness
- divided by Latency
- QoE degradation

- game control inputs
- game screen dynamics
- regression model

Latency
QOE MEASUREMENT
QoE Measurement Approaches

- **MOS (Mean Opinion Scores)**
  - Inaccuracy due to recency effect
  - Costly due to its fewer and coarser (i.e., five-level) responses

- **SSCQE (Single Stimulus Continuous Quality Evaluation)**
  - Requires subjects to continuously report their present gaming experience
  - May disturb game play and diminish the flow experience
Facial EMG approach

(EMG: Electromyography)

1. Continuous emotion measures (can be at a rate of 1000 Hz or even higher)

2. Does not disturb game play

3. Objective since the emotional indicators are directly measured rather than told by subjects
Facial EMG Measurement Setup

The corrugator supercilii muscle

Negative emotions

The amount of annoyance caused by latency
Measurement devices

Electrodes

Wires

PowerLab 16/30
During game play...
Latency Emulation

- Reasons
  - Full control of the latency injected into the input-response loop
  - Free study of games that are not available on current cloud gaming platforms

- Our approach
  - Use SetWindowsHookEx API (in Windows) to intercept all the keyboard and mouse inputs
  - Each of the input events is firstly queued in a buffer and later replayed after a desired time $t$
Selected Games

- **ACT**
  - LEGO Batman (Batman)
  - Devil May Cry (DMC)
  - Sangoku Musou 5 (Dynasty Warriors 6) (SM5)

- **FPS**
  - Call of Duty: World at War (COD)
  - F.E.A.R 2 (FEAR)
  - Unreal Tournament 3 (Unreal)

- **RPG**
  - Ys Origin (Ys)
  - Loki: Heroes of Mythology (Loki)
  - Torchlight (Torch)
Experiment overview

- Latency levels: 0, 50, 100, 200, 400 ms
- Each round: 10 minutes with 5-minute breaks
- Latency is randomly applied during game play
Trace Summary

- **Subjects**

<table>
<thead>
<tr>
<th># Subjects</th>
<th>15 (14 males and 1 female)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>19–28 (mean 24.7 with std. dev. 2.9)</td>
</tr>
<tr>
<td>Game experience</td>
<td>Mean 4.6 years with std. dev. 1.3 years</td>
</tr>
<tr>
<td>Game play time each day</td>
<td>Mean 3.4 hours with std. dev. 1.1 hours</td>
</tr>
</tbody>
</table>

- **Trace**

<table>
<thead>
<tr>
<th>Total trace length</th>
<th>1,350 minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. trace length each subject</td>
<td>90 minutes</td>
</tr>
<tr>
<td>Avg. trace length each game</td>
<td>150 minutes</td>
</tr>
<tr>
<td># fEMG signals measured</td>
<td>81,008,940 samples</td>
</tr>
<tr>
<td>Avg. input rate</td>
<td>11.7 inputs/sec</td>
</tr>
</tbody>
</table>
QUANTIFYING REAL-TIME STRICTNESS OF GAMES
Overall EMG potentials

![Graph showing the relationship between fEMG potential (mV) and latency (ms). The correlation coefficient (cor) is 0.998.](image)
EMG Potentials for each game

1. Diverse baseline EMG potentials for each game
2. The increasing rates of EMG potential are game-dependent as well
Deriving real-time strictness (RS)

\[ RS = \frac{\Delta fEMG \text{ potential}}{\Delta \text{latency}} \]

\[ \Delta fEMG \text{ potential} \]

\[ \Delta \text{latency} \]
RS of the studied games

- In general, FPS > RPG > ACT in terms of RS
- Game pace ↑ , RS ↑ , latency-critical ↑
MODELING THE REAL-TIME STRICTNESS OF GAMES
Real-time strictness modeling

- **Motivations**
  - Subjective measurements are labor-intensive
  - To understand the design elements that cause high RS

- **Expected prediction model**
  - Output: RS estimate
  - Inputs
    - User game play inputs
    - User game play video recording
Our conjecture

- How a game responds to players’ commands is associated with its real-time strictness.

- If its commands is “lightweight”
  - Simple, fast, local moves
  - Timing is important ➔ higher RS

- If its commands are “heavy”
  - Associated with long and large amounts of animations
  - Timing is not critical ➔ lower RS
Illustrations for command “heaviness”
Feature Extraction

- **Screen dynamics**
  - Based on the motion vectors in H.264 encoded video clips
    \[
    \sum_{i=1}^{p} \sum_{j=1}^{f} \frac{sd(|mc_{ij1}|, |mc_{ij2}|, \ldots, |mc_{ijm}|)}{f \cdot p} / \text{inter-frame-time}
    \]
  - Basically the variation of motion vectors’ lengths on each game frame

- **User input rate**
  - Used for normalization

\[
\text{command heaviness} = \frac{\text{screen dynamics}}{\text{input rate}}
\]

(indicates the average complexity of screen changes caused by each keyboard/mouse input event)
Command heaviness vs. RS

Are All Games Equally Cloud-Gaming-Friendly? / Kuan-Ta Chen
RS prediction

real-time strictness = 1.8 – 0.28 × command heaviness
Application #1: Balance games’ QoE degradation due to latency

- **Scenario**
  - $N$ users are playing different games at the same time
  - Users experience different latencies and games have different RS
  - $\Rightarrow$ Each player experiences different levels of QoE degradation

- **Usage**
  - Use the model to infer which players are having a worse gaming experience than others
  - Prioritize the server’s resources, such as CPU and GPU, to reduce those players’ latencies and thereby mitigate QoE degradation they would otherwise experience
Application #2: Co-optimizing data center cost and gaming experience

- **Scenario**
  - $N$ data centers, each has distinct operation cost (electricity and labor)
  - Whenever a user signs in, we need to assign a data center to him for real-time game play
  - Question: Which data center should we assign to the player?

- **Usage**
  - Use the model to predict users’ QoE in all the cases and choose the data center which provide a “just good enough” gaming experience
Summary

- Not all games are equally friendly to cloud gaming due to different design.

- The cloud-gaming-friendliness factor could be predicted by analyzing how a game is played.

- Such a prediction model can be used to simultaneously optimize server resource usages and gaming experience and help achieve a win-win situation for both the industry and gamers.
Thank You!

Kuan-Ta Chen
Academia Sinica

http://www.iis.sinica.edu.tw/~swc