Site:	http://peergaming.net
E-Mail:	info@peergaming.net
Repository:	https://github.com/autarc/peergaming

PeerGaming – Share the Fun

TL;DR: *Creating a Client-Side Multiplayer Gaming Framework for the Web,*
which handles Distributed Logic on Multiple Systems in "Real-Time".
(a Bachelor thesis by Stefan Dühring)

Multiplayer games are great: whether they are traditional ones on boards or digitals at computers – playing with other people increases the fun and also the social experience you get.

As you interact with others, the emotions you developed towards the original content & mechanics of a game are getting less important. The excitement will more likely be defined by the competition and the teamwork with other human players.

In contrast to their physical pendants, it's rather simple for video gamers to find folks with the same interest to setup a match. They just need to contact each other over a virtual interface and get connected via the same network. Nowadays the infrastructure of the Internet will be mainly used for this purpose.

Unfortunately modern video games tend to abuse this scheme. They often require a permanent Internet connection, as the game itself will be simulated on a external server and just displayed on your system.

For security and convenience it's mandatory to handle user authentication as an external service.
On their behalf you can access the persistent data stored with your account and compete with others.

But is this really necessary for the majority of PC games ? Not just action titles, but casuals as well ?

Why can't the game itself run on your computer directly – just exchanging the altered data ?

Not only as a controller or rendering screen, but in a completely decentralized manner ?

Answer:

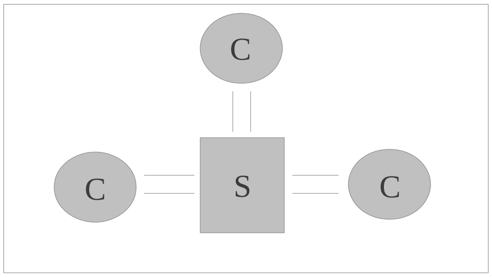
> simplified - It's complicated to create real-time distributed applications.
> detailed - *TBA (a part of my thesis will analyze this situation and the challenges)*

With *PeerGaming* I like to encourage more developers to create browser based multiplayer games using the Peer-to-Peer (P2P) approach. It's a front-end gaming framework which provides guidance and deals with common issues regarding the architecture.

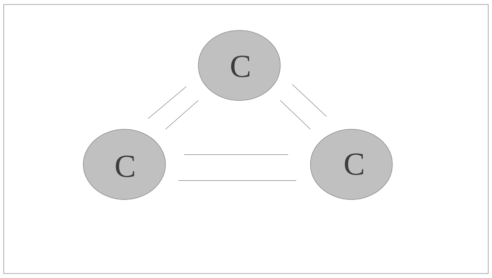
To ease the adaptation, existing games & engines can be used with the *PeerGaming* meta-framework – since it will be agnostic and just provide an API to handle the distributed data on multiple clients.

Besides the documentation, the thesis will also include a test game which demonstrates the capabilities.

Multiplayer Gaming Architectures



a - Client-Server



b - Peer-To-Peer

a.) Client-Server Model

The local systems are clients which receives data from the server and shows the current state of the game. As a player sends input to his client, it will first be handled by the centralized server. Afterwards the client will be informed like all the other connections.

b.) Peer-to-Peer Connections

Each system is a client which is directly connected to the other clients in the network. After the initial matchmaking through a server, they share the data with each other and handle the logic by themselves (without an external server).

There are different networking models using the P2P approach – starting by synchronized simulations, over to the usage of distributed hash tables. Following aspects should always be considered:

Hopes	Risks
+ faster data transfer	- less control of the data flow (temporary / persistent / assets)
+ user keeps control by themself (local)	- complicated, additional problems can occur (e.g. reconnection)
+ less maintenance costs for servers	- secure data handling on the client is difficult (!= cheating)

Traditional environments, which already could access the network directly, faced this agony of choice years ago - but just recently it's possible to think about P2P gaming in a new context: the browser.

The W3C worked with different browser vendors and defined a new standard called *WebRTC*, a set of APIs which allow real-time communication using your browser. Besides the environmental input from your camera or microphone - it also allows to establish PeerConnections to other systems and exchange media with them over binary streams. The newest specification includes DataChannels, which uses these streams to send arbitrary data from one peer to another.

WebRTC is a new front in the long war for an open and unencumbered web.
— Brendan Eich, inventor of JavaScript

Browser Evolution

In the debate of a “Client-Server vs. P2P”-gaming-architecture fulfills the browser a special role:

- there are different environments used for front-/back-end code execution
- accessing the systems hardware is restricted through a sandbox
- most browsers got build-in tools to inspect the client side code

The last point is for most developers, who are interested in P2P gaming, the biggest problem.
As the clients handle the logic by themselves without a server, it's quite easy to manipulate the data.
Since this would affect the game for all players, cheating on the local system needs to be prevented.

In contrast to standalone engines, running the code in the browser even simplifies such editing – you just need to hit F12 to inspect and change the current running code. The same goes for media data and assets, which can hardly be protected on an open platform like the web.

Nevertheless we got benefits from the browser as well, like using existing protocols and open ports.
While “native applications” often get problems with these - the IP translation can be leveraged.

Moving all gaming aspects into the browser makes it possible to focus on one environment:
instead of writing front-end code in JavaScript and the back-end in another language like PHP, Ruby or Python – no additional setup is required. This doesn't just ease the maintenance later on in production, but already simplifies the first steps for new developers.

Keywords: Open Web, Games, Multiplayer, Real-Time, Client, P2P, HTML5, WebRTC, DataChannels, Streams, Distributed Logic, Synchronized States, JavaScript

Reference excerpts

Media	Description
Professional HTML5 Mobile Game Development	current literature about HTML5 gaming
W3C – WebRTC draft	the official WebRTC specs
WebRTC	information site about WebRTC
WebP2P	the P2P web community
"What every programmer needs to know about game networking”	article about game networking
"Implementation of a Peer-to-Peer Multiplayer Game with Realtime Requirements”	Master-Thesis by Max Lehn,TU Darmstadt