

## Mémoire de Veille Technologique

En tenant compte de l'évolution actuelle du marché économique des jeux et des technologies des navigateurs, quels moyens seront utilisés pour les prochains jeux vidéo par navigateur ?  
Conviendront-ils aux joueurs ?

Le Cloud Gaming est-il une technologie d'espoir ?



## I. Table des matières

---

I. Table des matières.....	2
II. Abstract.....	3
III. Introduction .....	5
IV. Etat de l'art .....	6
1) L'analyse de notre matériel .....	6
a) Nos plateformes.....	6
b) Le marché des navigateurs.....	7
2) L'évolution des technologies du jeu par navigateur.....	10
a) Le cimetière du web .....	10
b) Nos technologies actuelles.....	12
c) Les possibilités de l'avenir .....	17
3) La décomposition des modèles économiques .....	18
a) Le Free-To-Play.....	18
b) Le Pay-To-Play.....	20
c) Le Pay-To-Win .....	21
4) Le souhait des joueurs.....	22
a) Techniquement .....	22
b) Economiquement .....	25
V. Critique personnelle.....	27
1) Description.....	27
2) Historique .....	29
3) Contraintes et explication.....	30
a) Latence contrôleur Input et retour vidéo .....	30
b) Coût.....	34
c) Futur du Cloud Gaming .....	35
VI. Conclusion.....	36
VII. Bibliographie / Webographie.....	37

## II. Abstract

---

The video games market is, in general, large and varied, by its type, its use, its platform, its procurement, the additivity, quality

More and more people are playing video games. At home or at work, even in public transports. How can we resume a game we paused during a journey and keep it in its best performances?

A Smartphone software won't be suitable at home, on a computer or even at work.

A software which is as available on computers as on tablets or smartphone? A browser game!

Browsers games are really suitable for any up-to-date browser and so for each platform. However, we will see that not all the platforms are equals. Their screen sizes or their power might be an issue in few cases. In fact, compatibility between user's interfaces can impact the overall display and more when we need to create 3D scenes.

Anyway, browser games have not always displayed 3D scenes. We did write a chronology about web technologies which are proper to games.

It begins with the HTML, CGI and Flash with the first browser game: Earth2025. Released in 1997.

Next, we have actual technologies such as Javascript, HTML5, Node.js and Unity. In this paragraph, we explain what we are able to create with these.

At last, we got the future technologies and developing technologies like the Cloud and connected devices.

Games are not all alike. In fact we can separate them in 3 categories that represent their economic model.

At first, the popular "Free-To-Play", then we have the pretty unloved "Pay-To-Win" and last but not least "Pay-To-Play".

Just as their name suggests, one of them is free and the 2 others are paying.

In order to have an accurate eye about technologies and economic models, we will consider what the players wish and then explain why they want to play some well-developed games regardless the performances of their devices or computers.

Moreover, we will see that they are fond of "Free-To-Plays" with online stores.

We will conclude on the fact that future games will be "connected" by multiplayer mods and thanks to social networks.

A technology that is close to my heart and which will surely become necessary in the near future, the Cloud Gaming, we will analyze it in detail and why it will become inevitable.

Browser games will be much more good looking and powerful (physically). They may require heavier add-ons to run computers despite the smartphones and tablets, luckily, softwares will replace these.

### III. Introduction

---

Le marché du jeu vidéo s'est beaucoup développé depuis l'arrivée de l'ancêtre Pong. On joue aujourd'hui à des jeux nécessitant de plus en plus de ressources de la part de nos machines.

Il existe dans le marché actuel des jeux vidéo une très vaste partie, pourtant peu mise en avant par rapport à la plupart des autres jeux, qui concerne les Browser Games.

Le jeu par navigateur, ou Browser Game en anglais, se joue par définition dans un navigateur Internet.

Depuis maintenant une dizaine d'années, le secteur des jeux vidéo par navigateur est en constante expansion grâce aux technologies composant les différents navigateurs qui ont permis de repousser de plus en plus les limites techniques.

En observant l'évolution des technologies dont nous disposons à l'époque puis en comparant les parts de marché selon les critères d'aujourd'hui, on peut remarquer une nette évolution du marché des navigateurs et des nouvelles technologies.

Une nouvelle mouture de langage de description des pages web est arrivée en apportant son lot de fonctionnalités clefs.

C'est en effet grâce aux différentes évolutions des standards ainsi que celles des technologies que les jeux par navigateurs ont pu subir un tel essor au cours de ces dernières années.

En tenant compte de l'évolution actuelle du marché économique des jeux et des technologies des navigateurs, quels moyens seront utilisés pour les prochains jeux vidéo par navigateur ? Convieront-ils aux joueurs ?

## IV. Etat de l'art

---

### 1) L'analyse de notre matériel

#### a) Nos plateformes

En premier lieu, nous allons définir le terme « plateforme ». Qu'est ce qui est considéré comme plateforme ?

Il s'agit de nos matériels, sur lesquels sont implémentés un système d'exploitation permettant l'utilisation de logiciels, et donc de jeux. On utilise le terme plateforme pour les ordinateurs, les smartphones, les tablettes portables ou même les serveurs web.

Tous peuvent être utiles dans le cadre des jeux-vidéos, avec un rôle ou une ampleur différente.

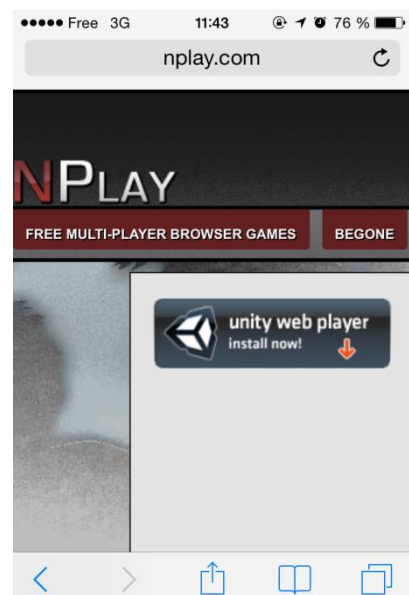
Les ordinateurs, de par leur puissance de processeur et surtout de la carte graphique (même les processeurs graphiques intégrés au chipset de la carte mère), réussissent à faire fonctionner la plupart des logiciels, souvent optimisés pour toutes les configurations, tous les logiciels y compris nos jeux et surtout dans notre cas :

Les navigateurs web, avec une puissance de RAM largement suffisante et un accès à l'accélération graphique, c'est-à-dire le calcul par la carte graphique.

Mise à part la taille de l'écran, il n'y a pas de grandes différences entre smartphones et tablettes portables (même les processeurs tournent tous entre 1.4 Ghz et 2.5 Ghz).

Nous allons donc mettre les deux types de plateforme dans le même panier.

Leur puissance permet de faire aisément fonctionner la plupart des pages dans un navigateur, cependant il y a un réel manque de puissance pour ceux-ci. Ceux-ci ne pourront pas, à l'heure actuelle, utiliser l'accélération matérielle ou des plugins nécessaires couramment en 2015 comme Unity web Player, Java, Battlelog, ou encore Flash et n'arriveront donc pas à lancer certains jeux.



Site de Begone, demandant le plugin Unity Web Player depuis un iPhone  
<http://www.nplay.com/>

Pour ce qui est de l'exécution de jeux en HTML5 (Technologie d'affichage d'une page web, voir 2.b), il est nécessaire que les développeurs adaptent l'aspect graphique pour des différentes résolutions, tandis que d'autres vont directement recréer leurs jeux sous forme d'application par exemple en changeant le modèle économique de gratuit à payant).

Pour finir, parlons du serveur web.

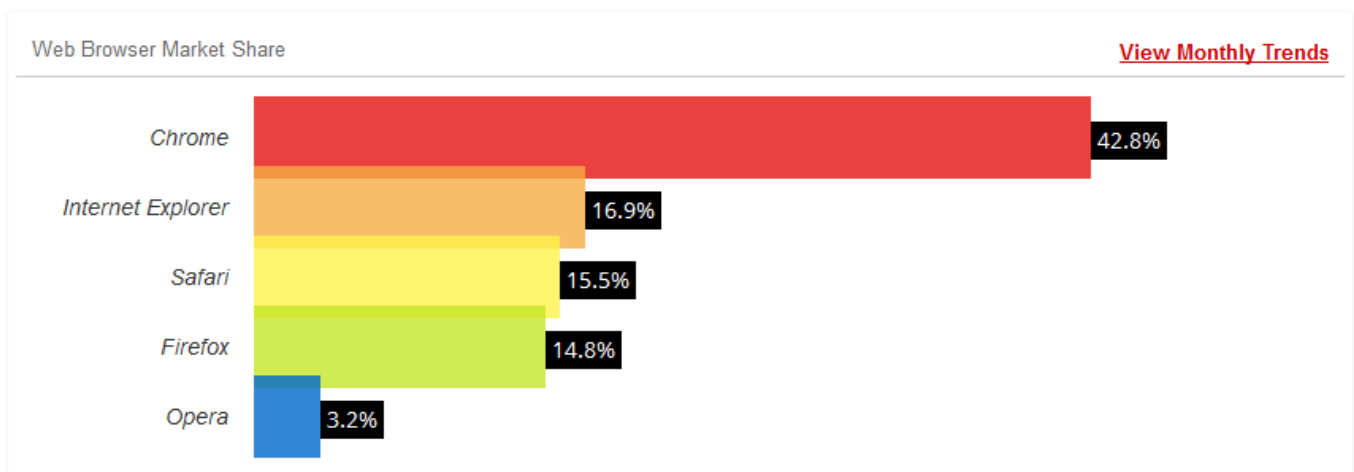
Si les ordinateurs et smartphone/tablette jouent un rôle de Client (exécutant le code permettant de jouer) le serveur, lui va jouer son rôle éponyme.

Le rôle de serveur dans le monde des jeux-vidéo en général sera de gérer les connexions client, la base de données, le système de jeu, en résumé : toute la mécanique du jeu.

Pour la spécialisation de serveur de jeux web, il pourra être configuré et même programmé de façon différente, nous verrons plus tard qu'il pourra même utiliser JavaScript comme langage pour son moteur et il ne faut pas oublier que sa partie Hardware (la partie physique de la machine) joue un rôle majeur dans la bonne fluidité du jeu.

## b) Le marché des navigateurs

Actuellement, le marché se divise en 5 grands acteurs : Google Chrome, Internet Explorer, Mozilla Firefox, Safari et Opéra. Il y'en a bien sur d'autres mais qui ne représentent seulement, parts de marché fusionnées, 7% du marché.



Opéra et Firefox sont assez équivalents en termes de puissance, de vitesse d'affichage et de ressources utilisées. Par rapport à la totalité des navigateurs ils sont des bons compromis et leur utilisation sera souvent bonne pour les jeux par navigateurs.

La réelle différence que l'on pourra noter est que sur Firefox la création de plugin est plus simple, ce qui souligne que les développeurs de jeux par navigateurs préféreront créer un plugin pour Firefox que pour Opéra, rendant ce dernier inutile pour le jeu.

Safari ne disposant que de 15% du marché. il est présent sur tous sur tous les systèmes. Parts de marché des navigateurs mondial en février 2015 navigateur peut aussi être <http://www.w3counter.com/globalstats.php>, catalogué comme navigateur pro-Apple, il est trop souvent oublié comme choix, même sous Windows, alors que ses performances de vitesse et de consommation sont exemplaires.

Il faut savoir, et garder à l'esprit, qu'Internet Explorer est de loin le navigateur qui exécute le plus rapidement le JavaScript (une parmi les plusieurs technologies des jeux par navigateur) et que malgré sa réputation douteuse due aux ancienne versions, la dernière version affiche parfaitement bien la plupart du contenu HTML5 tout en ne consommant quasiment aucune ressources.

Le succès de Chrome est présentement dû à sa gigantesque campagne de « publicité » (la demande d'installation de chrome à chaque navigation de google.com (rappel : 360 millions de recherches par heures)), au détriment d'Internet Explorer et de Firefox qui ont perdus des parts depuis 2008. Certes il est rapide et fiable, et certains sites ou jeux web ne sont compatibles que sur Chrome (Notamment ceux utilisant l'API FileSystem (API : Interface de programmation), qui permet de télécharger en JavaScript les données du jeu dans un espace local) mais il est très gourmand, 26 fois plus qu'Internet explorer, ou 45 fois plus qu'Opéra, il ne sera pas conseiller sur les plateformes avec peu de mémoire RAM. Sur Android (smartphone ou tablette) on essayera de privilégier plutôt CM Browser ou Firefox, qui sont tous deux équivalent en respect de l'affichage mais plus rapide que Chrome en exécution.

Navigateur	Ressources	Speed Battle	Sunspider	Peacekeeper
Chrome	1066 MB	639.94	961.5	3895
Firefox	177 MB	1279.04	162.7	4165
IE 11	41 MB	664.41	<b>100.0</b>	2807
Opera	200 MB	959.86	183.0	<b>4315</b>
Safari	<b>24 MB</b>	<b>317.12</b>	203	2522

Tableau comparatif des navigateurs (Septembre 2014)  
<http://www.web-eau.net/blog/comparatif-les-navigateurs-internet>

Nous l'avons bien vu, les navigateurs les plus utilisés sont relativement différent les uns des autres, mais en général les 5 protagonistes ne devraient pas avoir de souci dans le cadre des jeux par navigateurs. Ils intègrent tous Flash player, Unity web player, Java, HTML5, JavaScript qui sont les technologies les plus utilisées pour les browser-games.

## 2) L'évolution des technologies du jeu par navigateur

### a) Le cimetière du web

Depuis l'invention de l'internet dit « public » (encore avant nous parlions de l'internet de l'armée ou des centres de recherches) surnommé le Web, développé en 1989 grâce au système Hypertexte de Tim Berners-Lee. Ce système de balises permettait de mettre en forme des pages web et devait encore évoluer pour pouvoir convenir à l'envie des développeurs pour faire les premiers jeux sur navigateurs.

Il aura fallu donc 8 ans, l'ajout du css (feuille de style pour les pages HTML), l'évolution du balisage pour donner HTML 3.2 (Hypertext Markup Language), l'ajout d'image et les navigateurs Netscape (2.0 et 3.0) ainsi qu'Internet Explorer (3.0) pour qu'en 1997, le premier Game-browser existe. Earth 2025, développé par Mehul Patel. Ce jeu comme ceux qui ont suivis après on tous utilisés des techniques du DHTML « Dynamic HTML », c'est-à-dire les premières techniques de dynamisme des pages, méthode possible grâce aux CGI.

Ce sont des programmes toujours d'actualités qui vont générer une page web avant de l'envoyer, en la modifiant le plus souvent (comme le PHP par exemple). Puis, en 1998 viendra Utopia, qui a eu un plus grand succès mais qui reste sur les mêmes technologies de CGI, les « Server Side Includes », qui permettait de découper ses pages en plusieurs bout de page, mais cette technologie a depuis longtemps été remplacée, au même titre que les servlet java qui ne dispose maintenant que de 0.5% du web.

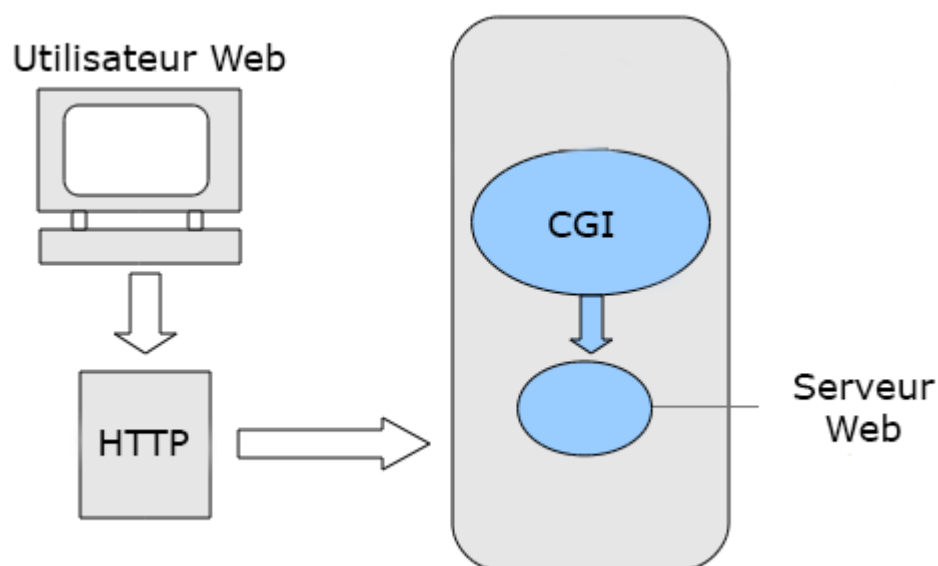


Schéma de l'action du CGI sur une requête

Basé sur : [http://fr.wikipedia.org/wiki/Common\\_Gateway\\_Interface](http://fr.wikipedia.org/wiki/Common_Gateway_Interface)

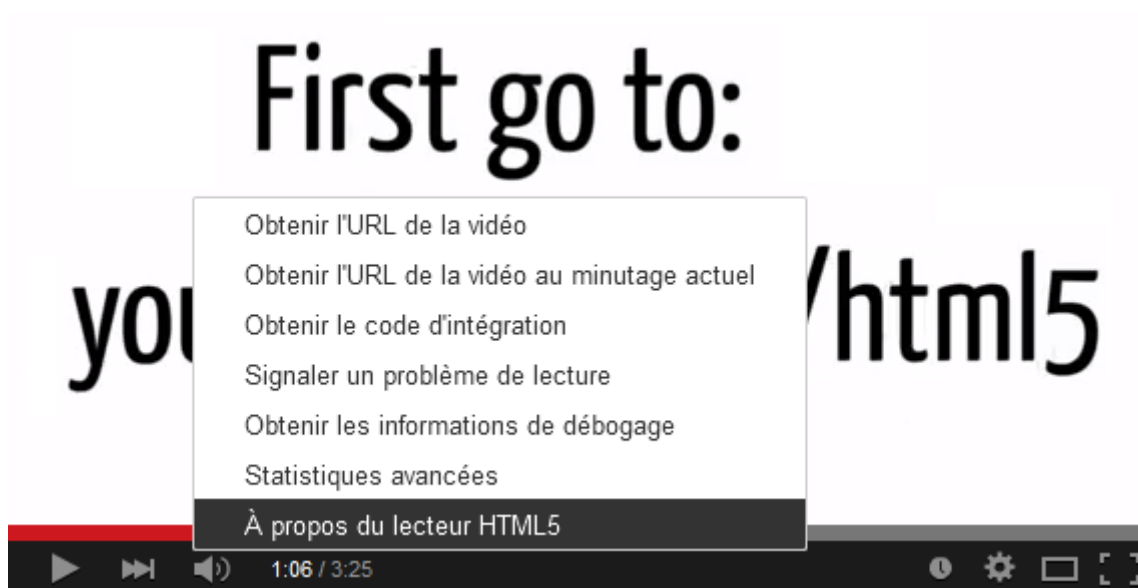
Au fil du temps PHP se concrétisait, utilisé comme CGI principal, en 2000 sortait sa version 4.0.0, il intégrait des fonctions de génération d'image (travail réalisé par Rasmus et Jouni) ce qui lui a valu son utilisation pour de nombreux jeux où une génération dynamique d'image devenait un grand plus graphique et interactif.

Cette méthode était révolutionnaire, notamment avec les gros jeux d'aventure comme le populaire jeu Néopet après 2001, mais nous verrons plus tard que les technologies actuelles ont réussi à remplacer les fonctions image de PHP.

Au même moment sortait l'énorme version 5 de Flash, développé par Macromedia avant le rachat par Adobe. Flash est un plugin (logiciel greffable) pour tous les navigateurs qui permet la lecture de fichier interactif, sans rechargement de page et avec des animations et du son. C'est une technologie qui allait très bien aux jeux par navigateurs. En 2002 Flash est déjà sur 92% des plateformes navigantes sur le Web.

Avant la sortie de Flash Server 2.0 en 2005 les jeux flash étaient souvent des jeux solo, les développeurs souhaitant faire des jeux multi-joueurs partaient plutôt sur d'autres technologies (Shockwave pour rester sur navigateur, sinon directement avec des clients). Cependant des jeux multi-joueurs avec Flash existaient (et existe toujours) : Habbo Hotel, Dofus.

Cependant le Flash est entrain de souffrir de la grande évolution du HTML dans sa 5eme version. Tout ce que Flash peut faire, HTML5 sait le faire dans le domaine du jeu vidéo comme n'importe quelle autre domaine du web, même le site le plus visité qui utilise Flash (YouTube) dispose déjà de son lecteur en HTML5 qui est devenu par défaut sur les dernières versions des navigateurs depuis février 2015.



Vidéo tutorial d'activation du HTML5 sur YouTube  
<https://www.youtube.com/html5?hl=fr>

## b) Nos technologies actuelles

La nouvelle technologie en pleine expansion, c'est le HTML5, suite inévitable du HTML 3 et 4, il évolue et continuera d'évoluer grâce au groupe de travail WHATWG (Web HyperText Application Technology Working Group, un collaboratif des différents développeurs de navigateurs web pour développer le web en général).

Il intègre de nombreuses API ou balises comme la balise Canvas, traduit par 'toile à dessiner', c'est une balise qui permet des rendus graphique dynamique bitmap 2D voire même 3D via du JavaScript.

La puissance énorme du JavaScript depuis V8 (moteur JavaScript libre développé par Google) ainsi que la facilité de langage (qui requiert quand même des notions en programmation objet) rend le développement de jeu relativement facile, rapide et beau.

Le couple JavaScript/Canvas permet aisément de dépasser les 60 fps (frames per seconds, soit le taux de rafraîchissement de l'image du canvas par seconde, 60 étant le standard de bonne fluidité) et la prise en compte des entrées clavier/souris. Cependant ces entrées ne sont pas compatibles avec les smartphone, sauf grâce à une librairie JavaScript (TouchPunch.js) née il y a quelques mois, donc assez récente et pas encore assez exploitée.

Grâce à ce duo, pour le moment nous ne pouvons créer que des jeux 2D (ou 3D avec quelques autres librairies JS) solo. Il suffit de rajouter une technologie, « WebSocket » (API Javascript permettant un échange bilatéral synchrone entre client et serveur), pour que notre jeu passe d'un jeu solo à un jeu multi-joueurs.

La librairie JavaScript la plus connue et utilisée de WebSocket est Socket.io qui permet donc des connexions de client à serveur, très rapide de fonctionnement et rapide à prendre en main pour les développeurs, cette librairie requiert quand même que le serveur distant, de jeu, soit écrit avec Node.js (nous y reviendrons).

Un exemple de moteur de jeu en 50 lignes de code avec JavaScript et canvas:

<http://codepen.io/ericlathrop/pen/FCGwq>



```
HTML
1 <canvas id="game" width="640" height="480">
  </canvas>

JS
1 var canvas = document.getElementById('game');
2 var context = canvas.getContext("2d");
3 var lastTimestamp = -1;
4 function mainLoop(timestamp) {
5   if (lastTimestamp === -1) {lastTimestamp = timestamp;}
6   var timeDiff = timestamp - lastTimestamp;
7   lastTimestamp = timestamp;
8   simulation(timeDiff);
9   draw();
10  window.requestAnimationFrame(mainLoop);
11 }
12 window.requestAnimationFrame(mainLoop);
13 var player = { x: 50, y: 50, width: 50, height: 50, xaccel: 0, yaccel: 0 };
14 function simulation(timeDiffMillis) {
15   var elapsedSec = timeDiffMillis / 100;
16   if (keys["left"]) {player.x -= elapsedSec * 70;}
17   if (keys["right"]) {player.x += elapsedSec * 70;}
18   if (keys["space"]) {player.yaccel = -100;}
19   var gravityAccel = 50;
20   player.yaccel += elapsedSec * gravityAccel;
21   player.y += elapsedSec * player.yaccel;
22   if (player.y > 400) {player.y = 400;}
23 }
24 function draw() {
25   context.clearRect(0, 0, canvas.width, canvas.height);
26   context.fillStyle = "#00ff00";
27   context.fillRect(0, 450, canvas.width, 30);
28   context.fillStyle = "#ff0000";
29   context.fillRect(player.x, player.y, player.width, player.height);
30 }
31 var keys = {};
32 var keyMap = {32: "space", 37: "left", 38: "up", 39: "right", 40: "down"};
33 for (var kc in keyMap) {
34   keys[keyMap[kc]] = false;
35 }
36 window.onkeydown = function(event) {
37   // console.log("keydown " + event.keyCode);
38   if (keyMap.hasOwnProperty(event.keyCode)) {
39     keys[keyMap[event.keyCode]] = true;
40   }
41 }
42 window.onkeyup = function(event) {
43   // console.log("keyup " + event.keyCode);
44   if (keyMap.hasOwnProperty(event.keyCode)) {
45     keys[keyMap[event.keyCode]] = false;
46   }
47 }
```

Le son fait partie intégrante du jeux-vidéos et sur le web c'est aussi possible ! Une interface de programmation, Web Audio API, permet de gérer des sons, de les modifier et de les jouer sur le navigateur web, il est également possible dans certains jeux 3D web de faire des rendus réaliste selon la position et la direction de la caméra du joueur et des ambiances « temps réel », ce qui est extrêmement important pour l'immersion dans un jeu, et donc la garantie de revenir rejouer.

Etude sur l'importance du son dans les jeux-vidéo

<http://www.ina-expert.com/e-dossier-de-l-audiovisuel-le-son-dans-tous-ses-etats/du-son-et-des-jeux-video.html>

Le catalogue d'API et de bibliothèques JavaScript est grand et varié dans le domaine du jeu vidéo. Voyons les différentes utilités que nous avons :

- Les Social API : Interface de programmation relativement facile à prendre en main, permettant d'intégrer dans son jeu des liaisons aux réseaux sociaux (Facebook, Twitter..) ou encore de voir directement ses amis dans le jeu sans avoir à les ajouter manuellement.
- La Full Screen API : Permettant de disposer de son jeu en plein écran, cela peut donner une immersion très bonne, ajoutée à la gestion des sons et ambiances de l'API sonore précédemment évoquée.
- L'API FileSystem : Disponible uniquement sous Chrome et Chrome-for-Android actuellement, cette API permet le téléchargement en mode sandbox (Littéralement Bac-à-sable, il s'agit d'une zone limitée et confinée) des différents fichiers de ressources (texture, modèles, sons, script et exécutables) du jeu, ce qui permet par exemple de jouer depuis 2010 à Quake II en ligne sur navigateur : <http://quake2playn.appspot.com>
- WebGL : sûrement une des API les plus puissantes, elle permet de modifier les méthodes de bases de dessins dans le Canvas par une bibliothèque de méthodes de dessin 2D et surtout 3D. De plus, WebGL bénéficie de l'accélération matérielle en faisant appel au pilote OpenGL (« Open graphic Library », une bibliothèque de fonctions graphiques bas niveau, généralement installée de base avec tous les pilotes graphiques de plateforme) et permet donc un affichage 3D très poussé et fluide. Cependant la difficulté de développement est plus grande et de nombreux jeux préfèrent rester avec les méthodes de dessins Canvas classique.

En restant sur le JavaScript, mais en changeant complètement de plateforme et même de rôle : le Node.JS. Comme la plupart des autres bibliothèques JavaScript son nom finit par ".JS " mais celle-ci est utilisée côté non pas client (navigateur du joueur) mais côté serveur (le système et serveur du jeu). Bien que cette technologie soit naissante et complexe, elle est déjà utilisée sur beaucoup de serveurs de jeux web, c'est un programme développé sous le moteur V8 de JavaScript, il permet une gestion asynchrone et non-bloquante des requêtes, cela permet de traiter de très nombreuses requêtes en même temps avec une gestion des processeurs multi-cœurs optimale.

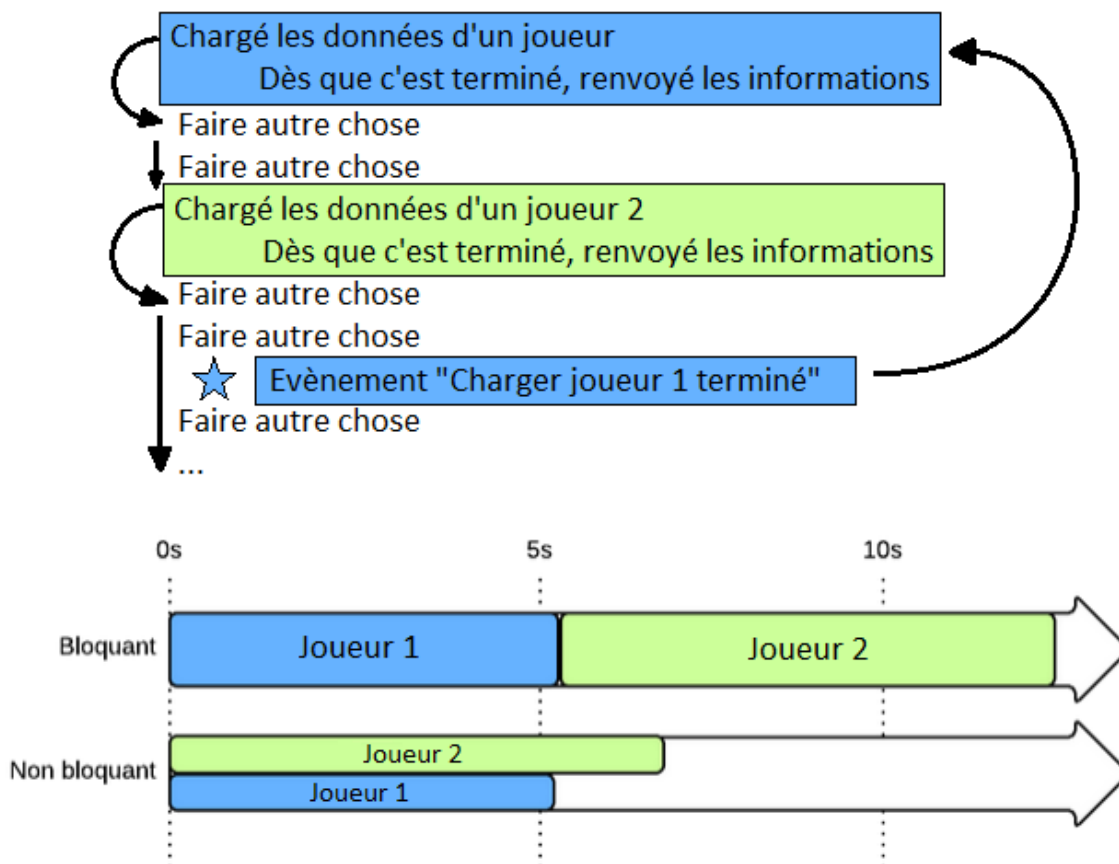


Schéma du système de tâches non bloquantes (asynchrones) de Node.Js

Basé sur :

<http://openclassrooms.com/courses/des-applications-ultra-rapides-avec-node-js/node-js-mais-a-quoi-ca-sert>

De plus, comme dit précédemment, l'API Socket.io, l'interface de WebSocket, a besoin de communiquer avec un serveur Node.Js.

Nous en avons fini avec JavaScript, nous avons vu qu'il est très utilisé côté client comme serveur, mais ce n'est pas la seule technologie côté serveur disponible pour les jeux par navigateur :

- Les taches CRON (ou taches planifiées sous Windows server) : Ce sont des tâches qui permettent, à une date et heure donnée, périodiquement ou non, d'exécuter un programme. Comme exemple pour les serveurs de jeux, un gain de bonus journalier peut être lancé chaque jour à 01h00 ou encore une sauvegarde journalière du village et de tous les joueurs à 04 :00.
- IndexedDB, cas spécial pour les jeux non-multi-joueurs, indexedDB permet d'intégrer nativement au navigateur une base de données d'objets organisée par index. Son utilisation est très rapide, grâce aux index, et utile dans le cas où un jeu doit continuer à vivre sans connexion, quelqu'un souhaitant jouer pendant 4 heures pendant un vol d'avion, pourras charger tout le jeu en local et ensuite utiliser IndexedDB pour sauver l'état de chaque objet du jeu.
- Le Cloud permet en général une meilleure sécurité des données en cas de multi-joueurs, et permet une sauvegarde de notre partie en cas de jeu solo, avant nous devions sauvegarder notre partie en local comme sur console, maintenant le cloud permet de jouer sur son ordinateur puis de reprendre sa partie 1 heure après sur son téléphone dans le train par exemple.

Pour développer des jeux par navigateurs, on peut pour plus de facilité, utiliser des moteurs de jeux, la plupart peuvent générer les jeux en JavaScript ou en HTML5. Ce n'est pas le cas pour le moteur Unity, de plus en plus de jeux sortent sous ce moteur, qui dispose d'une version gratuite, et qui est bien riche en documentation. Unity a choisi d'utiliser lui, comme Flash, un addon « Web Player », qui nécessite l'installation préalable d'Unity Web Player qui contient le moteur exécutable du jeu que l'addon exécute au chargement du jeu.

Cette technologie permet la simplicité de développement mais aussi et surtout un rendu 3D (ou 2D) fluide, et beau.

Cependant le Web Player limite les fonctionnalités à celle de base du moteur, par exemple un moteur sonore ajouté à notre projet qui n'est interne à Unity ne pourra pas être utilisé dans notre jeu via un player.

### c) Les possibilités de l'avenir

Les technologies actuelles évolueront dans le futur, Unity aura plus de fonctionnalités, le HTML (qui va se renommer en HTML Living Standard) va continuer son évolution et sa compatibilité.



Logo du HTML Living Standard  
<http://whatwg.org/>

Cependant, à côté de ces évolutions, les « futurologues » du web parlent déjà de Web3.0, il s'agirait de « l'internet des objets », un internet où un très grand nombre d'objets connectés ou de capteurs seraient compatibles aux sites, jeux via des API, des plateformes.

Avec le développement du Cloud et des fermes de calcul (la location d'un grand nombre de machines reliées, utilisant la pleine puissance de calcul des cartes graphiques, travaillant sur une même tâche, comme le rendu d'une image ou encore la compilation d'un jeu) les jeux par navigateurs pourraient devenir colossaux tant au niveau de la qualité graphique qu'algorithmique (calcul de physique). Et cela commence à se développer:

- Calcul algorithmique des états des objets du jeu sur des ordinateurs du Cloud du serveur.
- Calcul du rendu graphique depuis les fermes de calculs qui ont reçu les données des objets.
- Affichage simple du rendu reçu depuis le serveur du jeu.

Ce système qu'essaye de promouvoir Amazon, permettrait de faire fonctionner de gros jeux sur des navigateurs web de plateformes qui ne disposeraient même pas de carte graphique.

Si Node.js (et Socket IO) vient à devenir encore plus performant qu'il ne l'est déjà, n'importe quel jeu pourrait tourner de façon exemplaire avec une qualité époustouflante.

Cependant, il y a un bémol, la grande consommation de ressources des serveurs du Cloud serait un gros coût pour la société du jeu, ainsi le jeu lui-même aura un prix, pour une rentabilité et un coût de serveur variable, si des jeux de ce genre viennent à exister, ils devraient utiliser une méthode d'abonnement comme rentabilisation.

### 3) La décomposition des modèles économiques

#### a) Le Free-To-Play

Littéralement traduit « gratuit à jouer », le **Free-to-play (F2P)** désigne un jeu qui est accessible de façon totalement libre et gratuite.

Il est important de noter qu'un jeu dit « **F2P** » propose dans la plupart des cas un système de micro paiement dans une boutique qui lui est propre afin d'obtenir un minimum de rentabilité.

La plupart des **F2P** sont des jeux vidéo à accès libre, la tendance économique des jeux massivement multi-joueurs étant l'abonnement, le **F2P** lui se distingue par sa gratuité apparente. Ils sont considérés comme des jeux sans abonnements disposant d'options d'achats.

Ces jeux peuvent également être financés par de la publicité.

On peut citer Ogame (un pionnier du jeu par navigateur édité par Gameforge) dans la dynamique du jeu ayant quelques publicités ainsi qu'une boutique virtuelle proposant divers options d'achats.

Afin de rentabiliser le modèle économique du F2P, le jeu met en place quelques concepts essentiels à son bon déroulement.

#### **La frustration.**

Il est important de noter que dans un F2P, le joueur pourra, dans ses premières minutes de jeu, avoir de nombreuses ressources et jouer sans problème. Il sera au fur et à mesure confronté à une difficulté croissante. On lui proposera alors de nombreux objets de triche permettant de dépasser cette frustration en les achetant.

#### **L'engagement.**

Fortement lié à la routine, le concept d'engagement cherche à faire rentrer le jeu dans les habitudes quotidiennes du joueur. De nombreuses mécaniques liées au temps sont ainsi mises en place, comme par exemple le gain journalier d'avantages, un panel d'objectifs à accomplir ou encore la construction étalée dans le temps d'un bâtiment en deux ou trois étapes.

## Les achats « in-game ».

Aussi appelés « *In-App Purchases* », le financement du jeu F2P vient principalement de tous ces avantages que le joueur peut acheter via des micros transactions.

A noter que l'on peut obtenir toutes sortes d'avantages : objets, boosters, blocage de publicités, des niveaux ou de l'expérience.

## Exemples de jeux par navigateur F2P :

- Ogame
- Battlestar Galactica Online
- Darkensang Online
- Anno Online

## Avantages et inconvénients du modèle F2P.

Avantages	Inconvénients
Pas de prix d'entrée.	Prix des achats <b>in game</b>
Grande communauté	Lassitude entre les temps d'attente (ex : construction)
Bouche à oreille important	Qualité parfois inférieure aux jeux payants

## b) Le Pay-To-Play

À l'inverse du F2P, son principal concurrent est le **Pay-to-play** (P2P) « payer pour jouer » qui lui n'est accessible qu'en échange d'un paiement.

On peut distinguer les jeux ayant un accès payant, c'est-à-dire que l'on achète un droit d'entrée et les jeux comportant un abonnement où le joueur devra payer chaque mois une certaine somme afin d'accéder à son compte.

Pour les P2P, la plupart sont des jeux massivement multi-joueurs. La plupart de ces jeux sont, comme dit plus tôt, par abonnement et disposent également d'une boutique supplémentaire comme celle du Free-To-Play, permettant également d'obtenir des objets bonus.

Le P2P ne nécessite pas particulièrement de publicités pour se financer, l'abonnement étant là pour rentabiliser le modèle.

Le modèle P2P possède également quelques concepts importants.

### L'engagement.

Tout comme le Free-To-Play, le concept d'engagement cherche à faire rentrer le jeu dans les habitudes du joueur. De nombreuses mécaniques liées au temps sont ainsi mises en place, comme par exemple un certain nombre de quêtes à réaliser ainsi que des bonus journaliers.

### Les achats « in-game ».

De la même façon que le F2P, le joueur peut acheter via des transactions directement en jeu. A noter que l'on peut obtenir les mêmes types d'avantages.

### Exemples de jeux par navigateur P2P :

- RuneScape
- AD2460
- Rise of Europe
- Fantasy Rivals

### Avantages et inconvénients du modèle P2P

Avantages	Inconvénients
Qualité des services fournis	Coût(s) mensuel ou à l'achat
Grande communauté	

### c) Le Pay-To-Win

Traduit littéralement en payer pour gagner, le **Pay-To-Win** (P2W) consiste en l'achat d'objets pour avoir un avantage significatif dans le jeu ou par rapport à ses adversaires. À de rares exceptions, tous les jeux Free-To-Play comportent une part de P2W mais elle peut être plus ou moins envahissante. L'argent devient alors un moyen de gagner, dans une mécanique injuste mais addictive.

Voilà pourquoi les boutiques des Free-To-Play «raisonnables» ne proposent aujourd'hui que des personnalisations d'apparence, des familiers, des gadgets cosmétiques ou encore des bonus d'expérience ou d'argent.

De la même façon que le modèle Free-To-Play et Pay-To-Play, le P2W possède un ensemble de concepts plus ou moins semblables aux deux autres.

#### **La frustration.**

Cette notion de frustration, notamment présente dans le Free-To-Play, sera bien évidemment beaucoup moins présente dans le P2W, ou plutôt, elle pourra être largement compensée par la sensation de joie quant à la progression apportée par les achats de la boutique

#### **L'engagement.**

Toujours basé sur «l'addiction» journalière du Free-To-Play, le P2W propose souvent les mêmes mécaniques liées au temps avec des objectifs journaliers à remplir ou encore la réalisation d'objectifs en plusieurs étapes.

#### **Les achats « in-game ».**

Le principal financement du F2P vient de tous ces avantages que le joueur peut acheter via des micros transactions.

Dans le cas d'un **Free-To-Play** orienté vers le P2W, on obtient des joueurs appelés «whales» ou «baleines». Cette désignation représente des joueurs dépensant beaucoup d'argent et qui jouent régulièrement au jeu. Pouvant augmenter jusqu'à des milliers de dollars pour certains joueurs.

#### **Exemples de jeux par navigateur P2W :**

- Candy Crush
- Ikariam
- Ogame
- Battlefield Heroes

#### **Avantages et inconvénients du modèle P2W**

Avantages	Inconvénients
Pas de prix d'entrée.	Achats quasi obligatoires pour progresser
<b>First User Experience</b> souvent exceptionnelle	L'addiction à l'évolution engendre des coûts élevés pour les joueurs.

## 4) Le souhait des joueurs

### a) Techniquement

Actuellement, les joueurs ont tendance à s'attacher aux technologies des browsers games « actuelles » (Voir 2.a, 2.b). En effet, les jeux développés depuis plusieurs années reposent sur des technologies qui vont arriver en fin de vie. Cependant ils satisfont la plupart des joueurs qui jouent en ligne.

Nous avons, à quelques exceptions près, des jeux à affichage statique développés en **Php**, **Javascript** ou encore en **Flash** comme les jeux de simulation ou de gestion en temps réel, ils sont les jeux les plus joués à l'heure actuelle. La simplicité des interfaces ainsi que le fonctionnement pur du jeu de stratégie / simulation sont en parfait accord avec les limites des technologies actuelles.

Généralement, ce genre de jeu requiert de la part du joueur une certaine imagination, par exemple la création d'un système économique propre entre les joueurs tel un taux d'échange entre les différentes ressources ou encore une gestion des alliances et clans dans le but d'évoluer ou de faire des guerres.



Screenshot des ressources de Forges of Empires  
<http://fr.forgeofempires.com/>

Cependant, de nouvelles technologies sont en train de se développer fortement, surtout dans le browser gaming (voir 2.b, 2.c) grâce à celles-ci, les joueurs peuvent commencer à attendre plus des jeux, plus beaux, plus grands.

En observant l'évolution du marché des navigateurs et des technologies, on peut voir que de plus en plus de choses sont et seront possibles. Dans un futur proche, les joueurs pourront imaginer accéder à des transferts de données fluides et très peu consommateurs de ressources pour un affichage performant qui permettra donc la réalisation potentielle de jeux en 3D (comme avec Unity3D par exemple).

On pourra aisément imaginer une simulation de vol, des FPS (Jeux de tir à la première personne) ou des jeux de combats plutôt que des jeux pour la plupart basés sur du tour par tour.

Par ailleurs, on peut aisément imaginer que les navigateurs pourront bientôt intégrer un système de réalité augmentée afin de contrôler son avatar via un casque Oculus ou les futurs SteamVR (Casque réalisé par HTC pour Steam) et FeelReal (simulant odeur et sensations de vent, vibrations, humidité).



Screenshot de GameGlobe par SquareEnix & BigPoint

<http://www.mmohunter.com/gameglobe-a-new-online-game-platform-from-bigpoint-and-square-enix.html>

L'avancée des nouvelles technologies et de l'évolution du marché des navigateurs permettra aux joueurs de profiter de cette nouvelle expérience de jeu comme celle qui a été fournie par Quake Live (FPS sur navigateur) de la société id Software basé sur Quake 3 Arena.

Seul bémol de ce genre de technologies étant que parfois il est nécessaire d'installer certains plugins pour pouvoir jouer, ce qui peut être gênant plus les utilisateurs.

Avec le développement et l'évolution de la nouvelle génération des Smartphones, les navigateurs ne sont plus uniquement basés sur les clients lourds et légers. Avec les services de géolocalisation comme Maps de Google ou encore Openstreetmap sont de plus en plus performants, les jeux développés pour navigateur pourront comprendre une partie de Responsive Design.

Grâce à ces technologies, on peut facilement commencer à envisager des jeux géolocalisés comme une capture de drapeau à grande échelle sur une carte comme la ville de Paris.

Les joueurs ne se repèreront qu'à l'aide de leur Smartphone ainsi que les points clefs comme la base adverse, la leur ainsi que leurs alliés.

Par ailleurs, on peut également imaginer un jeu dans le genre du mode multijoueur d'Assassin's Creed, basé sur le même principe de la géolocalisation. Les différents joueurs auront sur le navigateur de leur téléphone une « cible » à éliminer ainsi qu'un temps limité pour parvenir à terminer la partie.

Les joueurs n'étant que de simples points sur une carte et en sachant que la plupart des personnes circulant dans la rue utilisent des smartphones, le camouflage est parfait. À chaque contrat effectué le serveur ajoute un certain nombre de points au score du joueur.

Le serveur pourra alors trier les meilleurs joueurs dans un classement hebdomadaire.



Screenshot de Zombie, Run !

<http://www.faismoijouer.com/2011/03/09/la-ville-comme-terrain-de-jeux-petit-tour-des-jeux-geolocalises/>

## b) Economiquement

Selon les différents modèles économiques précédemment étudiés, la plupart des joueurs auront tendance à se tourner vers les jeux en Free-To-Play.

En effet, comme présenté plus tôt, c'est le modèle le plus accessible pour tous les joueurs potentiels de par sa gratuité. Chaque personne n'aura besoin que d'une inscription afin de créer un compte pour se connecter et démarrer l'expérience de jeu.

Les différents navigateurs, outre les quelques technologies n'étant pas fonctionnelles sur certains environnements comme Linux, permettent donc un accès au(x) jeu(x) en ligne.

De plus, de par leur gratuité, les jeux basés sur le modèle F2P bénéficient d'un bouche à oreille important dès que la First User Experience (première expérience utilisateur) est concluante, ceux-ci cherchent à progresser efficacement dans le jeu sans rester seuls. C'est d'ailleurs pour cela que beaucoup de jeux Free-To-Play possèdent une communauté très conséquente.

Ce modèle permet souvent de faire découvrir un jeu à des potentiels nouveaux joueurs, comme une version « light » le ferait.

Cependant, cela n'empêche pas les joueurs de s'intéresser de près aux jeux Pay-To-Play car ce modèle économique, contrairement à ce que l'on pourrait penser, n'est pas sans être peu peuplé.

De la même manière que le F2P, le Pay-To-Play possède les mêmes atouts pour les navigateurs, à ceci près que dans la plupart des cas, le service de support est souvent plus rapide et performant grâce aux revenus permettant d'employer des personnes dédiées. De plus, on peut constater que ce modèle permet d'offrir plus de contenu aux joueurs comme des quêtes ou des compétences de personnage plus poussées ainsi qu'un système de mise à jour plus régulier.

Les joueurs ont tendance à chercher ce genre d'à-côté en plus d'une nouvelle expérience de jeu car en effet, dans la plupart des Pay-To-Play, la carte du monde est généralement plus vaste et permet donc de faire varier l'expérience de jeu de l'utilisateur.

Pour finir, il y a un type de joueurs potentiellement amené à dépenser de grosses sommes d'argent tout au long de son expérience de jeu, en effet le développement du Pay-To-Win au sein du « Free-To-Play ». De par des offres de bonus alléchantes et beaucoup de moyens de faire progresser rapidement son avatar contre une somme d'argent.

Il est vrai que beaucoup de joueurs se laissent vite tenter par cette « chance » de pouvoir avoir l'avantage de façon extrêmement rapide sur les autres. Il faut avouer que l'on a tous un jour rêvé de pouvoir gagner facilement face à un autre.

Par ailleurs, il n'y a pas que des désavantages, grâce à ce modèle, le jeu propose une expérience de jeu semblable à un Pay-To-Play comme des mises à jour régulières et relativement importantes afin de fidéliser la population de joueurs en proposant toujours de nouvelles améliorations et corrections de bugs.

## V. Critique personnelle

Les jeux pas navigateurs sont jouables sur les navigateurs de nombreuses plateformes et sont développés dans de nombreux langages différents. Pour tous les jeux par navigateurs qui souhaitent de la 3D et de la physique, le résultat est souvent limité par la puissance du moteur dans le navigateur, ou bien la technologie et les langages associés.

Imaginons que l'on souhaite un jeu 3D vaste et multijoueur, un FPS qui demande de la physique, qui est jouable et adaptable sur plusieurs résolutions du téléphone jusqu'au 1080p en tournant à 60fps, tout cela sur navigateur. Beaucoup pensent que c'est impossible, et effectivement ça l'est si ni oublions la technologie qui va devenir, à mon avis, la révolution des jeux par navigateurs (et peut être des jeux-vidéo en général) : Le Cloud Gaming.

### 1) Description

Le Cloud Gaming (ou « jeu à la demande ») est une technique liant plusieurs technologies permettant de jouer à un jeu-vidéo sans avoir besoin de disposer du contenu en local sur notre plateforme (pc/console/téléphone...).

Cependant il existe 2 types de Cloud Gaming, le premier type ne nous intéresse pas trop, il s'agit du fait de jouer à ses jeux sans avoir besoin de l'installer. Soit le jeu est téléchargé au fur et à mesure que nous jouons (exemple : on change de carte, on la télécharge, car nous n'en avons pas besoin avant, pareil pour les assets graphiques ou sonores) comme le font de nombreux jeux par navigateur (flash...) ou encore le développeur Blizzard qui nous propose de jouer à ses jeux en ayant installé le strict minimum.



2 Screenshots du Launcher et du client WoW (Blizzard) mettant en scène le fait qu'il y a un minimum de fichiers à télécharger, le reste ce faisant en jeu aux moments nécessaires.

Source: Launcher/client [World of Warcraft](#)

Le 2eme type de Cloud Gaming nous intéresse beaucoup plus, il s'agit enfaite du premier concept mais poussé à son extrême. C'est-à-dire qu'il nous propose juste de télécharger une interface client très « light », souvent de quelques Mo qui vas nous servir à la fois de moniteur (pour voir le jeu) et de contrôleur (pour interagir avec le jeu) car le jeu en lui-même seras hébergé sur les serveurs (du développeur) dans le cloud. Ainsi tous les fichiers et tous les calculs processeur et carte graphique vont être géré dans le cloud tandis qu'il ne fera que renvoyé l'affichage (comme du streaming vidéo classique) sauf que notre client nous permet quand même de nous déplacé, de tirer, d'ouvrir des menus, de jouer au final.

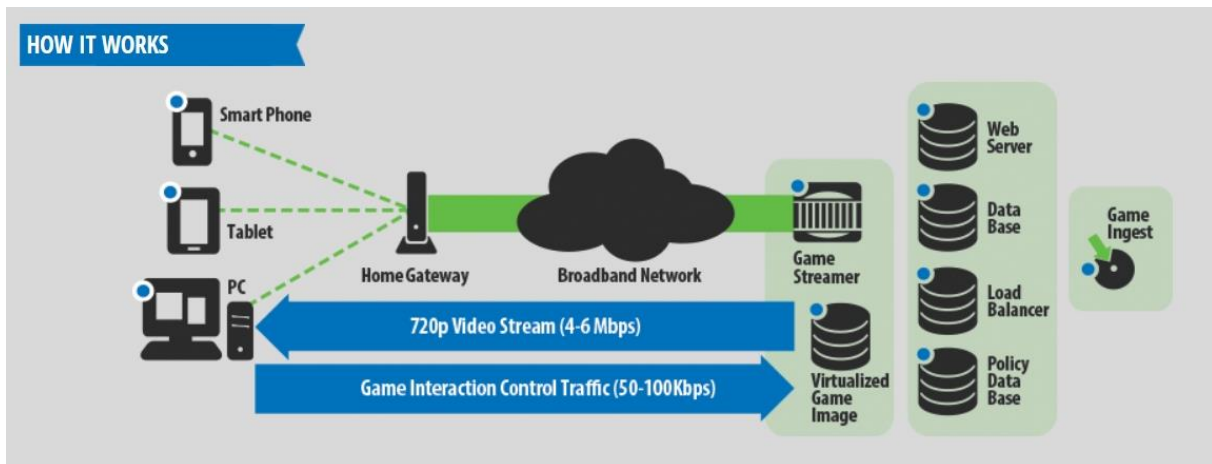


Schéma d'un réseau de Cloud Gaming, à gauche le client sur une plateforme, à droite le serveur de jeu.  
La vidéo streamée du serveur au client et le sens inverse pour l'interaction.

<http://www.onlivespot.com/2013/02/european-games-download-service.html>

## 2) Historique

Il y a peu de grands noms dans l'histoire du Cloud Gaming mais le plus « important » est G-Cluster. Cette société a présentée puis déployée la première le Cloud Gaming en 2000 et 2004 au Japon (2005 en Europe) dans le cadre de développement de jeu (notons Crysis par exemple, il a malheureusement abandonné le cloud en 2007).

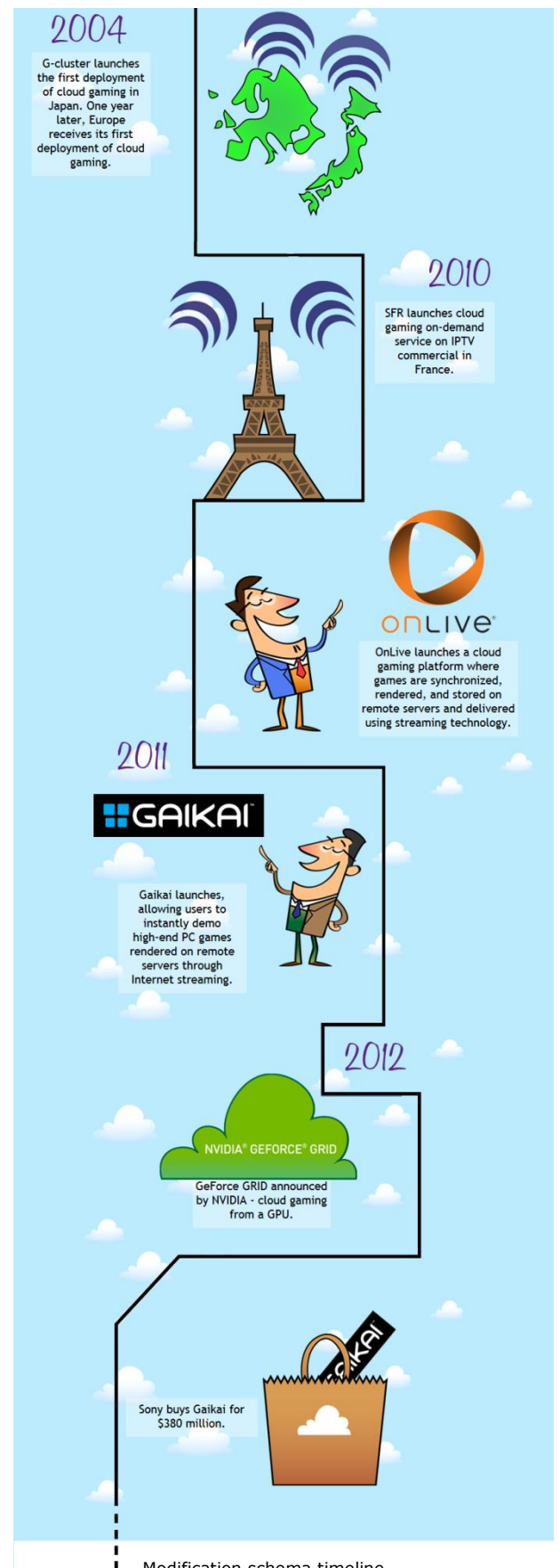
Ensuite en 2010, OnLive annonce son service de jeu à la demande, cette société a élaboré de nombreux brevets sur le sujet des jeux à la demande en streaming mais n'as pas réussi à tenir le coup (en 2012) et a dû fermer en 2015. (Brevets rachetés par Sony)

Entre temps, fin 2010 SFR lance son service de Cloud Gaming sur les IPTV en France, via G-Cluster, c'est la première société au monde à le proposer. (Orange propose la même offre, toujours via G-Cluster, en 2012).

Gaikai, nouveau nom du jeu à la demande, propose en 2011 un panel de jeu démo de très bonne qualité (Mass Effect2, Sims3 ...) par l'intermédiaire d'un navigateur Web en streaming vidéo. Le service est tellement prometteur qu'il se voit racheté en 2012 par Sony (380 millions de dollar).

S'en suit d'autres sociétés qui décident de créer leurs services de Cloud Gaming :

- Nvidia : Nvidia GeForce Grid en 2012
- SquareEnix : Core Online en 2012.
- AMD : Rachete CiiNow en 2012.



Modification schema timeline

<http://www.bigfishgames.com/daily/cloud-gaming/>

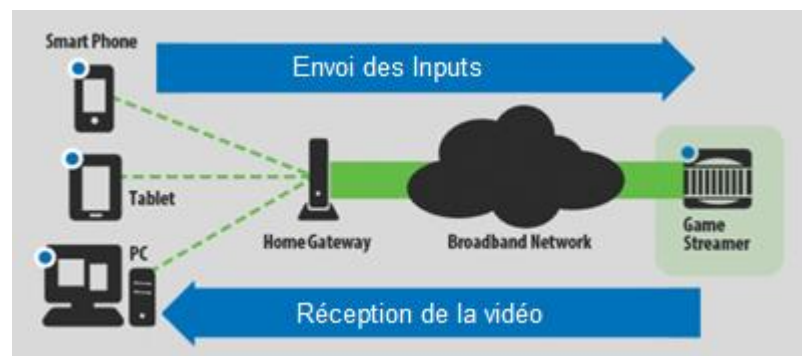
### 3) Contraintes et explication

Le Cloud Gaming peut apporter énormément à la qualité des jeux-vidéos, surtout les jeux par navigateur qui sont plus limités techniquement que les autres (console/pc client lourd). Cependant il y a des contraintes au Cloud Gaming, la connexion est le plus gros frein du Cloud Gaming, il suffit que la connexion du client ou bien du serveur ne soit pas assez puissante pour que l'expérience du joueur se dégrade énormément, mais il y a aussi son coût.

#### a) Latence contrôleur Input et retour vidéo

La plus importante des contraintes du Cloud Gaming est bien sur le temps de latence entre ce que l'on choisit de faire et ce qui se passe à l'écran. Nous allons dissocier la partie interaction et la partie retour vidéo car la seconde partie peut être atténuée tandis que la partie contrôle et interaction elle ne peut pas être modulable en modifiant la taille des paquets émis.

Un schéma simple de fonctionnement du Cloud Gaming est que le client envoie une action au serveur via le client léger, le serveur traite les calculs de l'action (et de celles des autres joueurs), puis produit un rendu graphique et ensuite l'envoi au joueur (en ajoutant des techniques d'encodage et de compression d'image).



Modification du schéma plus haut

En moyenne un joueur de FPS exige un temps de « retour d'action à l'écran » inférieur à 100 ms (millisecondes), cette valeur est différente pour chaque jeu mais reste dans cet environ. C'est là toute la force mais aussi la faiblesse de cette technique qu'est le jeu à la demande, nul besoin de matériel, juste d'une connexion suffisamment puissante.

Le problème est simple, si je choisis dans un jeu de type FPS de tirer sur une cible mouvante en cliquant et qu'au retour image je vois que je n'ai pas touché ma cible c'est qu'il y a eu un trop gros temps de latence entre le fait que j'ai tiré au bon endroit et l'action où le serveur calcul mon tir au moment où il reçoit mon « input » de clic.

La partie Input ne représente que la plus faible demande en bande passante, si comme dans le cas précédent la connexion ne convient pas il y a encore des progrès à faire du côté des FAI (Fournisseurs d'accès internet) car la taille et le nombre des paquets émis pour les actions sont très petit, on parle d'une connexion à 50/100 Kbps.

Pour la suite, partons du fait que les Input sont bons au niveau timing, mais que c'est la vidéo qui n'arrive pas à suivre, c'est saccadé, c'est en retard ... etc.

Là le problème provient soit de la connexion client, soit du code serveur ou de sa puissance.

Pour avoir un affichage dit « de haute qualité » il nous faudrait du 1080p 60fps (le format d'image streaming le plus haut (or 8K/4K)) or le poids de ce flux vidéo est de 4.5 à 9 Mbps, c'est-à-dire une très bonne connexion filaire ou une connexion à fibre optique, ce qui n'est pas le cas de beaucoup de personnes. De ce côté c'est toujours aux FAI et aux pays de changer cela. Cependant souvent dans ce type de cas, les jeux-vidéo à la demande, comme les vidéos streaming, proposes de changer la qualité de la vidéo demandée.

#### 1080p à 60 ips

- Assurez-vous de cocher la case "60 ips" sous l'onglet "Paramètres d'intégration".
- Résolution
  - 1 920 x 1 080
- Débit vidéo
  - Maximum : 9 000 kbit/s
  - **Recommandé : 6 800 kbit/s**
  - Minimum : 4 500 kbit/s

Débites d'évènements en direct

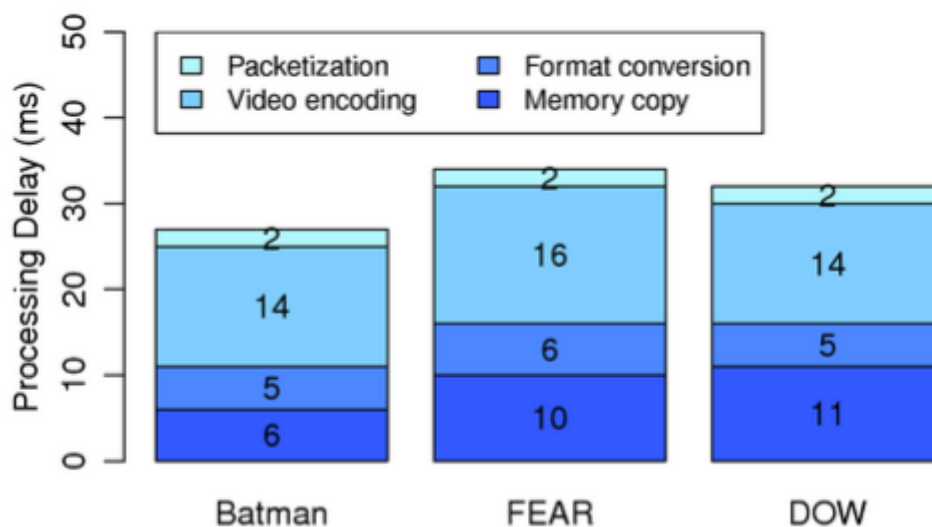
<https://support.google.com/youtube/answer/2853702?hl=fr>

D'accord, nous avons contré le fait que la connexion client ne soit pas assez puissante pour de la haute qualité en réduisant le format, cependant même avec un format plus petit si la connexion est ok mais que la vidéo est toujours saccadé ou mauvais, c'est que c'est le serveur et donc le jeu qui n'est pas assez puissant / bien configuré. Et là un paquet de raisons peuvent être en cause.

Le serveur, à la réception du paquet d'action (ou pas), vas le décompresser, vérifier que l'action est légitime (pour éviter la triche), puis va produire l'action dans le moteur du jeu, sauvegardé en base de donnée, puis va enfin traité un rendu graphique du jeu. C'est là que les latences peuvent se

produire. Le jeu va produire son rendu à la taille et qualité souhaitée par le joueur, plus c'est grand et beau plus c'est long. Mis à part ces choix, le serveur peut être mal configuré et perdre beaucoup de temps sur les 4 parties qu'il doit faire en plus d'un jeu classique :

- Copie mémoire : L'action de copier les données pixels (des bitmaps pouvant peser plusieurs méga-octets) entre le GPU et le tampon mémoire du processus d'envoi.
- La conversion de format : L'action de transformer notre Bitmap de plusieurs Mo, en image beaucoup moins lourde comme le PNG (« Portable Network Graphics, format de fichier beaucoup moins lourd gardant la même qualité).
- L'encodage en vidéo : Pour un gain de fluidité, plusieurs images peuvent être envoyées en même temps, elles sont regroupées en vidéo. Cette action est le temps d'encodage de la vidéo.
- La paquetsation : L'action de découper notre vidéo en plusieurs paquets pour ensuite les envoyés sur le réseau.



Graphique montrant le temps de chaque action en millisecondes pour trois jeux différents (Lego batman, Fear2, Warhammer 40k dow2)

<http://gaminganywhere.org/perf.html>

Nous pouvons voir que nous sommes à environ 35ms pour la partie post processus du jeu. Ajoutons à cela les latences réseau, le temps des calculs du serveur pour le jeu en lui-même, et enfin le cours temps (environ 12 ms) pour que le client léger puisse décompresser les paquets et affiché les images. Cela peu vite dépassé les dits 100ms exigés, et que même si les développeurs se donnent à fond, ils auront du mal à faire descendre la barre des 35ms pour cette partie.

A noter qu'en modifiant les méthodes de compression d'images, le résultat peut-être plus rapide d'un côté (serveur) mais sera forcément plus long de l'autre (client) et cela, bien que positif sur un navigateur web dans notre cas, peut-être plus lent sur un smartphone et donc négatif. Ces résultats, positifs ou négatifs, se mesurent avec la PSNR « Peak Signal to Noise Ratio » qui est une mesure de distorsion utilisée dans la compression d'image analysant les performances à encodé et à décoder des images de même qualité que l'originelle.

Sur ce point-là le Cloud Gaming s'en sort très bien, les temps pour restituer une image compressée à l'exactitude de l'image originelle sont très bons.

## b) Coût

Les coûts pour un jeu par navigateur utilisant la technologie du Cloud Gaming est importante pour les développeurs. Plusieurs raisons rentrent en compte :

- Le coût de développement : Les techniques de développement ne sont pas les mêmes, des postes spécifiques seront nécessaires, il faut développés une partie calcul graphique coté serveur, un domaine méconnu pour le moment. De plus il faut aussi développer une application client. Enfin, le développement d'un jeu en réseau (même pour un jeu solo) dans le cloud requiert beaucoup plus de test et de personnel.
- Le coût de l'infrastructure : Les serveurs où le jeu se situeras demanderas un lourd investissement financier, en effet il faut que le jeu tourne très bien et soit rapide, mais aussi pouvoir géré tous les risques que les jeux en réseau requiert (crash de serveur, incendie, attaque réseau ...etc.).

Enfin forcément ces coûts doivent se répercuté à un moment, et ce seras sur le joueur, soit il devra payer son jeu, soit l'expérience en jeu sera un peu pollué par des pubs ou autre moyen de rentabilisation (une boutique par exemple).

A mon avis, un système d'abonnement conviendrait le mieux aux jeux utilisant cette technologie, les serveurs dans le cloud sont quelque chose de modulable, au niveau de la puissance, de la capacité et de la bande passante (voir l'Auto Scaling des serveurs EC2 d'Amazon, le leader mondial du Cloud), alors plus il y a de joueurs plus les serveurs auront besoin de puissance, et plus cela aura un coût, et à l'inverse, moins de joueurs jouent et donc paye leurs abonnement, moins les serveurs ont besoin d'être grand et donc cher. Ce serait un peu le moyen de paiement « scalable » associé à la technologie.

Les joueurs qui, de nos jours, achètent un ordinateur à 800€ et une console à 400€, n'aurait avec le Cloud Gaming qu'à acheter un ordinateur à 200€ pour faire tourner n'importe quel jeu, cela deviendrait alors peut-être rentable pour les joueurs.

### c) Futur du Cloud Gaming

Le Cloud Gaming pourrait permettre de faire jouer à des jeux exceptionnellement puissants et beaux sur des plateformes ridiculement peu puissantes, nécessitant un coût qui pourrait être élevé, mais serait largement bénéfique qualitativement.

Cependant il demande une quantité de ressource réseau importante, une connexion à plusieurs mégas assez grande pour faire transiter tout le flux vidéo.

La 4G pouvant gérer 60Mbps pratique (150 théorique) serait une éventuelle bonne solution pour les joueurs dans le futur, actuellement (décembre 2014 selon arcep.fr) seul 62% de la France est équipée de 4G. La 5G peut monter à 50Gbps théorique, quant à elle seulement disponible au Japon. Ces deux technologies seraient très bonnes pour le Cloud Gaming, autant que l'augmentation de la couverture de la fibre optique. 3 technologies qui ne sont pas encore bien ou totalement implantées sur les territoires mais qui sont quasi indispensables au jeu à la demande.

La technologie de Cloud Gaming est prête, des jeux sont disponibles, les techniques de réduction de latence via les compressions d'image sont bonnes, elle n'attend que le fait d'être utilisée dans de bonnes conditions et c'est à nos FAI et gouvernements de faire le nécessaire.

Dans un futur proche, et comme l'a déjà tenté Microsoft récemment avec la Xbox One, le Cloud Gaming va s'emparer du monde des jeux et aussi des consoles, les jeux seront jouables sur navigateur car c'est le seul réel « client » communs de toutes les plateformes. Dans ce futur proche le Cloud Gaming sera présent, sera utilisé en masse et prendra de grandes part de marché, à l'instar des composants PC.

## VI. Conclusion

D'après les différentes évolutions du marché au cours de cette dernière décennie, nous avons pu observer l'arrivée du Dynamic HTML ainsi que le Flash permettant d'avoir une nouvelle expérience de la navigation et des jeux par browser.

Suite à quoi, une fois le formalisme du HTML4 dépassé, la version 5 arriva avec son lot de fonctionnalités et de nouvelles normes comme par exemple la livraison du CANVAS qui a permis de faire un bon remarquable dans le développement du jeu par navigateur épaulée par les standards PHP et Javascript, offrant à l'utilisateur une forte expérience de jeu de par les nouveaux procédés de calculs de rendus graphiques.

Nous pouvons d'ores et déjà imaginer que, grâce à l'analyse faite partie 2, les technologies qui vont grandement se développer comme le fameux Node.js et sa capacité à effectuer un nombre de requêtes simultanées colossal de par son emploi côté serveur, ainsi que le développement en marche de l'interface Websocket qui, pour sa part permet l'exécution de ces requêtes de façon asynchrone, allégeant donc l'impact sur le navigateur pour l'utilisateur.

Par ailleurs, l'évolution de ces technologies demande toujours plus de puissance. De ce fait, c'est là qu'interviennent le Cloud et les clusters (ferme de calcul) en fournissant machines et capacités de calculs pour la bonne utilisation de ces technologies.

Le rendu obtenu pour les jeux par navigateur permettrait de faire entrer les joueurs dans une nouvelle ère.

Pour la plupart des joueurs ces évolutions ne seront pas flagrantes, cependant il existe une partie de la population adepte de la nouvelle technologie qui va chercher à découvrir de nouvelles possibilités dans les jeux vidéo.

L'actuel Free-To-Play, recueillant à l'heure actuelle un très grand nombre de joueurs va encore croître de façon phénoménale car ce modèle représente, pour les joueurs comme pour les compagnies, le modèle économique le plus équilibré et avantageux.

Que ce soit avec le HTML Living Standard ou encore avec l'intégration d'Unity, le web audacieusement surnommé « 3.0 » va permettre de donner un nouveau souffle aux jeux par navigateurs pour les joueurs, néophytes comme confirmés.

## VII. Bibliographie / Webographie

- . <http://php.net/ChangeLog-4.php#4.0.0>
- . <http://trends.builtwith.com/framework/Java-Servlet>
- . [http://fr.wikipedia.org/wiki/Free\\_to\\_play](http://fr.wikipedia.org/wiki/Free_to_play)
- . [http://fr.wikipedia.org/wiki/Pay\\_to\\_play](http://fr.wikipedia.org/wiki/Pay_to_play)
- . <http://www.statista.com/statistics/189660/breakdown-of-us-computer-game-sales-2009-by-genre/>
- . <http://www.jeux-navigateur.org/>
- . <https://www.everquest.com/home>
- . <https://www.vossey.com/news/L-Uber-Maj-jour-4---TF2-devient-un-F2P---Meet-The-Medic--i17871.htm>
- . <http://forums.jeuxonline.info/showthread.php?t=753612>
- . <http://aequitasmultigaming.enjin.com/forum/m/4669326/viewthread/3752539-list-nonpaytowin-mmos-some-discussion-about-computer-games-in-general>
- . [http://www.reddit.com/r/MMORPG/comments/28dquh/which\\_games\\_arent\\_pay\\_to\\_win/](http://www.reddit.com/r/MMORPG/comments/28dquh/which_games_arent_pay_to_win/)
- . <http://www.blog-jeu.fr/f2p-jeux-free-to-play/>
- . <http://www.oqaming.tv/news/les-mmo-a-abonnement-toujours-viables/8984>
- . <http://www.nplay.com/>
- . <http://www.w3counter.com/globalstats.php>
- . <http://www.web-eau.net/blog/comparatif-les-navigateurs-internet>
- . <http://www.ina-expert.com/e-dossier-de-l-audiovisuel-le-son-dans-tous-ses-etats/du-son-et-des-jeux-video.html>
- . [http://en.wikipedia.org/wiki/Browser\\_game](http://en.wikipedia.org/wiki/Browser_game)
- . <http://quake2playn.appspot.com>
- . <http://gaminganywhere.org/perf.html>
- . [http://en.wikipedia.org/wiki/Cloud\\_gaming#cite\\_note-ga-6](http://en.wikipedia.org/wiki/Cloud_gaming#cite_note-ga-6)
- . <http://www.bigfishgames.com/daily/cloud-gaming/>
- . <http://gaminganywhere.org/perf.html>
- . [https://en.wikipedia.org/wiki/Cloud\\_gaming](https://en.wikipedia.org/wiki/Cloud_gaming)
- . <http://mobilemarketingmagazine.com/itsmy-launches-html5-mobile-gaming-platform/>
- . <http://www.engadget.com/2012/07/02/sony-buys-qaikai/>