



Join-In

Senior Citizens Overcoming Barriers by Joining Fun Activities

AAL Joint Programme: Project No. 031121

Deliverable: 4.1

Low-Cost Solutions for Developing Home-Based Platforms Adapted to the Elderly

Date of deliverable: 10-02-12

Lead contractor for this deliverable:

Norut

Contributors:

Norut, IT Carlow, PAS, NST, HMGU, Happy, Val

Dissemination Level:

Public

Project Duration: Nov. 2010 – Feb. 2014

Project co-founded by

Disclaimer

The content of the document is owned by the Join-In Consortium and is protected by the applicable copyright laws. Any duplication, processing, distribution or any form of utilization is not authorized without prior written consent.

The commercial use of any information and data from the Join-In web sites requires a license from the owner of the information.

The Join-In Consortium members were committed to publishing correct information. The Join-In Consortium members do not assume responsibility for any errors or omissions nor do they accept liability for any direct, indirect, special, consequential or other losses or damages of any kind arising out of the use of this information.

When citing the deliverable please use the following information

Join-In Project Report D<Nr> <Title>, <Year>. Retrieved <The date on which you retrieved the information> from <URL>.

A short note on the publication to the Coordinator (contact@join-in-for-all.eu) would be very much appreciated.

Table of Content

About Join-In.....	1
1 Introduction.....	3
2 Overall architecture.....	5
3 User environment technologies	6
3.1 TV Set-top-boxes.....	6
3.2 PCs.....	7
3.3 Mini computers attached to TVs	7
3.4 Touch screen PCs	8
3.5 Mobile and tablet devices	8
3.6 Video gaming consoles.....	9
3.7 Small form factor computers.....	9
3.8 Smart Internet TVs.....	9
3.9 Game controllers and sensors.....	10
3.10 Discussion	10
4 System environment technologies.....	11
4.1 Elgg	12
4.2 Indivo Health.....	12
4.3 Drupal	12
4.4 Liferay Portal.....	13
4.5 Other technologies.....	13
4.6 Videoconferencing Technologies.....	13
5 Gaming technologies	15
5.1 Simple games	15
5.2 Advanced games with their own game server	16
5.3 Candidate 2D Game Technologies.....	17
5.3.1 Definitions And Terminology	17
5.3.2 Maturity and support of software	18
5.3.3 HTML5 game engines.....	18
5.3.4 The Render Engine http://therenderengine.blogspot.com/	18
5.3.5 Rocket Engine http://rocketpack.fi/engine/	19

D4.1 Low-cost solutions for developing home-based platforms adapted to the elderly

5.3.6	Isogenic Engine http://www.isogenicengine.com/	19
5.3.7	Game Closure http://gameclosure.com/	20
5.3.8	LimeJS http://www.limejs.com	20
5.3.9	Akihabara http://www.kesiev.com/akihabara/	21
5.3.10	GameJs http://www.gamejs.org	21
5.3.11	Flixel http://flixel.org/	22
5.3.12	JGame http://www.13thmonkey.org/~boris/jgame/	22
5.3.13	Slick http://slick.cokeandcode.com/	23
5.3.14	SDL.....	23
5.3.15	SFML	24
5.4	Candidate 3D Game Technologies.....	25
5.4.1	Panda3D http://www.panda3d.org	25
5.4.2	Unity3D	25
5.4.3	jmonkeyEngine www.jmonkeyengine.com	26
5.4.4	Open Wonderland http://openwonderland.org/	27
5.4.5	SmartFoxServer	27
5.5	Discussion	28
5.5.1	Comparison Tables	28
5.5.2	2D Desktop	29
5.5.3	Integration of game technology for the Join-In platforms	29
5.5.4	HTML5 solution	29
5.5.5	Java solution	30
5.6	Conclusion.....	31
6	Appendix A - All submitted requirements.....	39
7	Appendix B – HbbTV and CE-HTML	46
	HbbTV with stripped down Browser.....	46
	HbbTV with CE-HTML	47
	Examples of CE-HTML Applications.....	47
	CE-HTML Web Platforms, Portals and Content providers	48
	HbbTV Terminals with CE-HTML Browsers.....	48
	The future of HbbTV	49
8	Appendix C Open game controllers and motion sensors.....	51

About Join-In

Join-In aims at providing the methodology and the technologies for elderly persons to participate in social activities and have fun via digital media.

Loneliness in the elderly is a major problem in elderly care. Studies in Britain show that more than half of the people over the age of 75 live by themselves. Many of these suffer from loneliness and social isolation¹. Activities offered by social services do, however, often not reach those most in need. Challenges for the elderly include: social deprivation, low self-esteem or physical inability. Social isolation and health are closely related and may lead to a variety of physical disorders and even depression. Studies have shown the correlation between loneliness and poor health. Especially the effects on immune system, the cardiovascular system and the onset of Alzheimer's disease could be shown²³⁴.

The Join-In project aims at counteracting loneliness in the elderly by providing a concept, the methodology and technologies for elderly persons to participate in social activities.

Join-In is setting up a social platform for the elderly; it allows communication by TV, Tablet and PC. A multi-player serious game for the elderly is being developed. The interest in gaming is high in seniors: In a survey performed in Germany with 1200 participants, age above 61, two out of three PC users stated that they enjoy playing games regularly on the internet⁵. Studies⁶ could demonstrate the increase of cognitive skills, reaction times, self-esteem and the sense of well-being in the elderly when playing computer games. Another positive effect is that gaming is multigenerational and enables the elder generation socialising with the younger one, e.g. grandchildren. The concept includes exercising either by exergames or by moderated exercises as physical activity -besides supporting good health- counteracts the feeling of loneliness, while loneliness leads to less physical activity⁷. Recent results indicate that exergames create physical benefits and counteract

¹ Office of National Statistics: Older people,. Living arrangements. At: <http://www.statistics.gov.uk/cci/nugget.asp?id=1264>

² CARMA – Care for the Aged at Risk of Marginalization (QLK6-CT-2002-03421) - Recommendations and Guidelines to Policy Makers. (2005). <http://www.egga.ee/RecommendationsFinalwCoverTOC.pdf> Last accessed:2/10

³ Sorkin D, Rook KS, Lu JL: Loneliness, lack of emotional support, lack of companionship, and the likelihood of having a heart condition in an elderly sample. *Ann Behav Med.* 2002 Fall; 24(4):290-8

⁴ Tomaka J, Thompson S, Palacios R: The relation of social isolation, loneliness, and social support to disease outcomes among the elderly. *J Aging Health.* 2006 Jun; 18(3):359-84

⁵ OE24.at. Deutsche Studie - Sechs von zehn Senioren spielen am Computer. <http://www.oe24.at/zeitung/digital/article318942.ece>. Last accessed: 2/10

⁶ Basak C, Boot WR, Voss MW, Kramer AF: Can training in a real-time strategy video game attenuate cognitive decline in older adults? *Psychol Aging.* 2008 Dec; 23(4): 765-77).OE24.at

⁷ Hawkey LC, Thisted RA, Cacioppo JT: Loneliness predicts reduced physical activity: Cross-sectional & longitudinal analyses. *Health Psychol.* 2009 May; 28(3):354-63



Fig.1 Join-In Platform

loneliness⁸. Join-In encourages contacts with peers in the region and with family and friends living further afield - if necessary facilitated by an assistant.

Active participation is vital if the individual is to profit from the Join-In developments. Yet motivation for participation among the elderly is a challenge. One of the problems is the heterogeneity of the elderly, among other things regarding interests and health. Join-In is developing a methodology for elderly persons to participate in social activities. This is based on a thorough user requirement analysis. User groups are set up in Germany, Hungary, Ireland and Norway. The lead user group is based in Munich. Based on the results of the user requirement analysis and the analysis of relevant studies and related work a methodology for setting up a social networking platform which will encourage and enable involving homebound senior persons in social networking activities being developed. Digital inclusion and factors hampering its acceptance -such as accessibility, motivation, lack of skills and confidence- will be tackled and form part of the methodology. The involvement of user groups in four different countries will help us to achieve a European solution which will also be useful in other countries.

The Join-In project web-page:

<http://www.join-in-for-all.eu>

⁸ <http://www.theatlantic.com/technology/archive/2011/02/physical-video-games-may-help-the-elderly-psychologically/71184>

1 Introduction

In this report we give an overview of available low-cost technologies and open software systems that can be exploited in Join-In as technology building blocks for developing home-based services for the elderly. We describe their characteristics and discuss their usability in Join-In. We also do a first attempt to identify some overall technology and platform requirements, even though these will be influenced by the requirements of each individual game in question. This document is thus more a state-of-the art document with the first high-level requirements, and more detailed requirements will arise from the different work packages.

The (exer-)games and tools requirements are identified in WP2, WP3 and WP5.

WP2 “Social Networking Methodology and User Requirements”

D2.1 Report on User Requirement Analysis (M12) – Responsible: DMM. This report presents the results of user involvement showing how user needs of the elderly can be analyzed and how the input to product development can be identified. It involves the identification and evaluation of existing social networks and activities that are suitable for implementation in this project and the approach towards the envisaged technologies and activities.

WP3 “Game Design and Development”

D 3.1: Comprehensive report on Computer Based Games for Adaptation/ Development (M12) – Responsible: ITC. This will be a thorough report on the results of the analysis concerning the games as well as the special user needs to be considered for development based on the results of WP2, the target game(s) and possible platform issues considered for implementation It will provide the requirements for the software engineering.

WP5 “Interactive Applications for Physical Exercises for the elderly”

D.5.1 Report on requirements and state of the art in exergames for the elderly, low-cost motion tracking and respective HCI for elderly (M12) – Responsible: NST

The social part (defined in WP2) and the games (defined in WP3 and WP5) will lead to technical requirements that direct which gaming and social technologies to use, which again direct which user environment hardware platforms that are needed – and how they need to be integrated. However, these requirements will not be finalized until well after this report on platform requirements.

This report provides an overview and discuss various candidates for the user environment hardware platform, available gaming technologies, candidate social server technologies – and how they can be integrated in Join-In.

We do not aim to design the games and choose the technology and platform at the same time. When the limitations of the platform are known the design of games and tools can work within and around them.

2 Overall architecture

As shown in the figure below, the **user environment** is where the user accesses the Join-In services. The user environment will consist of a **hardware platform**, the **software** running on the platform, and the **user controllers**. For example, a user environment can be a set-top-box (i.e. hardware platform running the Join-In software) running an online game, where the user can control the gameplay with a game controller like the Wii Remote. In order to provide interaction with other players the user environment platform will be integrated with the system environment of Join-In.

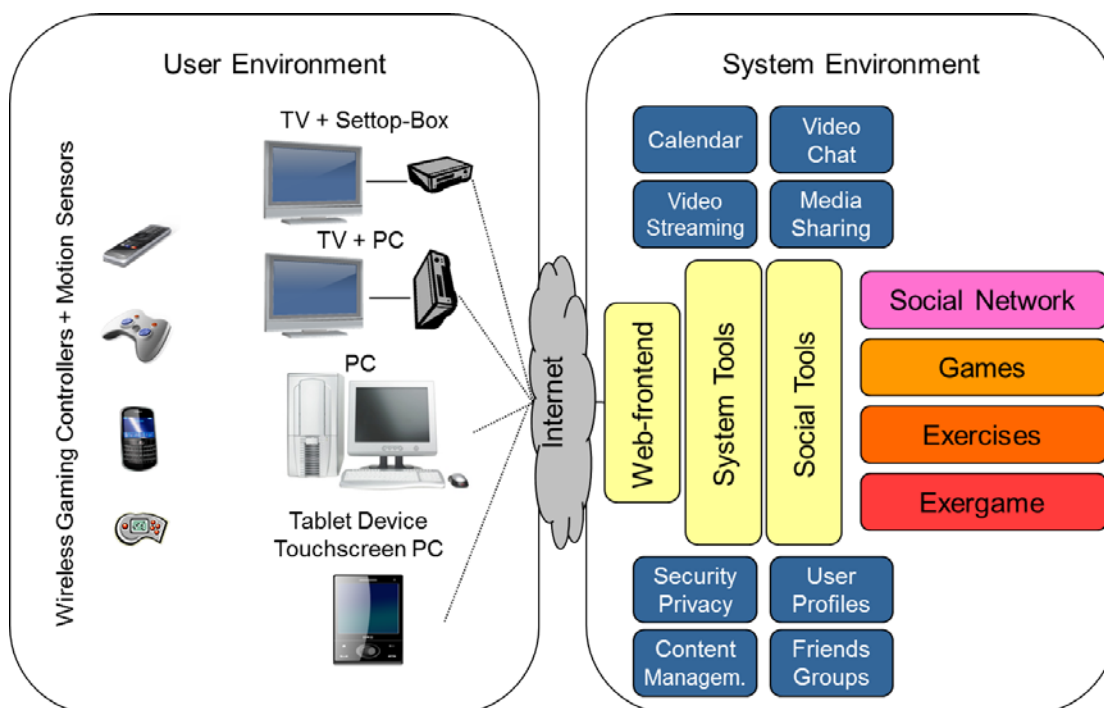


Figure 1 An overview of the user and system environments of Join-In

The system environment provides different services to the games and applications of Join-In. These are classified in 1) system tools 2) social and interactive tools and 3) game technology. The various games will be integrated in the system environment but may also be hosted separately. The system environment can be an ecosystem of servers and systems that are integrated seamlessly. For example, a game-server running a multiplayer game such as Ludo will need to interact with the Join-In social tools to identify which players are available or to send invitations to play to the user's friends.

3 User environment technologies

With the user environment we mean the environment where the user accesses the Join-In services. This may be in the user's private home, at a senior club, or even in an exercising club.

The Join-In technologies in the user environment will consist of a hardware platform, the software running on the platform, and devices for controlling the platform and its games and tools. The various user environment technologies will have different capabilities, which need to be taken into account when designing the Join-In platform.

The Game Controller Interface will need to be developed in order to allow communication between the game controller and the games on a specific user environment technology. This communication might be bi-directional to allow for feedback from the game. The Game Controller Drivers may be platform specific, and this may limit the selection of controllers for the various user platforms. If the controller's protocol is proprietary or complicated an important requirement would be the availability of a developer SDK (Software Development Kit) for the game controller, particularly if it has special features (e.g. accelerometer).

Nowadays there is a broad range of candidate hardware platforms available where social games can be deployed in the user environment, some of them are: set-top-boxes attached to TV's; media PCs attached to TV's; touch-screen PCs; standard PCs; video consoles attached to TV's; mobile and tablet devices; embedded PCs; and Smart Internet TVs.

The following sections discuss these platforms, their characteristics, capabilities, and suitability – including the possibilities of combining multiple platforms in the user environment.

3.1 TV Set-top-boxes

A set-top box (STB) is a device that connects to a TV screen and an external source (the Internet in the case of Join-In), turning the signal into content, which is then displayed on the television screen or another display device. Usually there is a dedicated remote control for the STB. STBs with a DVB tuner and Internet access are called hybrid boxes. Push services are mainly provided from the broadcaster and interactive services through the Internet.



There are a large number of commercial STBs available on the market, varying in performance, functionality, and price - from the simplest unit to the more advanced one with PC-like flexibility and processing power.

The appeal of using an STB as user environment platform is the low cost per unit and the ease of maintaining the client-side due to its server centric approach. Its weaknesses are that it is less flexible with user side interaction and support for user input controls as well as the lack of CPU power and browser functionality.

Join-In partner Pasifie has set-top-boxes that support HbbTV and CE-HTML. HbbTV is both an industry standard and promotional initiative for harmonizing the TV broadcast and TV broadband, and CE-HTML is a language for creating user interface pages and applications. More detailed information about this system is given in appendix B.

3.2 PCs

Standard PCs, as desktop units or laptops, are available in all sizes, shapes and price ranges. However, a standard PC is a multi-purpose tool meant for varying tasks, such as web browsing, word processing, Internet banking, travel booking, email/chat, social media etc.



The aim of the Join-In project is to provide the services in an easily accessible manner to user groups not necessarily familiar with general PC usage. Therefore we do not envision a general PC as the most suitable platform for Join-In – however some Join-In services can be accessible from a PC in order to be able to interact with users (relatives, caretakers) outside the core Join-In users.

3.3 Mini computers attached to TVs

The popularity of media PCs have created a market for small, silent and powerful mini computers, also known as barebones, that are suitable for homes. A mini computer attached to a TV in the user environment gives flexibility on the client-side for configuring and running advanced applications and provides much more local power than with an STB. The cost of this approach is however higher than using a standard STB. In addition not all the mini computers have interface to connect to TVs and therefore special attention must be put into their graphic cards.



Some application and services in Join-In may require more functionality and power than available with current commercial STBs, and while waiting for next-generation STBs to

come on the market a mini computer attached to a TV may be the best choice for experimenting with these applications and services.

3.4 Touch screen PCs

Touch-screen PCs are usually all-in-one devices integrating the processing unit in the screen itself. It is fully possible to dedicate the usage of such a PC for special purpose applications (like Join-In) and having the user control the functionality using only the touch-screen or another dedicated controller. The touch-screen PCs are supported by Linux and Windows, and have multiple input/output possibilities making it feasible to connect to other devices.



In contrast with Tablet devices, touch-screen PCs have standard desktop operative systems such as Windows or Linux Ubuntu. Full desktop operating systems allow the creation of more complex applications and solutions.

One affordable example is the ASUS Eee Top all-in-one PC, which also happens to be the current commercial user environment hardware platform of Join-In partner Valentia Tech.

3.5 Mobile and tablet devices

Mobile multitouch smartphones, and lately tablet devices, have experienced a tremendous increase in popularity the last couple of years. One major difference between the old generations of smartphones and the new ones are that the latter are (to various degrees) open programmable devices that are part of online ecosystems like Apple App Store and Google Android Market. There are a huge number of downloadable applications of all kinds available for these platforms making the devices useful for many purposes.



A tablet device could be a candidate as a Join-In user environment platform, running selected Join-In services. With its multitouch screen it is a good candidate for board games, and even for activity games exploiting its camera or a wireless motion sensor.

Since smartphones and tablet devices are normally open programmable platforms with wireless capabilities they are also candidates for being used in combination with other user environment platforms. They can for example function as dedicated remote controls or as game controllers in combination with a STB-based user environment platform.

However, some platforms such as the Apple iPhone and iPad have many restrictions for development and may therefore not be suitable for Join-In. For example, iPhones and

iPads provide very limited access to the Bluetooth interface making it harder to integrate external game controllers.

3.6 Video gaming consoles

The video consoles currently on the market - Nintendo Wii, Microsoft Xbox, and Sony Playstation 3 - are not open systems that can easily be interfaced to services envisioned in Join-In. The development of applications/games for those devices is a complex and closed process that can take months or years.



Video consoles are therefore not candidates for Join-In user environment platforms. However, some of their specially designed game controllers may be exploited and interfaced to a Join-In user environment platform.

3.7 Small form factor computers

There are also a variety of very small form factor computers, one being the plug computer type. They are lower cost, consume little power, often do not have a video card, and are suitable for running a media server, back-up services, file sharing, and not at least for supporting remote access. Possible candidate platform is the SheevaPlug, a Linux powered plug computer with networking and USB connectivity, including a development kit to assist in the development of software for the platform.



Their performance is limited, but in combinations with other user environment hardware platforms, like a STB, they could support input from controllers that are not supported by the specific STB.

3.8 Smart Internet TVs

Many new TV's nowadays are sold with built in access to Internet. However, for 2011 it is expected that only 20% of those will have full browser functionality while 80% are limited to HbbTV features.



Smart TVs include some features that were traditionally limited to PCs such as web browsing, video conferencing and access to custom made applications.

The market of Smart Internet TVs is emerging⁹ and therefore there is not a common set of characteristics across the different platforms. Most TV makers have different versions of the web browsers based on different technologies. That variability of capabilities and market fragmentation complicates the development of solutions for those TVs. In addition, these TVs are normally the most expensive ones and less likely to be owned in the near future by senior users.

3.9 Game controllers and sensors

There are many technologies that can be exploited as game controllers such as motion sensors, keyboards, balls, webcams, etc. Commercially available motion sensor kits like the *Freespace motion solution integrated with Bluetooth* and *Inertia Technologies ProMove-2* come with advanced capabilities and development kits. Another approach is to exploit existing gaming controllers that are intended as companions to the major video game consoles PlayStation3, Xbox and Nintendo Wii.



One main aspect to take into consideration when designing the games and services for the Join-In system is the type of sensors and controllers that can be supported by the user environment platform. Exergames may require the possibility to track users' movements and in many cases it may be necessary to support wireless connectivity to motion sensors. High data rates will also increase the power consumption of a wireless transmission significantly. Game controllers and sensors are often based on proprietary developments. Devices with standardized profiles are preferable.

Both the requirements of the game itself and the support of the chosen hardware platform will define the game controllers that are most suitable.

3.10 Discussion

As described in the chapters above several of the mentioned technologies can be suitable hardware platforms for the user environment. The games and exergames in Join-In, and their requirements, together with user preferences, cost and maintenance issues will define which hardware platforms to use.

⁹ <http://www.digitaltvresearch.com/ugc/press/22.pdf>

4 System environment technologies

This category encompasses examples of technologies for storing and maintaining information about users, games, and social and gaming relationships on-line. These technologies need to provide a common set of tools and services that allows the integration of the different games and services implemented in the Join-In System. For example, a game G will need to access information about the user U. The game will need to be assigned by a supervisor to that user and may also need to store some information about the user. In addition, some tools and services such as a calendar need to be provided for all the users of the Join-In System.

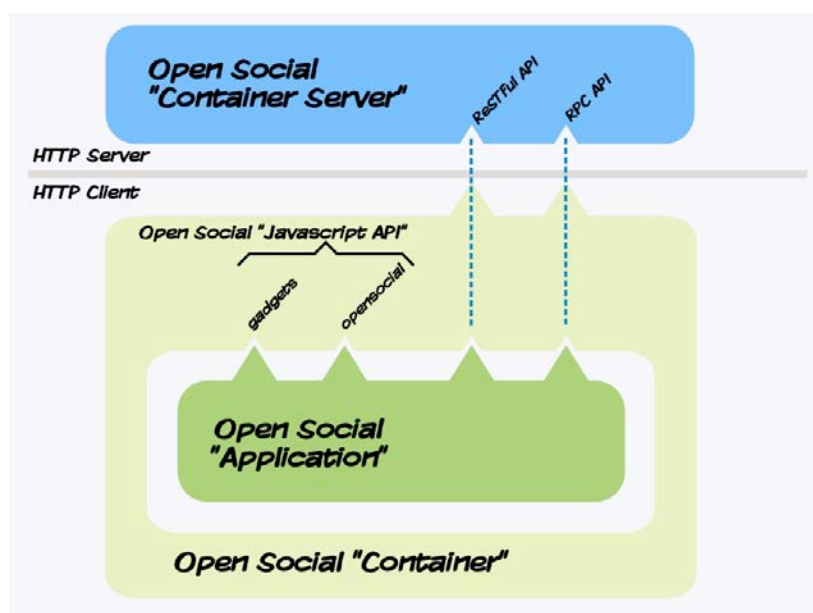


Figure 2 Open Social with different applications run in “containers”

The approach of having a common technology platform where multiple applications and games are integrated is widely used in the social web. Existing online platforms such as Facebook provide a set of tools and services via its API to third party applications and games. This is also the approach followed by Open Social (see image above) where different applications are developed for a standardized API and can be run in different container servers. In the Join-In case the different games and applications will be integrated in a Join-In Container Server that provides a set of services to the applications via a common API. The following section describes candidate technologies.

4.1 Elgg

Elgg is a popular open source engine to create social networks. It provides features that are traditionally required by social networks: users, friends, events, groups, sharing files, etc. In addition, it provides support for technologies such as RSS, OpenID, tagging, search, REST, multi-language, etc. It also incorporates the Open Social API that facilitates the creation of third party social applications. Extra modules can also be developed.



most forums, most web

Since Elgg is based on standard web technologies (e.g. PHP) it can run in nearly all web-browser and it can incorporate Join-In games that are developed using web technologies.

However, effort must be placed on adaptation of the user interface to the different devices and some of the functionality might not be useful, depending on the controllers (e.g. a keyboard will be required for text input).

The flexibility of this engine plus the amount of available features makes it one of the best candidates to host the different applications and the users' portal.

4.2 Indivo Health

Indivo is an open source Personal Health Records solution that allows the storage of private health information. It is designed to integrate web applications and also to share records among different users. The main advantage of Indivo versus other technologies is its enhanced security, since it is designed to store very sensitive information.

However, this technology is specialized for the health domain and is thus not the best solution for Join-In.

4.3 Drupal

Drupal is one of the most complete open web content management systems. It incorporates multiple modules to add social features and third party applications/games.



source network

Drupal is also based on PHP technologies and can be displayed in nearly all web browsers. Since it is one of the most popular open source content management systems there is a wide range of add-ons and plug-ins improving the functionality of the system. For example, there are extra modules that make the websites developed with Drupal adapted to different devices, including social networking tools, multi-language, etc.

Similarly to Elgg, Drupal can be one of the best alternatives to host the different applications and to host the user portal.

4.4 Liferay Portal

Liferay Portal is an open source (LGPL) software based on Java consisting of the Liferay Portal kernel application, the Liferay Content Management System (CMS) and the Liferay Collaboration for Web collaboration and social networks. Liferay Collaboration allows to assign tasks, create meetings, and share knowledge using built-in calendaring, task management, wikis, blogs, and more.

Liferay is designed to support business processes and is based on a service oriented architecture (SOA). Therefore Liferay is open for the integration of external applications.

The Liferay Portal Enterprise Edition is a commercial variant with long term support.

4.5 Other technologies

There are many different technologies that can be used to develop web applications and services, but as web technologies are continuously evolving it is not possible to provide a complete overview.

Among many others, Java EE is a popular solution for advanced web systems. Java Platform Enterprise Edition is a commercial solution for creating next-generation web applications¹⁰. Java EE allows the creation of complex applications and it is based on a well-known Java programming language, which is well supported by many devices. Another candidate technology can be Django that allows the creation of rich web solutions using the programming language Python.

The main challenge of using either Django or Java EE as the starting point for developing the Join-In system is the need to start from scratch - compared with developing on richer open source systems such as Drupal or Elgg.

4.6 Videoconferencing Technologies

Videoconferencing can be used by elderly people for communication with children, friends and relatives. However, low cost, easy to use and to administrate videoconferencing is a challenge.

Skype covers all basic requirements and now offers multi-user videoconferencing, but is not free of costs. High quality videoconferencing requires an expensive video bridge (Cisco:12 users 20k€). The problem is the simultaneous display of video streams from several users on one screen and the set up of a conference call.

¹⁰ <http://www.oracle.com/technetwork/java/javasee/documentation/index.html>

Open source approaches are for instance available in Ekiga (VoIP and video conferencing application), VLVC (VideoLan VideoConverence) and Jitsi (SIP communicator).

HTML5 videoconferencing technology is coming up in Open Source projects¹¹.

VCEG and MPEG and the Unified Communications Interoperability Forum founded by HP, LifeSize, Microsoft, and Polycom have jointly developed a new standard technology, scalable video coding, or SVC in 2007. The SVC design is an extension of the H.264/MPEG-4 AVC standard, not an entirely new scheme, so it is relatively easy for people who use the base standard to enhance their products to also support SVC. Google, Hitachi, Ricoh, and others, have since adopted SVC technology in a number of products. Vidyo's implementation of SVC is also behind Hangouts, the multipoint video-chat system in Google+¹². SVC has some licensing issues hindering the usage in Open source projects.

¹¹ Precence Displays used in Connected Vitality (<http://www.connectedvitality.eu/>)

¹² The picturephone is here. really. IEEE Spectrum Sep. '11

5 Gaming technologies

In this section we discuss different gaming technologies that can be used when creating games for Join-In.

The different games that Join-In shall support will have to operate within the chosen Join-In system environment. This means that the game must be managed by the Join-In system environment, and preferably also be able to cooperate with this environment.

Another requirement for the Join-In gaming technologies is that the games shall be able to be played over the Internet. We also assume that our chosen gaming technology at least must support a web browser interface, and that support for other interfaces is an advantage.

The simplest gaming technologies need only to be started and stopped by the Join-In system environment. These may be simple games that do not need to store any information online or share information with others from one play round to another. This category of simple games are discussed further in section 15.1.

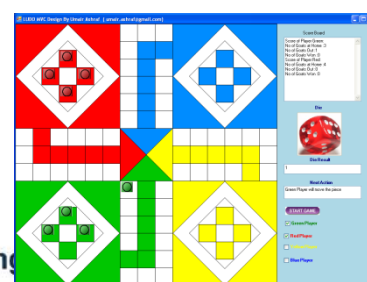
The more advanced gaming technologies have to cooperate with and within the Join-In system environment. These are games that store gaming information between play rounds, and are capable of sharing and coordinating game state and game information with others playing the same game. This category of advanced games are discussed further in section 15.2.

The candidate game technologies are discussed in sections 15.3 - 15.4. In section 15.5 comparison tables and our conclusion are presented.

5.1 Simple games

These simple games can be written in a wide range of languages that at least supports the creation of interactive web interfaces. Possible platform solutions are Silverlight, Flash, JavaScript, Python and Java. Our partner HappyWise Oy recommends Flash or Silverlight for the client side. Since this is a wide range of technologies that are quite standardized and well proven, they will not be described any further here.

The most important thing is that the game technologies must be able to communicate and integrate well with the Join-In system environment, and maybe even use the system environment as their game server. Currently we believe that this is possible for most of the game technologies.



It is also a possibility that Join-In can adopt some existing well-proven free open source games and interface them to the Join-In system environment. This will enable us to have a wide range of familiar games adapted to the Join-In system at a low cost.

The pictured Ludo games gives an example of a free Open Source LUDO Game written in the C# (CSharp) .Net framework as a windows desktop application by Umair Ashraf in 2010.

It is likely that most of the Join-In games will be in the category of simple games, and that some exergames also will be of this category.

5.2 Advanced games with their own game server

The main requirement for this category of games is that they can communicate, also through their game server, with the chosen Join-In system environment. These games are played over the Internet and operate via a connection to a dedicated game server. There are usually two categories of game servers; those that support a persistent world and those that do not:

- The server creates a persistent world and it relies on a dedicated server. These games must be running continuously. One example is Word of Warcraft (WoW).
- The server does not create a persistent world; it creates a playing arena for the purpose of a single round. The game server is used only for that single round. One example is Battlefield 2.

In addition, the most common usage of the game server is that the player has his own perspective on the game world.

Network (upstream) bandwidth and delay is one of the major limitations when hosting these game servers since the server application collects data from players and distributes them to other players. The game client connects to the game server. It is the server application that provides the connection and sends packets of information back to the game client. Many clients can be connected to the game server at the same time. Most advanced game clients are a hybrid of client–server and peer-to-peer architectures.

A software system created for the development of real-time game graphics is called a game engine. The main functionality is a rendering engine for 2D or 3D graphics and automatic collision detection. They also often offer sound, scripting, animation, artificial intelligence, multiplayer, streaming, memory management and threading. A new trend is game engines being built upon higher-level languages such as Java and C#.NET (e.g. TorqueX, and Visual3D.NET) or Python (Panda3D).

5.3 Candidate 2D Game Technologies

The following sections present a detailed look at the different 2D browser and non-browser based game engines and frameworks available to use in the Join-In project. The evaluation criteria for each engine include consideration of attributes such as maturity, performance, support and levels of abstraction. The latter attribute is important as engines or frameworks that offer high levels of abstraction typically enable developers to iterate faster during the development cycle. The browser-based engines were evaluated separately; and include ‘native’ engines (requiring no plug-in software to operate) and ‘plug-in’ engines (those requiring additional runtime software to operate).

Many modern browsers today support the canvas tag through HTML 5, which allows games to run without any plug-in. WebGL¹³. This technology allows developers to remove possible problems for users, such as downloading the required plug-in. However, not all users currently have browsers or devices that support HTML 5 based technologies, so consideration must be given to the possibility of plug-in based options that may be available.

For completeness, we also evaluate non-browser based solutions. These are frameworks capable of producing executables for specific operating system platforms.

5.3.1 Definitions And Terminology

Levels of Abstraction

Low	Medium	High
Provides the programmer with an interface to the hardware, for example OpenGL gives the programmer an interface to render graphics.	Provides the programmer with an interface a level above direct hardware access. In the case of OpenGL it provides access to graphics processing to create primitive objects. A medium level of abstraction will provide the ability to create sprites and 3D models.	Provides the programmer with an interface to functional algorithms. In game development examples of these algorithms would be functions for collision detection, artificial intelligence and physics.

¹³ Web-based Graphics Library - does the 3D graphics and is a context of the canvas

5.3.2 Maturity and support of software

Low	High
New software, either still in preview, alpha or beta, or in first or second iteration. Brief or no official documents, with a small amount of users.	Established software used in many commercial titles. Represented in IRC chat channels and has a large community forum. A complete set of official documents for set up, help and tutorials.

5.3.3 HTML5 game engines

HTML5 game engines are based on the new HTML5 specification, which has new features for making games, such as Canvas, Video, and Audio elements. HTML5 combined with the web technologies of the Open Web Platform, gives plug-in-free gaming. A good example of how HTML5 and open source can be used to create web browser games is given in (Pena&Costa 2010).

Examples of HTML5 game engines are RocketEngine, Aves, Isogenic Engine, Gameclosure, LimeJS, Akihabara and The Render Engine. Zynga, the developer behind Facebook’s Farmville has recently acquired a HTML5 engine called Aves, which may accelerate the support and adoption of HTML5 technology in web browsers.

The advantage of HTML5 is that there is no need for additional browser plug-ins, if the browser supports the full HTML5 specification. The disadvantage is that it is still not a mature technology. But HTML5 combined with open source is a new and rapidly growing technology, and is a technology option for Join-In to consider.

A history of feature grows including HTML5 in the Opera browser is listed at <http://www.opera.com/docs/specs>. The Opera presto25 level is currently used in set-top-boxes available in the market.

5.3.4 The Render Engine <http://therenderengine.blogspot.com/>

Details: The Render Engine is a cross-browser, open source game engine. It claims to support all modern browsers, and has fall back flash support for browsers such as Microsoft Internet Explorer 8 (IE8).

Code repository: <https://github.com/bfattori/TheRenderEngine>

Programming Language: Javascript

Performance: From the demos provided it appears to work well with 2D games and has Box2D integration in some demos.

Cost: MIT licensed, free to use.

Pros: Well documented API at

Cons: Currently has a flash requirement for sound playback and IE8.

Abstraction: High: Provides support for 2D maths, sprites, particles, collisions and other high level features.

5.3.5 Rocket Engine <http://rocketpack.fi/engine/>

Details: RocketEngine is an HTML5 game engine that has support for multi-platform multi-player 2D games. It also claims to work in every major browser without the need for plug-ins.

Programming Language: JavaScript/CSS.

Performance: Appears to run 2D games with multiple clients and no visual performance drop. Networking demo: <http://youtu.be/zZyoDfau3is>

Cost: Rocket Pack is currently inviting selected developers to test drive the Rocket Engine. See http://rocketpack.fi/press/files/rocketpack_pressrelease_01092010.pdf for more details.

Pros: Comes with an editor

Cons: No WebGL support yet.

Maturity/Support: Unknown since it is by invitation only.

Abstraction: Unknown.

5.3.6 Isogenic Engine <http://www.isogenicengine.com/>

Details: The Isogenic Game Engine is a web-based MMO (Massively Multiplayer Online) game engine.

Programming Language: Javascript

Performance: Demo with 500+ sprites being rendered:
<http://www.youtube.com/user/coolbloke1324>

Cost: Currently in beta, basic free version or \$200 dollars for a commercial license.

Pros: Comes with NoSQL storage (MongDB) and Node.js for server functionality in MMO games.

Cons: Costs at least \$200. Support may be an issue as it would appear there is only one developer working on the project.

Maturity/Support: Beta invite, documentation in the process of being written

<http://www.isogenicengine.com/documentation/>

Abstraction: Unknown since it is a beta sign up.

5.3.7 Game Closure <http://gameclosure.com/>

Details: Game Closure claims to offer an HTML5-based multi-player game development kit.

Programming Language: JavaScript

Performance: The demonstrations show 2D games with multiple clients and no visual performance drop.

Card game and editor demo: <http://youtu.be/k3FPJn26Kqk>

Tower defence demo: <http://youtu.be/8KACnnquQRo>

Cost: Currently private beta. Available at <http://www.gameclosure.com/signup.html>.

Pros: Comes with an editor

Cons: No WebGL support yet.

Maturity/Support: Unknown since it is a beta sign up.

Abstraction: Unknown.

5.3.8 LimeJS <http://www.limejs.com>

Details: HTML5 game framework. Claims to work on all modern touch screens and desktop browsers.

Programming Language: Javascript

Performance: The showcase games available are basic but run without any performance issues.

Chain game: <http://www.limejs.com/static/roundball/index.html>

Another chain game: <http://www.limejs.com/static/zlizer/index.html>

Cost: Apache License v2.0, free to use.

Pros: Closure support.

Cons: No WebGL support.

Maturity/Support: Well documented <http://limejs.digitalfruit.ee/docs/> Has an active mailing list <https://groups.google.com/forum/#!forum/limejs>

Abstraction: Medium: Provides support for animations, audio, transitions.

5.3.9 Akihabara <http://www.kesiev.com/akihabara/>

Details: Akihabara is a set of tools and libraries that use HTML5 canvas tag and some standard hooks to create games in Javascript.

Programming Language: Javascript

Performance: Performs all showcase demos with no performance issues.

Top-down shooter: <http://www.kesiev.com/akihabara/demo/game-solitude.html>

Legend of Zelda clone: <http://www.kesiev.com/akihabara/demo/game-tlol.html>

Pac-man clone: <http://www.kesiev.com/akihabara/demo/game-capman.html>

Tetris clone: <http://www.kesiev.com/akihabara/demo/game-tspin.html>

Cost: GPLv2/MIT(X11), free to use.

Pros: Tutorials and documentation <http://tools.bostongamejams.com/akihabara/docs/>

Cons: No WebGL support

Maturity/Support: On version 4.0 and is actively maintained by a small group of users.

Abstraction: High: Provides high level abstraction for collisions, rendering transforms, font rendering, input, double buffering, audio support, staging/transition screens, HUD, Z-indexing, landscape display, touch input for Android and iOS.

5.3.10 GameJs <http://www.gamejs.org>

Details: GameJs is a thin library on top of the HTML canvas element. The GameJs API is a port of PyGame to JavaScript.

Programming Language: Javascript

Performance: A lot of the showcase demos appeared to run poorly

Racing game: <http://combatracer.allcx.com/>

Top-down shooter: <http://programmer-art.org/dropbox/fighter-static/index.html>

Cost: Free to use. License is unknown, possibly LGPL since it's a port of PyGame.

Pros: Built on a know API.

Cons: Does not appear to perform well.

Maturity/Support: Based on PyGame so the API has had a lot of release cycles. A small group actively maintains the library. The Active mailing list is hosted at: <https://groups.google.com/forum/#!forum/gamejs>

Abstraction: High: Provides high level abstraction for input (mouse, keyboard), font rendering, synchronous http requests to a game server, pixel perfect collision detection, audio support, sprite and sprite groups, transformations, matrix operations, AI and other tools.

Documentation: <http://docs.gamejs.org/>

5.3.11 Flixel <http://flixel.org/>

Details: Flixel is an open source game-making library that is completely free for personal or commercial use. It is written entirely in Actionscript 3, and is designed to be used with free development tools.

Programming Language: Actionscript

Performance: The entire showcase game run perfectly and show a wide variety of game genres.

Cannabalt: <http://www.adamatomic.com/canabalt/>

Mode (Platform shooter): <http://www.flixel.org/mode/>

Cost: MIT(X11), free to use. Designed to be used with free tools.

Pros: Works on browsers that do not support the canvas tag such as IE8.

Cons: Requires flash plug-in.

Maturity/Support Well documented API: <http://flixel.org/docs/> that is maintained by a small group of users.

Abstraction: High: Provides support for cameras, path-finding, replays, groups and collisions, tile-maps and auto-tiling, particles, game saves
<http://flixel.org/features.html>

5.3.12 JGame <http://www.13thmonkey.org/~boris/jgame/>

Details: Game is an open source 2D multi-platform game engine. It runs on the Java JRE 1.3+ platform with optional OpenGL (JOGL) enhancements, the J2ME (MIDP2.0/CLDC1.1) mobile platform, and the Android (2.1+) platform. There is also a Flash (Actionscript 3) version.

Programming Language: Java

Performance: Has a wide selection of demos, all allowing multiple objects on screen with no performance issues.

Showcase demo: <http://fubz.us/games/bin/demorpg.php>

Cost: BSD License, free to use.

Pros: Actively maintained, multi-platform support.

Cons: Basic API, a lot of functionality missing.

Maturity/Support: The project appears to be maintain by one a small group (maybe one person).

Abstraction: Medium: Abstracts a lot of gameplay features, but ignores many that others provide such as audio, tile map rendering, pixel perfect collision detection

<http://www.13thmonkey.org/~boris/jgame/JGame/javadoc/overview-summary.html>

5.3.13 Slick <http://slick.cokeandcode.com/>

Details: Slick is 2D Game Library based on LWJGL

Programming Language: Java

Performance: Performance does not appear to be an issue running any of the demos or user created games:

User prototype: <http://www.youtube.com/watch?v=AYIN6t1R-Jw>

RPG demo: <http://www.youtube.com/watch?v=dBopJA2HUZ0>

Showcase demos: <http://slick.cokeandcode.com/static.php?page=demos>

Cost: BSD Licensed, free to use.

Pros: High level library based on LWJGL

Cons: Requires plug-in to be already installed.

Maturity/Support: Active development, has over six members working on the code.

Abstraction: High: Provides support for Image Loading (png, jpg, etc), transforms, sprites, animations, rendering through gl, audio support, input, tile map rendering, geometry primitives for collision

Feature: <http://slick.cokeandcode.com/static.php?page=features>

API documentation: <http://slick.cokeandcode.com/javadoc/>

5.3.14 SDL

Details: Simple DirectMedia Layer (SDL) is a cross-platform, free and open source multimedia library written in C that presents a simple interface to various platforms' graphics, sound, and input devices.

Programming Language: C, with bindings to many other languages.

Performance: SDL has been used by a lot of publishers for games:
https://secure.wikimedia.org/wikipedia/en/wiki/List_of_games_using_SDL

Cost: Licence 1.2 LGPL, 1.3 zlib, free to use.

Pros: Extremely well documented, used in many professional games.

Cons: No browser support. Low level of abstraction, additional libraries would possibly be needed.

Maturity/Support Actively maintained by dozens of people. Active IRC channel on #freenode, and mailing list. Well documented API: <http://www.libsdl.org/docs.php>

Abstraction: Low: SDL itself is very simple; it acts as a thin, cross-platform wrapper, providing support for 2D pixel operations, sound, file access, event handling, timing, threading, and more. It is often used to complement OpenGL by setting up the graphical output and providing mouse and keyboard input, which are beyond the scope of OpenGL.

Also provides extra libraries such as:

SDL_image - support for multiple image formats

SDL_mixer - complex audio functions, mainly for sound mixing

SDL_net - networking support

SDL_ttf - TrueType font rendering support

SDL_rtf - simple Rich Text Format rendering

5.3.15 SFML

Details: Simple and Fast Multimedia Library (SFML) is an object-oriented, cross-platform, open source multimedia API written in C++. It is intended as a more modern alternative to SDL with a greater emphasis on object-oriented programming.

Programming Language: C++, with bindings to many other languages.

Performance: Appears to perform well - M.A.R.S Shooter:

<http://www.youtube.com/watch?v=M9FkgS4ZhTg>

Cost: zlib licensed, free to use.

Pros: Cross platform, a lot of functionality. There are many tutorials <http://www.sfml-dev.org/tutorials/1.6/>

Cons: No browser support

Maturity/Support Actively maintained by, with a new release coming soon, as well as an active forum. Well documented API: <http://www.sfml-dev.org/documentation/1.6/>

Abstraction: High: Provides support for input, sprites, multiple image types (jpg, png, etc), collision detection, audio, shaders, threads, unicode and networking. Also has integration with Qt or other tool kits for editors.

5.4 Candidate 3D Game Technologies

This section presents an analysis and comparison of 3D game engine technologies. General aspects for developing games such as performance, portability, programming language, browser support and cost will be addressed. Consideration is also given to 3D capabilities and peripheral support as they are both required for the development of an exergame.

Exergames require a controller such as the Wii Balance Board or the Microsoft Kinect. To successfully use these in a simulation, skeletal rigging is required to allow the avatar (or screen representation of a user) to mimic the users movements. Skeletal rigging is the process of attaching a virtual skeleton to a 3D model in order to manipulate a model in a realistic way. Selecting a programming language or engine that does not support 3D simulations or the use of peripheral devices (specifically game devices) will hinder the development of an exergame. All the technologies listed below have 3D support.

5.4.1 Panda3D <http://www.panda3d.org>

Details: Panda3D is a free, open source Object-Oriented 3D game engine, which includes graphics, audio, I/O, collision detection. The users of Panda3D include the developers of several large commercial games, a few open source projects, and some university courses. Wide range of user-provided features and examples (PhysX integration, AI, GUI, particle system etc.)

Programming Language: C/C++, Python

Performance: Appears to perform well

Cost: BSD licensed, free to use.

Pros: Easy to use. Web plug-in for deployment over the web. An "all-into-one" application packaging system

Cons: it may be too advanced for the usage we need.

Maturity/Support: Productive/Stable. Actively maintained and active forum. Well-documented API.

Abstraction: High. it's more than just graphics. It has integrated physics (panda and ODE), network, audio, a decent AI, and tools to pack and distribute your games

5.4.2 Unity3D

Details: Unity3D is a very popular and feature rich 3D game development tool. Scripting in Unity is done in: C#, Boo or JavaScript. It is multiplatform including also Nintendo Wii, PlayStation 3, iPhone/iPad and Google Android. However it is not open source and plug-in software is required. It won the Wall Street Journal 2010 Technology Innovation Award in the software category. Unity3D games are now

emerging on Facebook (<http://www.socialtimes.com/2011/03/gdc-how-will-unity-3d-propel-hardcore-games-onto-facebook/>)

Unity3D should be considered for Join-In, it has good development tools and can be used by several platforms

Programming Language: C#, Javascript

Performance: Appears to perform well on small games

Cost: Indie version is free to use. There will be high licensing costs for source code control, commercial platforms or extensive multiplayer functionality.

Pros: Easy to use.

Cons: A disadvantage is the need for plug-ins and costs.

Maturity/Support: Productive/Stable. Actively maintained and active forum. Well-documented API. There is, however, some worry that it is yet not a stable enough, that it is under-documented and under constant development.

Abstraction: Medium. What makes Unity really shine is the game editor.

5.4.3 jmonkeyEngine www.jmonkeyengine.com

Details: The jMonkeyEngine (jME) is a free, open source Java software and runs on any OpenGL2 -ready device with a Java Virtual Machine. It is built around a shader-based architecture. This ensures compliance with current and next generation graphics standards. Several commercial game studios and educational institutions use the jMonkeyEngine.

jMonkey may be a candidate for Join-In, but it may be too advanced for the usage we need.

Programming Language: Java

Performance: perform well

Cost: free

Pros: Easy to use.

Cons:

Maturity/Support: Productive/Stable. Actively maintained and active forum. Well-documented API.

Abstraction: Medium.

5.4.4 Open Wonderland <http://openwonderland.org/>

Details: Open Wonderland is an open source alternative to Second Life. It provides a client and server to host a 3D virtual world where the users are represented by avatars. The users can interact by using voice chat. In addition, they have access to a set of applications and games inside the world. Users who want to play and be able to communicate with others commonly use virtual worlds. In addition, virtual worlds have been used in education (e-Learning, simulations) and also for physiological treatment.

Since Open Wonderland is open source and Java it can be modified to satisfy the requirements of Join-In. However, the users may be overwhelmed by the wide range of user interactions (e.g. users can fly, chat, walk) making it less suitable for users with low computing skills. In addition, it requires a powerful user environment computer to handle the 3D rendering.

It might be costly time wise to choose this platform if there is no prior knowledge about it in the project.

Programming Language: Java

Performance: well, but sometimes unstable

Cost: free

Pros: multiuser environment for learning and interacting

Cons: complex to use and requires a powerful computer for 3D rendering

Maturity/Support: community support and mature

Abstraction: high

5.4.5 SmartFoxServer

“SmartFoxServer is a comprehensive platform for rapidly developing massive multi-user applications and games with Adobe Flash/Flex/Air, Java, Android, .Net, Unity3D, Silverlight, Apple iPhone etc...”

It was developed with simplicity in mind, allowing developers to quickly create any type of multiplayer interactions, from basic chat applications to complex real-time games and MMOs”

<http://www.smartfoxserver.com/>

This should be considered as a server candidate since it supports most technologies and aspects visioned in this document and is designed for rapid development.

5.5 Discussion

5.5.1 Comparison Tables

2D Native

	The Render Engine	Rocket Engine	Game Closure	Isogenic Engine	Akihabara	GameJs
Language	Javascript	Javascript	Javascript	Javascript	Javascript	Javascript
Performance	Medium	High	High	High	High	Medium
Cost	Free (MIT/X11)	Unknown	Unknown	Varies	Free (GPLv2/MIT(X11))	Free
Maturity/Support	Medium	Unknown	Unknown	Unknown	Medium	Low
Abstraction	High	Unknown	Unknown	Unknown	High	High
Browser Support	Yes	Yes	Yes	Yes	Yes	Yes

2D Plug-in

	Flixel	JGame	Slick
Language	Action-script	Java	Java
Performance	High	High	High
Cost	Free (MIT/X11)	Free (BSD)	Free (BSD)
Maturity/Support	Medium	High	High
Abstraction	High	Medium	High
Browser support	Yes (Flash plug-in)	Yes (Java plug-in)	Yes (Java plug-in)

5.5.2 2D Desktop

	SDL	SFML
Language	C	C++
Performance	High	High
Cost	Free (1.2 LGP, 1.3 zlib)	Free (zlib)
Maturity/Support	High	High
Abstraction	Low	High
Browser Support	No	No

5.5.3 Integration of game technology for the Join-In platforms

This section compare and discuss the game technologies that can be used on the Join-In platform. The special user environment (including settop-boxes and tablet-PCs) is putting constraints on the technical solution that have to be taken into account. The social platform will provide access to different games and manage the plays as it supports looking for gaming partners by providing player profiles. The results of the games – e.g. level and score – will be stored at the platform. The games and the social platform do not necessarily have to run on the same server, thus interfaces like provided by OpenSocial have to be established that enable the integration of the data needed for and resulting from the plays.

The results of these experiments will further guide us in selecting the appropriate technology for the social platform. This will be done later in the project.

5.5.4 HTML5 solution

There is evidence of developers who have successfully used HTML5 engines and peripherals such as Kinect (<http://www.youtube.com/watch?v=UMBWYBH3TKg>) and Wii Remote (<http://www.youtube.com/watch?v=vliPnOKbrv4>), however these examples are not fully functional and the source is not openly published. These examples also appear to use additional technologies such as Flash. There are risks associated with using newer technology such as HTML5 in place of an established programming languages such as Java. Development work could be hindered by engines or frameworks with incomplete functionality such as lack of peripheral support.

No evidence of existing HTML5 exergames was found, although a Unity3D project titled “GameChanger: A Middleware for Social Exergames” investigated writing integration for HTML5 and WebGL for additional portability¹⁴

HTML5 is a new technology, and although it is an extension of HTML 4.01, all new features are not yet fully supported by all the actual browsers. Additionally, support for HTML5 on the commercial set-top box platform is not guaranteed; HTML5 is one of a number of features currently under discussion for the next generation HbbTV 2.0.

However, we could develop and exploit some middleware software (for peripheral support) that would mitigate the above risks to a certain extent.

5.5.5 Java solution

The choice of Java as the development platform presents the strongest case at the time of writing, given the immaturity of HTML 5 and the software technology restrictions on the commercial set-top box platform.

Additionally, Java will run on all required platforms and supports peripherals such as Microsoft Kinect and the Nintendo Wii Remote through specific libraries. Currently, the two most promising candidate Java engines for Join-In are:

- LWJGL that has proven successful for creating complex 3D games in browsers such as MineCraft (<http://lostincod.net/blog/minecraft-client-java-and-ant>).
- jMonkeyEngine is also a very developed and well-supported engine that would be worth investigating.

The performance of Java applications is often criticised, which is a potential risk as the goal of the project is to build exergame applications within a 3D environment. Performance is relative to both the power of the machine running the software and the requirements of the software itself. In terms of the latter, it is usual for a game application to have multiple objects on screen at once with physics and artificial intelligence (AI) updates applied to each object on every frame. Conversely, it is typical for fitness games to have very few models on screen at one time. Physics is normally limited to basic acts such as kicking or bouncing a ball with minimal AI.

Minecraft is a Java-based massively multiplayer online game (MMOG) with demanding processing requirements containing thousands of objects on screen at once within a procedurally generated world. A brief summary of Minecraft performance on a range of different hardware from the Minecraft Wiki (ref: http://www.minecraftwiki.net/wiki/Hardware_performance) is provided below:

¹⁴<http://www.conference-publishing.com/download.php?Conf=ICSEWS11GAS&Paper=3d7a27aa7fe55a1d6a234367c85858>

When running Minecraft on high-end machines such as Intel Core i5 or Intel Core i7 processors with 2-6GB RAM and dedicated GPU, frame rates in excess of 200 frames per second (fps) are common. 30-60 fps is considered the industry standard requirement to avoid jitter and lag.

On mid-range machines such as the Intel Core 2 Duo architecture or equivalent, low end GPU/ integrated graphics unit, with 1-2GB RAM, the performance ranged from 30-100fps, still within an acceptable range.

On low-end machines (such as notebooks or netbooks) with Intel Core 2 ULV processors or less powerful Intel Atom processors, 1-2GB RAM and integrated or mobile graphics processors the performance range varied from 10fps (unacceptable) to 20+ fps (playable).

The performance of Minecraft on Windows and Linux operating systems was superior to the OS X results. The majority of OS X results were in the mid range with none exceeding 100 fps.

The evaluation of Java engines by the development team is ongoing and specific technology selection will be made in due course.

5.6 Conclusion

Is as an an open issue whether any of the Join-In games will be built using game servers or with game engines. These systems may be complicated for the Join-In games, and making the game server or game engine communicate with the chosen Join-In system environment may not be a simple. However, the first Join-In games will be developed for web browsers in order to support the widest range of user side technologies.

There is a degree of commonality between the proposed technology platforms; they all are devices that will be connected to the Internet, and web browser clients (of different capabilities) have been developed for each of them.

Therefore the optimal approach to development in the project is to select an engine that enables that *games can be embedded into browsers* and that will also *support the use of external gaming peripherals*. Either one of these requirements are manageable in most of the engines that have been evaluated. An engine or framework that supports both requirements narrows the available options.

A game engine that supports 3D and has a defined Application Programming Interface (API) for using peripheral devices such as Kinect or Wii Remote is desirable. Without an API it would not be possible to access a device. In this case the API would have to be defined by the development team. A Software Development Kit (SDK) may be provided by the manufacturer of a peripheral device depending on the level of complexity of that

device. Communication with that SDK may provide some technical challenges from either Java or HTML 5.

If we are limited to Java 1.4 in the set-up-boxes, the best approach may be to create a small demo using the Lightweight Java Game Library (LWJGL) that appears to support Java applets¹⁵.

In summary, there are two possible development technologies for game engines that support web browsers: HTML5 engines and Java engines which are now considered. HTML5 is now supported by all the major browser vendors and we believe that it is set to become a significant gaming platform with support for features such as hardware accelerated 3D graphics, web sockets and sound APIs. However, the HTML5 specification is still being developed and the degree of support of the HTML5 standard varies among the different web browsers. There are also uncertainties surrounding its support on commercial set-top box platforms.

For these reasons, the development team will initially focus on Java based solutions as the preferred game development technology for Join-In. Web-browser based games based on a Java-engine can also be integrated in a HTML5 framework, enabling integration of video and audio without requiring additional plugins in the client side.

¹⁵http://lwjgl.org/wiki/index.php?title=Deploying_with_the_LWJGL_Applet_Loader_-_Introduction

6 Platform and system requirements

It is not possible to define the Join-In system requirements in details at this stage of the project. Many of the requirements will be defined by the games to be developed later, and they will arise from different work packages, as pointed out in Section 1.

This is however a start, listing some high level requirements for the system. The requirements are prioritized based on the MoSCoW methodology¹⁶, categorising the requirements as **M**ust, **S**hould, **C**ould and **W**on't. The priority of requirements might change during the project due to work in other work packages, but those with highest priority are most unlikely to change.

These initial requirements are sorted by category and priority. The categories are:

1. Overall
2. Senior
3. Social tool
4. Game
5. User environment
6. Operability
7. Documentation
8. Security

The priorities (Pri) are as follows:

1 – M (Must have)

2 – S (Should have)

3 – C (Could have)

4 – W (Won't have) Requirements that are considered too detailed at this level are also set to 4.

¹⁶ http://en.wikipedia.org/wiki/MoSCoW_Method

Id	Overall	Pri
1.1	The system must be easy-to-learn/self-explanatory/obvious in use for an average senior user (few steps, intuitive menus, etc.)	1
1.2	The system must make consistent use of terminology, abbreviations, formats, titles	1
1.3	The system must present the social tools, games and exergames in an integrated and uniform view to the senior user	1
1.8	It must be possible to include audio elements (music, ambient sounds, etc)	1
1.9	Data formatting must be consistent	2
1.10	The system must accommodate future increases in the number of users, database sizes and processing loads with minimal degradation of performance, and without having to be redesigned or reimplemented.	2
1.12	Date/Time must default to local time.	3
1.13	Menus must be grouped and organized according to functionality.	3
1.16	The system must be usable “offline”, i.e. when the user portal is not connected to the internet.	4
	Senior	
2.1	The system must address hearing impairment of senior users.	1
2.2	The system must address reduced fine motor skills of senior users.	1
2.3	The system must address visual limitations of senior users (e.g., adjust contrast)	1
2.4	The system must support multilingual environments.	1
2.5	The system must be affordable for the average European senior.	2

Social tool		
3.1	A user must be able to view another user's on-line open public profile and game status	1
3.2	A user must be able to chat to fellow on-line users.	2
3.7	The social tools must support a search functionality so a user can look up a friend.	1
3.9	Some media sharing with other users must be possible.	2
3.10	Moderators must be enabled to create user-groups and invite users to take part in group activities.	1
3.11	The social tools must support a videoconference system.	2
3.13	An easy and intuitive email system must be supported.	3
3.14	The social tools must support calendar planning functions where activities can be set up.	3
3.15	Videoconference system must support transcoding or continuous presence solution for multi-user videoconferencing ¹⁷ .	3
3.16	Videoconference system must support fast generation of H264 (mpeg4) or H263 video streams. ¹⁸	3
3.17	Videoconference system must support SIP communicator for flexible initiation and administration of calls.	3
3.18	Videoconference system: Video cameras with microphone and excellent echo cancellation.	3
3.19	It must be possible for a user to delegate responsibility to a helper for setting up of the user profile, defining friends, etc.	4
Game		
4.2	The system must provide easy access to the games and exergames, like start/stop, storage of gaming points, log of progress, etc	1
User environment		

¹⁷ This allows each endpoint connecting to be a different speed or video algorithm. The video stream is generated by the video cards and transcoded to whatever format each endpoint is using. Transcoding screen layout is one site only whereas Continuous Presence can have all the different screen layouts available.

¹⁸ Comment: Current high end STBs offer two H264 video-encoders allowing the display of two communication partners.

5.1	The user environment must allow for “alternative” (non-keyboard, non-mouse) input devices (e.g. voice or remote control).	1
5.3	The screen must be easily readable under different lighting conditions.	2
5.6	The activity controllers must have low battery indication.	3
5.7	The system must be able to communicate with activity controllers like Wii Remote, Kinect, HilcrestLabs FreeSpace	3
5.8	The system must be able to receive signals and data from controllers, received from wireless receivers or via cable.	3
5.9	The system must only use solid-state technology for data storage (as tablet and laptop devices can fall from exercise bikes during biking games).	4
5.11	The user must be able to set the screen orientation (landscape/portrait).	4
5.12	The user environment must be fitted with a camera. The camera must have good-quality optics so as to capture high-quality images.	4
5.13	The user devices must be resistant to cleaner and wipes, and be splash-proof	3
5.14	The user devices must be fitted with a touch-screen.	3
5.15	The user devices must have multiple means of connecting to the internet Wi-Fi 3G LAN.	3
5.16	The user devices must incorporate 2+ USB ports.	3
5.17	The user devices must incorporate a Bluetooth transmitter/receiver to allow connection to wireless sensors.	3
5.18	The user devices must incorporate an infrared port.	3
5.19	Data must be stored locally on the user devices and automatically transferred to the back office system whenever network connectivity is established.	4
5.20	The Join-In system must be accessible through multiple devices.	4
5.21	The Join-In system must run seamlessly on different user environment hardware devices.	4

5.22	The Join-In user environment technologies must be able to operate in the absence of any network connectivity.	4
	Operability	
6.1	The system must be stable and robust	1
6.4	All system maintenance activity must take place at an agreed time of low system use (e.g. early Tuesday morning).	3
6.5	It must be possible to apply system updates remotely.	3
6.6	There should be no more than 45 minutes unscheduled down time in any month.	3
	Doc	
7.1	There must be comprehensive and meaningful user documentation to assist the users in getting the best out of the system	1
7.2	The user must have a mechanism by which they can suggest modifications and enhancements to the system	1
7.3	Two levels of error messages must be provided: error and warning.	1
7.4	Updated system documentation must be supplied whenever substantive modifications are made to the system	2
7.5	All system documentation must be supplied in softcopy format.	3
7.6	The system manager must provide level-2 support (functional application support)	3
7.7	The system must support "hotline.	3
	Security	
8.1	Access to the system must be controlled by user name and password	1
8.3	The system must provide user control: authorized access to the social tools, games, exergames etc. (security levels, audits, groups)	1
8.4	Users must be automatically logged-off after a period of inactivity.	1
8.11	Authentication of a registered user must require input from a	4

	remote control only.	
8.19	The device must be protected against computer viruses.	4
8.20	The registration of a new user must be possible in a web application or by phone.	4
8.21	The system must make sure that data are transferred between client and server: the client must poll the server for information in order to avoid problems with firewalls.	4

	Etc.	
9.7	The system must have an open design. This means it can be part of another Web-portal or link to services of other Web-portals, offering additional commercial and non-commercial services like streaming of exercise videos or shops linked to payment services and other health research systems.	4
9.8	The system must include a Web browser.	4

6 Appendix A - All submitted requirements

This appendix gives a list over all requirements that have been mentioned in connection with the work of this deliverable. Not all belong to the platform, and they will be taken care of in their respective work packages. More requirements will be added, some might be removed and many will change priority during the project. The fate of the requirements also partly depend on the results in pilot trials.

Table 1 All initial requirements

Id	Overall
1.1	The system must be easy-to-learn/self-explanatory/obvious in use for an average senior user
1.2	The system must make consistent use of terminology, abbreviations, formats, titles
1.3	The system must present the social tools/games/exergames in an integrated and uniform view to the senior user
1.4	Screen layouts, keyboard functions and navigation must be consistent across the system.
1.5	Date format must be dd/mm/yyyy
1.6	The system must be quick and easy to operate
1.7	It must be defined what resources the system must avoid monopolising
1.8	It must be possible to include audio elements (music, ambient sounds)
1.9	Data formatting must be consistent between entry and display.
1.10	The system must accommodate future increases in the number of users, database sizes and processing loads without redesign or coding and with minimal degradation of performance.
1.11	There must be no duplication of data entry.
1.12	Date/Time must default to local time.
1.13	Menus must be grouped and organized according to functionality.
1.14	Most commonly used functions must be quickly assessed in the fewest number of steps (e.g. during game play - pause game, increase/reduce volume) .

1.15	The system must make consistent and effective use of colour.
1.16	The system must be usable “offline”, i.e. when the user portal is not connected to the internet.
	Senior
2.1	The system must address hearing impairment of senior users.
2.2	The system must address reduced fine motor skills of senior users.
2.3	The system must address visual limitations of senior users.
2.4	The system must support multilingual environments.
2.5	The system must be affordable for the average European senior.
	Social tool
3.1	A user must be able to look up another on-line user’s interests
3.2	A user must be able to chat to fellow on-line users.
3.3	Other users must be able to see if a user is currently involved in a game.
3.4	The social tools must offer a virtual community to socialize and meet others.
3.5	The social tools must offer the possibility to interaction with family, including grandchildren.
3.6	The social tools must provide user profiles.
3.7	The social tools must support a search facility for one user to find an known friend.
3.8	The system must provide access to social tools: (start/stop, storage of user profiles).
3.9	Media sharing with others must be possible.
3.10	Moderators must be enabled to set up user groups and send invitations for activities.
3.11	The social tools must support a videoconference system.
3.12	When more than one chat is open for the user it must be easy to distinguish between them.
3.13	Easy intuitive email system must be supported.
3.14	The social tools must support calendar where activities can be set up.
3.15	Videoconference system. Transcoding or Continuous Presence solution

	for multi-user videoconferencing ¹⁹ .
3.16	Videoconference system: Fast generation of H264 (mpeg4) or H263 video streams. ²⁰
3.17	Videoconference system: SIP communicator for flexible initiation and administration of calls.
3.18	Videoconference system: Video cameras with microphone and excellent echo cancellation.
3.19	It must be possible to have a helper or representative (e.g. for defining friends and setting up of the user profile).
	Game
4.1	The system must keep previous game results for the user.
4.2	The system must provide access to the games/exergames. (start/stop, storage of gaming points, log of progress)
4.3	Game score cards must be accessible from all participating user.
4.4	Identification fields on the score cards must be automatically populated from the users profile.
4.5	The system must be able to present a list of previous score cards in date order (most recent first) and allow the user to (optionally) select one to view in detail.
4.6	The system must keep record of how many features or commands were used.
4.7	The system must keep record of the percentage of tasks the user completed.
	User environment
5.1	The user environment must allow for “alternative” (non-keyboard, non-mouse) input devices (e.g. voice or remote control).
5.2	The user portal must be quickly reset in case of lock up or data corruption and returns to the current state.
5.3	The screen must be easily readable at night or in lowlight conditions.
5.4	The user portal must not interfere with any electronic equipment that

¹⁹ This allows each endpoint connecting to be a different speed or video algorithm. The video stream is generated by the video cards and transcoded to whatever format each endpoint is using. Transcoding screen layout is one site only whereas Continuous Presence can have all the different screen layouts available.

²⁰ Comment: Current high end STBs offer two H264 video-encoders allowing the display of two communication partners.

	might be in the users home i.e. medical equipment.
5.5	The screen must be easily readable in direct sunlight.
5.6	The activity controllers must have low battery indication.
5.7	The system must be able to communicate with activity controllers like Wii Remote, Kinect, HilcrestLabs FreeSpace
5.8	The system must be able to receive signals and data from controllers, received from wireless receivers or via cable. <i>Remark: The interfaces to the drivers will depend on the controller type, wireless standard and if available from the profiles.</i>
5.9	The system must only use solid-state technology for memory/storage (tablet/laptop devices can fall from exercise bikes during biking games).
5.10	The user must be able to set the screen contrast and backlight the user.
5.11	The user must be able to set the screen orientation (landscape/portrait).
5.12	The user portal must be fitted with a camera. The camera must have good-quality optics so as to capture high-quality images.
5.13	The user portal must be resistant to cleaner and wipes.
5.14	The user portal must be fitted with a touch-screen.
5.15	The user portal must have multiple means of connecting to the internet Wi-Fi 3G LAN.
5.16	The user portal must incorporate 2+ USB ports.
5.17	The user portal must incorporate a Bluetooth transmitter/receiver to allow connection to wireless sensors.
5.18	The user portal must incorporate an infrared port.
5.19	Data must be stored locally on the user portal and automatically transferred to the back office system whenever network connectivity is established.
5.20	The system must be accessible through multiple devices.
5.21	The system must run seamlessly on different hardware configurations.
5.22	The user portal must be able to operate in the absence of any network connectivity.
5.23	The user portal must be splash-proof to accommodate sweat and possibly refreshments spill while user is participating in exercise games.

Operability	
6.1	The system must be stable
6.2	The need and cost of maintenance must be low.
6.3	The system must be robust and not easy breakable in a home environment.
6.4	All system maintenance activity must take place at an agreed time of low system use (e.g. early Tuesday morning).
6.5	It must be possible to apply system updates remotely.
6.6	There must be no more than 45 minutes unscheduled down time in any month.
Documentation	
7.1	There must be comprehensive and meaningful user documentation to assist the users in getting the best out of the system
7.2	The user must have a mechanism by which they can suggest modifications and enhancements to the system
7.3	Two levels of error messages must be provided: error and warning.
7.4	Updated system documentation must be supplied whenever substantive modifications are made to the system
7.5	All system documentation must be supplied in softcopy format.
7.6	The system manager must provide level-2 support (functional application support)
7.7	The system must support "hotline."
Security	
8.1	Access to the system must be controlled by user name and password
8.2	The system must ensure that only authorised users access personal data/ protect personal data or identity from mis-use.
8.3	The system must provide user control: authorized access to the social tools/games/exergames.
8.4	Users must be automatically logged-off after a period of inactivity.
8.5	Only authorized staff must be able to print a user profile.
8.6	It must not be able to unload data held on the user portal - it must only

	be transferred electronically to the back office system.
8.7	System audit utilities must be included.
8.9	A user must be able to be a member of more than one user group.
8.10	A user must only have one security level within a user group.
8.11	Authentication of a registered user must require input from a remote control only.
8.12	Authentication with a four digit pin or any other simple process must be sufficient.
8.13	Data attribute (field) level security must be provided.
8.14	Functional security by user group must be provided; e.g. only users sharing certain attributes (roles or business functions) have the ability to perform a particular system function.
8.15	Functional security by user must be provided; e.g. only specified users have the ability to perform a particular system function.
8.16	Maximum password age: 45 days.
8.17	Minimum password age: Immediately.
8.18	Password uniqueness: 20 cycles.
8.19	The device must be protected against computer viruses.
8.20	The registration of a new user must be possible in a Web application or by phone.
8.21	The system must make sure that data are transferred between client and server: the client must poll the server for information in order to avoid problems with firewalls.
8.22	User security/access levels must be supported for individual system functions (e.g. read only, update certain fields, etc.) .
	Etc.
9.1	Data required for later analysis must be mandatory.
9.2	It must be possible to port the system to another social networking ecosystem.
9.3	Text and data must be aligned and sorted for efficient navigation and entry.
9.4	The Join-In group must provide database schematics that allow third parties to develop data extracts as required.

9.5	The system must define how long it must take to complete a particular task.
9.6	The system must define how long it must take to learn a particular task.
9.7	The system must have an open design. This means it can be part of another Web-portal or link to services of other Web-portals, offering additional commercial and non-commercial services like streaming of exercise videos or shops linked to payment services.
9.8	The system must include a Web browser.
9.9	The system must provide mechanisms to integrate its information with that held in other systems to create large data sets for health research.
9.10	Web Services based on SOAP messages and a WSDL file must be an option.

7 Appendix B – HbbTV and CE-HTML

Hybrid Broadcast Broadband TV (**HbbTV**) is both an industry standard and promotional initiative for harmonizing the broadcast and broadband delivery of entertainment to the end consumer through connected TVs and set-top boxes. For the purpose of delivery of broadcast TV and broadband TV to the home, through a single user interface, creating an open platform as an alternative to proprietary technologies. An essential part of HbbTV is the CE-HTML standard.

Consumer Electronics HTML (**CE-HTML**)²¹ is a language for creating user interface pages for televisions. These CE-HTML pages are typically placed online and are based on a 10-foot user interface for easy control from a distance. It is profiled on XHTML and associated standards with special CE-HTML extensions. CE-HTML is part of the CEA-2014 standard (also referred to as Web4CE), defined within the Consumer Electronics Association (CEA). CE-HTML is a language for creating user interface pages and applications like Teletext and EPG. It is profiled on XHTML, CSS and JavaScript including Ajax, embedding of Audio/Video content, DOM event-handling (e.g. key events) and still image formats.

This appendix is discussing the best approach for developing games suited for HbbTV set-top-box software.

HbbTV with stripped down Browser

Set-top boxes often use stripped down versions of available browsers. Stripped down versions are available e.g. from the Webkit and the Opera Browser. Taking care of the limited CPU and memory resources in set-top boxes, both browsers cover the HbbTV standard.

In set-top-boxes the supported Java version depends on the SDK of the chip manufacturer. A Java 1.4 engine from ST-Microelectronics is available, but normally stripped down browser do not provide an interface for downloading Java applets and their execution.

Java applets are not old (outdated) although not much used anymore, and may be problematic if there are security questions etc loading problems. The Java 1.4 version however is old and if any Java development is to be done it would be better to do it with the current version. Java is needed if 3D is required but other things could be better to do with technologies like JQuery/ajax.

²¹ <http://en.wikipedia.org/wiki/CE-HTML>

If the limitation to Java 1.4 is required due to the set-top box, a small demo for testing could use the Lightweight Java Game Library (LWJGL)²² which appears to support Java applets.

Opera Presto 2.5 was one of the best performing browsers in set-top boxes. It claims to support HTML5, but evaluations showed that the given Presto 2.5 features was not completely supported in the selected set-top boxes. Such restrictions limit the features of games considerably and increase programming efforts.

The funspot.tv portals offers several games for HbbTV set-top boxes, but evaluations has showed that current set-top boxes allow to play very simple games only.

HbbTV with CE-HTML

With the core browser (CE-HTML) one can build interfaces that are better integrated to the tv screen. For usability that would be a better place for any software development although it might not then be directly compatible with web browsers in other systems.

For development of CE-HTML applications reference devices are listed below. Normally the devices are connecting itself to portals via preinstalled URLs or via URLs transmitted within the AIT (Application Information Table) from the selected broadcaster. There exist two ways for showing content on an HbbTV terminal. Either the portal provider allows establishing a link on its server or the terminal manufacturer provides a menu for adding and editing of URLs.

Examples of CE-HTML Applications

The remote control is the preferred input device. Therefore most of the applications are designed for navigation with a few keys only. Standard browsers may show a differing behavior than HbbTV terminals. Internet Explorer 8 does not work but Firefox 4 and Safari 5 will show HbbTV pages like those describe below.

On some pages additional inputs are possible using the Java Script console of Firebug for Firefox or the Web Inspector for Safari. The console allows code inspection and handing over of special key codes to the application. On some pages marked below it is possible to run `handleKeyCode(VK_RED)` in the console. It shows the RED button function. Other KeyCodes are `VK_UP`, `VK_DOWN`, `VK_OK`,

<http://hbbtv.daserste.de/>

[key (TAB) + mouse]; ARD HbbTV Mediathek

<http://itv.ard.de/ardtext/>

[key]; ARD HbbTV TEXT like teletext

²²http://lwjgl.org/wiki/index.php?title=Deploying_with_the_LWJGL_Applet_Loader_-_Introduction

http://itv.ard.de/ardepg/	[key]; ARD HbbTV EPG reacts to handleKeyCode(VK_RED) and other key codes
http://itv.mit-xperts.com/zdfmediathek/	[key]; ZDF HbbTV-Mediathek reacts to handleKeyCode(VK_RED) and other key codes
http://tv-html.irt.de/hbbtv/br-text/index.php	[key]; Bayerntext
http://arte.tv/redbutton/	shows red button from arte
http://www.hoerbuch-direkt.tv	[TAB-key + mouse]; too complex CE-HTML shop
http://193.104.51.171/HbbTV/TV3_hbbtv_MV/index.php	handleKeyCode(VK_RED) and others
http://itv.mit-xperts.com/mitxpgo/	reacts to handleKeyCode(VK_ENTER) & other key codes
http://funspot.tv/	Offers include CE-HTML games
http://funspot.tv/index.jsp?platform=pc&page=game	free trial for PC
http://www.hbbig.com/	Test portal for HbbTV enabled browsers only

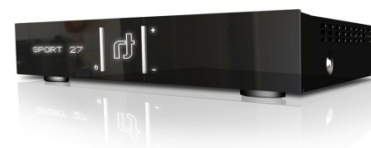
CE-HTML Web Platforms, Portals and Content providers

www.accedobroadband.com	World of APPs to IPTV including funspot
http://wdwip.tv/	Welt der Wunder Video Streaming platform with CE-HTML UI
http://www.ping247.de	e-commerce solution for HbbTV terminals, Personalization of applications

HbbTV Terminals with CE-HTML Browsers

Inverto Volksbox

Information available at www.inverto.tv



Price: < 170 € available e.g. from Amazon

Menu in English, German and a few other languages

A WebEdition with Webkit Browser and a remote control with a small keyboard on the rear are available.

Adding a user link is possible.

Eviado Set-Top-Box

Information available at www.eviado.com/en/



DiscVison prepares the software for eviado.

Adding a user link is possible from the user interface but links are lost after power off.

A pre-installed link to the Join-in (HbbTV) portal is available for Join-In partners. The eviado box will immediately display the corresponding Website after starting the Web Application.

The price of the eviado one box is about 300 € net currently.

Hybrid Set-Top-Boxes available in the market

Humax iCord HD series, NetFront browser from Access, no user links

iCord HD+ portal

Technisat ISIO series starting from 360 €, adding and editing of users links possible
 ISIOLive! Portal from Technisat

Other STBs and TVs are available from CreNova , Philips (Net TV), Smart Electronic, TechnoTrend (Astra HD+ portal), Vantage, Vestel, VideoWeb, Xtrend (Linux)

The future of HbbTV

About 100 manufacturers are currently testing devices for HbbTV version 1.x www.hbbtv.org . Many European broadcasters are introducing HbbTV. UK is focused on Freeview and Italy on mhp.

Features for next generation HbbTV 2.0 are currently under discussion:

- Adaptive Streaming
- Multiscreen (iPad, GalaxyTab, ..)
- Synchronization of TV and broadband video

- Improved graphics and animation
- Loadable fonts
- Better integration of digital rights management (DRM)
- HTML5

8 Appendix C Open game controllers and motion sensors

This is a list of “open” activity sensors and game controllers that can be considered for the Join-In project.

1.1. Blobo

Forget gamepads, joysticks and remotes. Blobo is an incredibly fun PC and Mac gaming experience: this golf ball sized device is packed with future sensor technology and works with every modern PC. <http://www.bloboshop.com>



Openness: Ball-it Releases Unity3D-compatible Blobo SDK for PC and Mac Date: 11/15/2010

Blobo SDK provides access to advanced sensory and environmental data, including:

- 6 degrees of freedom motion and orientation data
- Advanced analog squeeze and tap functions
- Mapping real world coordinate system, based on magnetic north and gravity

Available platforms:

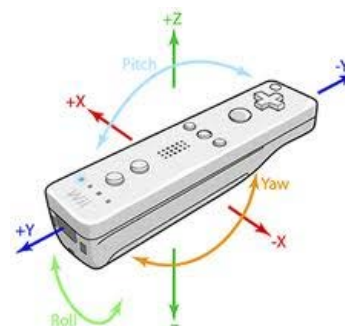
- Native win32/Windows
- Unity3D Windows / Mac

Price: 60 euro

1.2. Nintendo Wii remote

The Wii remote is a game control for the Nintendo Wii that features:

- Bluetooth connection.
- Three accelerometers for calculating the pitch yaw roll of the device.



and

- Seven buttons and a control cross (adding 4 more buttons)
- An infrared cam capable of distinguish four infrared spot in a 1024x768 pixel plane.
- A vibrator, a speaker and four leds.

This device has become very popular within the hacker community, since it offers all of the described features for only \$40 approximately.

Other devices can be plugged to the Wii remote augmenting its capabilities. The most common is the Nunchuck, a device to hold in the other hand that features another three accelerometers, two buttons and a control cross (4 more arrows). A Wii Motion Plus sensor improves the accuracy of the system by measuring angles.

Openness: Due to its wide popularity, the Wii remote has been the focus of attention of the hacker community since it was launched. The communication protocols of the Wii remote and its hardware architecture has been reverse engineered almost completely. Using this information, many people have programmed libraries and application that make use of the Wiimote from a personal computer, many of them available at <http://wiibrew.org>. Others have developed applications, using those libraries, which make a creative usage of the Wii remote beyond of it initials specifications.

The most common way to use the WiiRemote is to hold it in the hand and use the hand's movements to control the game. Another approach is to use the WiiRemote an infrared camera and track the infrared sources movements. For example, the WiiRemote can be next to the user and track infrared leds that are placed in the user's fitness device.

Price: approximately \$40

1.3. Nintendo Wii board

The Wii board is another control for the Nintendo Wii. It consists on a plain square device where the user stands up. The four zones in which the device is divided contain weigh sensors to measure the user weigh and (derivate from that data) the user movements and equilibrium abilities. This device uses Bluetooth for wireless connections as well.



Openness: there are open source APIs to access data from the device.

Price: around 50 Euros

1.4. iPhone and Android smart phones

Both devices have built in motion sensors that a mobile application (apps) can exploit in order to act as a controller interact with games on a bigger screen. They also tend to incorporate a compass that can be used to track the direction it points. Bluetooth will be the communication channel.



Openness: these mobile phones provide open APIs to access the hardware resources. However, they may put limitations on the distribution of software (apps) their on-line app stores.

Price: 300-700 Euros

1.5. Dancing Mats

Dancing mats are the main interfaces for dancing games where the users need to step on different parts of the mat according to the music performing aerobic exercises.

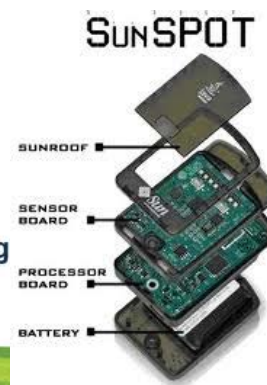


Openness: many of the dancing mats have open source libraries for Linux.

Price: around 20 Euros

1.6. Development mote with accelerometer

There are many development platforms for wireless sensor networks that incorporate accelerometers. For example, many



commercial motes are using <http://www.tinyos.net/>, also there are motes with Java (e.g. <http://www.sunspotworld.com/>).

Openness: completely open most of them, but the only provide low-level APIs.

Prize: 120 Euros

1.7. Inertia Technologies ProMove-2

An example of motion sensors and wireless sensor networks are the ProMove motes from Inertia Technology in the Netherlands (<http://inertia-technology.com/>).



ProMove-2 is a highly miniaturized inertial sensor node that captures and communicates wireless full 3-D motion and orientation information. It combines the latest advances in MEMS sensor design and low-power wireless communication. The sensor information consists of: 3-D acceleration, 3-D turn rate (gyroscope) and 3-D magnetic field intensity (compass). The data is communicated using the 2.4 GHz wireless radio to the FastGateway, which connects through USB to a computer. Multiple ProMove-2 nodes can form a network and report the sensor data fully synchronized to the FastGateway. The sampling rates scale with the number of nodes in the network, e.g. 200 Hz for 6 nodes, 100 Hz for 12 nodes.

The ProMove-2 is the motion sensors used in the AAL project IS-ACTIVE (<http://www.is-active.eu>) where Norut and Norwegian Center for Telemedicine participates. For IS-ACTIVE we use a ProMove-2 that also supports Bluetooth directly on each sensor node. IS-ACTIVE will use the Promove-2 as sensor and controller for applications aiming at increased daily activity and physical exercising.

Openness: The protocol is open

Price: N/A

1.8. Cube-Game

Using hardware similar to the wireless motes previously described it is possible to built small computer devices with LCD



screen and sensors. These can be used to build advanced pervasive games.

The product is not yet in the market.

<https://www.sifteo.com/product>

1.9. Kinect

Kinect for Xbox 360 is the latest game controller launched by Microsoft. Using advanced artificial vision technologies it is capable to recognize 3D movements with the highest precision by just using its camera.

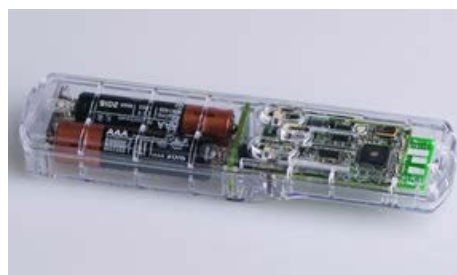


Openness: Microsoft has not made any attempt to limit the use of the technology for other purposes and the open source community has already developed open source APIs for the device. However, those APIs are only available for windows machines.

Price: 120 euro

1.10. Freespace / Hillcrest Labs FSRK-BT-1 Reference Kit (bluetooth version)

This reference design incorporates a complete six-axis Freespace® motion solution integrated with Bluetooth® RF. To simplify evaluation, the reference kit includes a remote control packaged in a small form factor. The removable module in the kit includes the [Freespace MotionEngine](#) and microprocessor together with MEMS inertial sensors, and supports prototyping with other form factors.



The FSRK-BT-1 can be used with the [Freespace MotionStudio](#) and [libfreespace](#) for performance evaluation, software development and device configuration.

<http://hillcrestlabs.com/products/freespace-tools.php> Freespace Motion Studio free download
<http://libfreespace.hillcrestlabs.com> Open Source Software

Price: \$119 for FSRK-BT-1 plus about \$100 for delivery to Europe

1.11. Loop Pointer / Hillcrest Labs

The Loop™ pointer is the in-air mouse that moves your cursor with natural hand motion. Simply point and click on the TV that's connected to your PC or Mac. Use the Loop pointer just like a mouse while you browse the Internet or view your own media content. The distinctive ergonomic design has just four buttons and a scroll wheel.



The Loop Pointer is sold in the US only (currently).

Openness: Controller development kits are available also from Hillcrestlabs
<http://hillcrestlabs.com/products/freespace-reference-kits.php>

Price: \$49 for Loop pointer only

1.12. Sixense Truemotion

This Sixense truemotion is a motion based controller that is essentially a refinement of the Nintendo Wii remote. It offers six degrees of freedom motion tracking and uses a magnetic field for tracking position and orientation. Unlike the Wii remote, line of sight is not required and it would appear to be targeted towards PC users.



<http://www.sixense.com/>

Openness: Development kits are available from Sixense, but it is not clear whether non-commercial game developers can avail of them.

Price: Not on sale yet but scheduled for release in 2011.

1.13. OCZ Neural Impulse Actuator (NIA) Game Controller

This product was released in 2008 and is a headset that translates facial expressions, eye movements, brain Alpha waves and Beta waves into control signals. The box that performs the signal processing is connected to the host PC via USB, so the user is effectively tethered to the PC.

http://www.ocztechnology.com/products/ocz_peripherals/nia



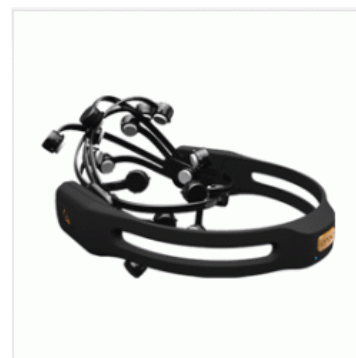
In our lab tests this device is hard to use and is not a real solution for the elderly.

Openness: An SDK was released in December 2010 and is freely available.

Price: €100 approx.

1.14. Emotiv Epoc Neuroheadset

The Emotiv EPOC is a high resolution, neuro-signal acquisition and processing wireless neuroheadset. It uses a set of sensors to tune into electric signals produced by the brain to detect player thoughts, feelings and expressions and connects wirelessly to most PCs. The headset has a built-in gyroscope to generate positional information for cursor and camera controls.



<https://emotiv.com/index.php>

The Emotiv Epoc is available to buy from the US only.

Openness: A researcher edition SDK is available for \$750.

Price: \$299 excluding shipping costs.

1.15. Playstation Move

The Move is a combination of two devices; 1) a handheld motion controller with a sphere that lights in different colours at the top and 2) a video camera, known as the Playstation Eye camera, that tracks the movement of the controller through the lit sphere. The Move is designed for the Playstation 3 game console. The Move controller includes a three axis gyroscope, a three axis accelerometer and a terrestrial magnetic field sensor, as well as a colour-changing sphere that is tracked by the PlayStation Eye camera.



The Move controller also includes several buttons that replicates some of the functionality of the Playstation dualshock 3 wireless controller (a separate product).

<http://us.playstation.com/ps3/playstation-move/product-information/>

Openness: The open source community is currently working on APIs for this device for the PC.

Price: €60 approx for one controller and the eye camera.

1.16. Genesis Game Controller

Genesis has developed special controllers, that are individually labelled (e.g. Dots) according to the keys on the screen, allowing to play the game without being forced to use mouse or keyboard.



A modular keyboard, which allows for any combination of the single keys (9cm x 9cm) e.g. in one row or as a cross, was developed. One is free to play with five, two or even with only one key, depending on the abilities of the person playing. The controller is constructed in a way that resists even strong beats.

As a second alternative a flat keyboard (35cm x 10 cm) is available, that responds to light touches. The threshold for reactions is adjustable. It is constructed with as a closed device with a flat surface made of Plexiglas, therefore it is easy to clean.

Up to four keyboards can be connected to one computer via USB, that allows that up to four players can play games together.

<http://www.efi.fh-nuernberg.de/genesis/>