Post mortem of Terraformers

a 3D game for blind and sighted

Thomas Westin

thomasw@dsv.su.se
# Table of Contents

Introduction.................................................................................................................. 4
Result............................................................................................................................... 4
Overview......................................................................................................................... 4
The 3D world.................................................................................................................. 5
Accessing the 3D world............................................................................................... 7
Accessing the Physics................................................................................................. 8
Accessing Details.......................................................................................................... 9
  Voice feedback......................................................................................................... 9
  Menu system............................................................................................................. 9
  Game objects............................................................................................................. 9
  Using game objects.................................................................................................. 9
  Conversations.......................................................................................................... 9
  Atmosphere and mood............................................................................................ 9
  Describing directions............................................................................................... 10
  Game state feedback............................................................................................... 10
Accessing Overview..................................................................................................... 11
Accessing Breakdowns and Support.......................................................................... 12
Accessing the Installation and Game Shell................................................................ 13
  The installation program......................................................................................... 13
  The Microsoft Windows Start menu options....................................................... 13
  The Player Documentation...................................................................................... 14
  The Game Start menu options.............................................................................. 14
  The License Key....................................................................................................... 14
  Introduction and cutscenes...................................................................................... 14
  The Game Loader................................................................................................... 15
Accessing the Web site and Payment system............................................................... 15
The Keyboard layout.................................................................................................... 15
Accessing the Game Environment.............................................................................. 16
  Learning the interface with an in-game tutorial..................................................... 16
  Powersuit.................................................................................................................. 16
  Menu system............................................................................................................. 17
  Sound Compass....................................................................................................... 18
  Sonar.......................................................................................................................... 19
  GPS.............................................................................................................................. 20
  Direct orientation..................................................................................................... 23
  High Contrast Mode............................................................................................... 23
  No 3D Graphics Mode........................................................................................... 24
  Backpack.................................................................................................................... 24
Accessing Game Objects.............................................................................................. 25
  Avatar....................................................................................................................... 25
  Powersuit.................................................................................................................. 25
  Powerup stations...................................................................................................... 25
  Sound keys............................................................................................................... 25
  Keyholes.................................................................................................................. 26
  Tone wheels............................................................................................................. 26
  Tone emitters............................................................................................................ 26
  Code locks............................................................................................................... 26
Doors........................................................................................................................................27
Bazooka......................................................................................................................................27
Ammunition...............................................................................................................................27
Laser gun....................................................................................................................................27
Battery.......................................................................................................................................27
Communication devices............................................................................................................28
Robots......................................................................................................................................28
Parts for the main computer.....................................................................................................28
Walls..........................................................................................................................................28
Discussion...................................................................................................................................29
Balancing the game for sighted and blind..............................................................................29
Aesthetics versus dynamics.........................................................................................................29
Writing a linear story for interactive, accessible games..........................................................29
Naming standards......................................................................................................................30
Issues with the runtime environment ......................................................................................30
Project resources versus scope of the game ...........................................................................30
Future actions for the industry..................................................................................................31
Further work...............................................................................................................................32
  IGDA GA-SIG..........................................................................................................................32
  Comparative studies of sighted and sight disabled gamers playing the same game ...........32
  Enabling game designers.......................................................................................................32
Literature......................................................................................................................................34
Appendix......................................................................................................................................35
  Presentations of Terraformers...............................................................................................35
  Community web sites............................................................................................................36
  Sites with Accessible Games:..............................................................................................37
Glossary.......................................................................................................................................40
**Introduction**

This description of the game Terraformers was made shortly after the release of the game while I had all the details about the development process. You may read it as a extensive post mortem to learn what to consider when making a 3D game accessible for blind and sighted.

**Result**

To fully understand the presentation of the result, I strongly suggest you download and play the free demo from the website, [www.terraformers.nu](http://www.terraformers.nu)

The first sections of the Result describe the accessibility issues in wide terms, e.g access to the 3D world and it's physics, getting an overview as well as details etc. The later sections describe in detail the methods we developed during the project to solve those issues in a practical sense. I also describe the technical implications in doing this.

**Overview**

Developing a 3D game demands in itself technical innovation in many areas like 3D graphics and other media optimization, game physics, game AI and so on. Here I will focus on the sound interface and how it relates to the graphic interface, since this combination is the most original aspect of Terraformers. However, the issues of optimization, physics, AI etc will be discussed to some extent as they are important parts of making the game world accessible.

The game is a visual / audio hybrid playable by both sighted and sight disabled / blind. The following features describe how the 3D world is made accessible to sight disabled. Some features allow direct access to the 3D world through the use of 3D sound (i.e spatial sound), other features allow access through voice feedback. Also, all game objects are accessible in a similar way (not only the 3D world space).

The game is an adventure style game in a first person view. Your mission in short is to find out what has gone wrong on a planet where mankind is trying to survive by creating a breathable atmosphere. This process is called “terraforming”. Game features include pick up, equip and use objects to/from an inventory. You can fire different guns, unlock doors, attack/be attacked by NPCs, talk to NPCs. Other challenges include avoiding NPCs who fire short distance weapons at you and keeping the energy level in the “Powersuit” above zero.

Here I will present the features of the 3D world in Terraformers, and how we have made it accessible and why the game and the game interface was designed in the way it is.
The 3D world

Environments range from indoor bunkers to outdoor mountain areas to underwater areas. The 3D world is divided into levels, which in turn are divided into areas. All areas are delimited by doors for a number of reasons.

With the 3D engine used (Shockwave3D), the concept of doors provide a convenient way to keep the polygon and model count down to a minimum, resulting in much improved performance. This is also true for the physics engine, the 3D sound system and the game engine itself.

Discreet areas also provide a reasonable amount of information to be presented for the user at a certain moment. If the gamer use the GPS for instance, only the important objects in the specific area are presented; listing all objects in the entire game world would take a long time to step through every time.

Overview of the Tutorial level used in development (translated from Swedish). Numbers indicate areas, and thin lines indicate doors.

On the next page follow a few screenshots to give you an idea of the 3D world rendering.
<table>
<thead>
<tr>
<th>Image</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.jpg" alt="A Terraformer robot" /></td>
<td>A Terraformer robot</td>
</tr>
<tr>
<td><img src="image2.jpg" alt="An outdoor scene in the Terraformers mining area" /></td>
<td>An outdoor scene in the Terraformers mining area</td>
</tr>
<tr>
<td><img src="image3.jpg" alt="An flooded, underwater area" /></td>
<td>An flooded, underwater area</td>
</tr>
</tbody>
</table>
**Accessing the 3D world**

The features for accessing the 3D world through sound are the result of a lot of prototyping, testing, programming and re-programming during the entire project. We aimed at making the 3D world accessible with a free movement in a 3D realtime graphic and physically simulated world. I.e. we avoided making a tile-based movement system which would have been easier to make accessible, like “press the key X times to reach the other end of the door”.

We wanted to give the blind gamer the same resolution of avatar control as the sighted gamer, and have similar features as in a regular mainstream game, as far as possible. Also, we wanted to make the game a realtime experience, not a turn-based one (as in chess) which would have simplified many things.

The following is an overview of the most important features:

- **Sonar**: getting an audio perception of the 3D surroundings and details of what is in front of you, including feedback about when you are aiming at an enemy target. This also helps the blind gamer from bumping into rigid bodies such as walls or doors.

- **GPS**: provides number and type of objects in an area, as well as their positions in the 3D world and your current 3D position in coordinates of north-south, east-west. This provides an overview of important objects and where you can find them.

- **Sound Compass**: provides a perception of where north is, using a discrete 3D sound as well as spoken feedback on demand.

- **Direct Orientation System**: this feature orients the avatar directly in eight directions (North, NorthEast, East, SouthEast, South, SouthWest, West, NorthWest) using the numeric keyboard where 8 represents north, 9 north-east and so on.

- **High-Contrast mode**: this feature let gamers with low vision (i.e. not blind) access the 3D-graphics of the game world.

These features and others are described in more detail in the *Game Accessibility Methods* section below.
**Accessing the Physics**

Sliding along or bumping into objects are two different events that occur in a physically modelled 3D world, such as in Terraformers.

We used a commercial, award winning physics engine (Havok). However, we did not have a **physical modelling audio** engine at our disposal. Havok provides functions for forces such as gravity etc, but to present the effect of forces on different surfaces with audio, you need a synthesizer that dynamically create the audio depending on the amount of force applied, and depending on different types of materials involved in the collisions. Today this is an emerging field (e.g *Staccato Sound System* or *Phya*) but we only had a simple audio engine based on DirectSound 3 in Macromedia Director where we could play back and queue sounds that were prerecorded, and use DirectSound 3 features (i.e very basic 3D sound features)

The events of sliding vs bumping into can not be decided in a discreet way, rather it is a gradual transition between those two states. With a physical modelling audio engine, this gradual transition could have been represented directly with audio. Since we lacked this technology, the main difficulty was to determine when to give feedback that tells the blind gamer which type of event is happening. This took much time of tweaking the settings for playing back audio in a way that was satisfactory, in terms of how much do you have to slide to play a sliding sound instead of a bumping sound.

Simply playing back one audio file for the bumping sounds gave a quite static impression, so we have a function that randomly plays different audio files, with slightly different bumping sounds. This gives a perception of bumping into objects in a varied, more natural manner.

The sliding sound also need to present information of which direction the object was, e.g if you slide along a wall to your left, the sliding sound needs to come from the left. Hence we use 3D sounds to play back the sliding sound.

Bumping is triggered by colliding into a rigid body with an angle greater than the value set as condition for bumping. When sliding is triggered, it also uses a ray casting to the left and to the right of the avatar. The closest target is chosen and the position of where the ray hits the target is used to position the 3D sound of the slide sound.

To avoid the perception of floating forward/backward, hearing your footsteps is important in a first-person game. Also, to get the correct perception of the surface material, the sound must vary accordingly. Footstep sounds are implemented by playing a queue of four different and separate step-sounds (non 3D) in an order which gives the impression of some variance. It is triggered by a collision callback in the Havok physics engine.
**Accessing Details**

**Voice feedback**
Recorded actor speech was a choice we made from an aesthetical point of view. It would have been much easier to develop the game with speech synthesis, since changes in the gameplay or the graphics had severe consequences for the spoken feedback, which was re-recorded many times. To further improve the aesthetics we also wanted to use external actors instead of our own voices, which added to this problem, and also became expensive and time-consuming.

**Menu system**
Menus are used for many things both in the game and in the game shell. The same system is used for all different tasks, which took a long time to develop but was definitely needed. It handles hierarchical menus, queued voices and sound effects and any type of front-end interface. Selecting a menu item can represent settings in the game shell, using a game object in the inventory, getting spoken feedback from the GPS and so on.

**Game objects**
These can be found directly in the game world through object-type unique 3D sounds, as well as by using the GPS or Sonar. Game objects are mostly picked up by walking into them, with a clicking sound followed by spoken feedback telling you the type of game object. When picked up they can be found in an inventory, where they are sorted into different categories accessible through a visual menu with spoken feedback in combination with game object sound effects (same sound as the objects’ 3D sound). Game objects can then be used and put back into the backpack.

**Using game objects**
If the gamer tries to use an object like a key / a weapon too far away from a lock / an enemy, spoken feedback informs about this. You can also check what object you are holding in your hand, and of course put the object back into the backpack.

**Conversations**
These are done with “communication devices”, a simple type of Non Player Character which play back prerecorded messages which give the gamer directions of what to do in the game. This feature also use the Menu System.

**Atmosphere and mood**
The accessibility features makes the 3D world and physics accessible for blind, but the feedback from the GPS, Sonar and Sound compass etc are very concise and don't give any sense of the “mood” or “atmosphere” of the environment. To counter this, we worked hard to implement simple but effective, context based sounds like cross-faded ambient sounds, walking sounds and a spoken description of how things look, feel, smell etc. When designing these features we also thought a lot of how these features would also give the sighted gamer a better gaming experience.
Describing directions

we had to think about how the spoken environment descriptions would work in conjunction with the interactive 3D world, e.g not using relative notions such as "left" or "right", but absolute directions e.g "east" or "west".

Game state feedback

This is mostly presented by spoken feedback like when the avatar is hit by enemy fire, when the powersuit is low on energy or when the avatar is close enough to use an item.
Turning off 3D sounds: When you are standing close to an object with a 3D sound source attached it becomes loud since you also need to hear it from a great distance for navigation. This can make it hard to hear voice feedback, when using the GPS, Backpack, Sonar etc.
**Accessing Overview**

When a menu is presented to a user, you want to present an overview, ready-to-hand. A sighted person can see the entire menu at once and quickly get a perception of the number of items in the menu. To give a blind gamer a quick overview, we came to the conclusion that there are basically two possible options. One is to have a voice saying e.g \"3 items\". The other is to have some sound effects, e.g three beeps.

We had a general strategy of avoiding spoken feedback when possible, since we thought it took too much attention from the game play – you have to focus on what is said rather than the game. Also, too much speech presented the problem of either queueing a lot of speech (resulting in feedback too late) or having multiple speech feedback simultaneously, resulting in making it hard to hear.

The menu system is used for overview in the GPS system. When you bring up the GPS heads-up display, it presents all important items in that are where you are at the moment. (See the GPS section for more information).

Overview is also provided through the use of 3D sounds, i.e spatial sounds representing different types of objects, e.g doors, robots which has a distinct location.

Overview of walls presented a specific problem, since they are widespread and there is no point in positioning a 3D sound at the center of the 3D wall mesh (i.e at the pivot point). Instead we implemented the Sonar which positions a 3D sound at the closes object in the direction you are currently facing. (See the Sonar section for more information).
**Accessing Breakdowns and Support**

One specific experience I got during development and pre/post release support was that there are also a problem for blind accessing a break down, i.e when the game doesn't work properly. In cases where a standard Windows component presents information the information is accessible through the standard accessibility tools used by blind, e.g if a "Script error" dialogue box appears.

In other cases a blind person have a hard time finding out what is really going on. To add to the complexity, the developer has to try and understand an error description which is describing the issue from a blind person's perspective.

An example of this is the following scenario that happened:

The blind gamer X is playing and suddenly all 3D sounds in the environment gets silent. X also tries to walk around in hope to get an idea of what has happened. After a while X makes the conclusion that something has gone wrong with either the programming of the game or maybe the sound card drivers. (See the Discussion section for more about issues with the runtime environment).

X then writes to the support with a description similar to the above. The developer has not experienced this himself but tries many times to reproduce the issue (which is a prerequisite to fix the issue). In a long e-mail conversation the developer and gamer then tries to omit possible causes, by making certain sound card drivers are OK etc.

After testing on his own fast, clean install development machine, the developer tries to run the game on an older Windows system with a lot of applications which hasn't been reinstalled from scratch ever. Suddenly the developer finds out that the in-game physics can make you fall through the ground by gravity, if the update is not done properly by a too busy processor. This also cause you to travel fast downwards in virtual space, away from the game environment, where... yes, all the 3D sounds are located!

So, finally the developer implements different features that make certain that it should not happen again, or at least won't leave the gamer falling down through space forever if it happens anyway. A patch is released and this certain issue is fixed.

This issue would have been much faster fixed if the breakdown was accessible somehow. One way in this case would have been to make the GPS system also detecting not only the surface position but also the height position.

Please note that this was not a one time issue; many similar situations have appeared with other problems and fixes.
**Accessing the Installation and Game Shell**

The term "game shell" comprise the setup and start menus of the game, i.e the part of the game software that is handling things outside the game world. It constitutes a framework for the game world. In Terraformers there are the following parts in the game shell.

**The installation program**

The installation program was easy to make accessible. Since it is a regular Windows software it is compatible with assistive technology e.g screen readers with speech synthezis/braille. Hence we did not need to develop any dedicated accessibility features for this software.

The main issue with creating the installation program was to manage patches and also how to implement multiple language support.

**The Microsoft Windows Start menu options**

After installing the game, Terraformers is found under Start / Programs / Terraformers. Since this is a regular Windows interface we did not need to develop any dedicated accessibility features for this.

The options include:

- **Set3D Renderer**: This option sets the method with which the 3D graphic engine renders the 3D graphics: Auto, OpenGL, DirectX5, DirectX7 or Software. The default is Auto.
- **Player Documentation**: this opens a RTF file with text describing how to start and play the game.
- **Start game**: Starts the Game Shell
- **CreateSupportData**: starts a program that outputs data to a textfile about your system used for technical support
- **OpenThePrefsFolder**: opens a folder on the hard drive where user specific files and preferences are stored, e.g the License Key
- **TerraformersWebSite**: opens the default web browser and goes to the Terraformers web site
The Player Documentation

The Player Documentation is distributed as a text file in RTF format, accessible through a regular screen reader. There is no paper version distributed, except for in the CD-version of the game. One reseller have printed a braille version to their clients.

The content is structured in a plain way to make it easy to use for blind. We could of course have made this with HTML and CSS to give it a better look and feel for sighted, but we decided to skip this in order to be able to focus on and finalize the game itself.

The Game Start menu options

The Game Start menu is the first interface the gamer meets. The menu system used here is the same that is used in the game itself, so the gamer need only to learn one menu interface. The main difference compared with Windows menus is that you use Space to select items in the menus. In Windows you can use Space, but you can also use the Return key. In the game though, the Return key is dedicated for opening the Backpack.

In the Start menu, you can Start a new game, Load a previously saved game, set the 3D graphic rendering off, check your License key and exit the game shell to Windows.

Being able to set the 3D graphic rendering off in the start menu is an important feature if you have an old computer without a 3D graphic hardware and with a slow CPU. You can also turn on/off 3D rendering after the game is loaded, but loading the game with 3D rendering on can be a problem on an old, slow computer.

The License Key

The game can be downloaded and played as fully playable demo for free. A license key is needed to unlock the entire game. We wrote our own algorithm for encryption/decryption, which is stored in a text file. This text file is created upon order and is unique to the buyer.

A text file is easy to distribute via e-mail, while the big download is done via the game website (via HTTP). Also, by distributing a user unique text file, it is easy for the client to install who don't need to write any license key by hand, which of course is a possible source of error. Further, we did not need to write any special accessible input program for the license key. In the Game Start menu, there is also an option for checking the license key using speech synthesis. This is the only place where we use speech synthesis; since the key is user unique, there is no way to have prerecorded voice for this.

Introduction and cutscenes

There are two linear stories told which don't allow interaction by the user. First, there is an introduction played every time the game shell is loaded. This can be skipped by pressing any key on the keyboard. There is no information saying you can skip it, but this is a de facto standard in every game that has an introduction. Second there is a cutscene played after you have finished the tutorial on the space station, and you travel down to the planet Tellus 2 to start your mission. This cutscene is only played once and is not replayable, and hence you can't skip it. It would of course be a nice feature if you could play back the cutscene, e.g from the Game Start menu but there has not been enough time to implement such a feature.
The Game Loader

The Game Loader is setup as a FSM (Finite State Machine) which manage the process of loading, saving, going between game levels and quitting the game and exiting game shell back to Microsoft Windows. This process is quite technically complex, since it needs to be able to initialize, setup, run and shutdown many different processes in the correct order on demand. This comprise a 3D graphics engine, 3D sounds, queued sounds, timeout objects, the Havok physics engine and so on. It must be able to handle commands from both the user (load / save / quit) and game commands (e.g Game Over states etc). On top of all this, it must also provide the user with the correct feedback, visually and with voice. By utilizing a FSM we were able to setup state tables where we could delimit the complexity of these processes, so they were manageable to program correctly. FSM:s are also used for other objects in the game, such as non player characters (robots).

Accessing the Web site and Payment system

During the entire development we had a simple web site which was OK from an accessibility perspective, but not optimal. When we planned the release we wanted to restructure the site from being a "project" site to a "commercial game" site, and still keep it accessible enough.

Also we needed to implement some secure payment system with a price model and which was also accessible. We found ShareIt to be a good and secure system with many features, however it lacked some basic accessibility, e.g alt texts on important buttons. We contacted them about this issue, and they responded but it took some months before they actually fixed this. In the meantime we made a step by step description of which buttons to use for what purpose.

An issue that has got to our attention is the choice of colors, red text on black which is a problem for some color blind. There are many things that can be said about the accessibility of the web site and I am aware that we should have put more resources into this long before release. On the other hand, there has been few complaints about it so it seems to be OK although not optimal.

The Keyboard layout

The keyboard layout was designed to keep the most used features ready-to-hand [4] without the need to move your fingers. E.g while moving (both in menus and in the 3D world) with the cursors keys you easily reach the keys for opening your backpack (Return), closing all menu based features (End), going up in a menu structure (Home), putting things back into the backpack (Delete) etc.

With your left hand you can easily reach the Space key for selecting items, while checking the Sonar (B), Compass (C) or your own GPS position (V) etc.

Some gamers have complained about the keyboard layout, rightfully stating it doesn't comply with standards. On the other hand, many games use non standard controls, but there are conventions or de facto standards that have developed in the game industry. We agree that we could have done a better job following those, but since there are many special features not available in other games we found it hard to do so.

One feature we did think of but deferred, was supporting user defined keyboard layouts. The main issue with this are problems of providing support. You could of course refer to the functions rather than the actual keys to press, e.g Turn on the Compass and not saying anything about the actual key to press, leaving this to the user according to the user settings. However, developing an interface for setting all the keys up would take quite some time also, and we were not sure of how many users would actually be using it. Hence we decided to skip this feature.
**Accessing the Game Environment**

The Game Environment represents everything in the game world except Game Objects, see section below.

The user feedback about the Game Environment comprise

- 3D graphics
- Sound effects
- Ambients sounds
- Footstep sounds on different ground materials
- Voiced descriptions of visual as well as other sensory input

Environments range from indoor bunkers to outdoors mountain areas to underwater areas.

Screenshots of the different areas can be found in the *3D World* section above.

**Learning the interface with an in-game tutorial**

The tutorial is set up as part of the game. It is where you get both important objects and training for your mission. The goal is to educate the gamer about how the game works without reading the text documentation (included as a text file, accessible for blind). Most of the interface features are presented in the tutorial.

In the tutorial there more voice feedback giving you feedback when you come close enough to objects to interact with, or when you have taken an object to use, inserted a key, fired at a target and so on. This was an extra layer that was added on late which made it more difficult to implement.

A special issue was whether to allow moving while listening to tutorial voices. As you move you can get to points where a new tutorial voice need to be played, although the previous voice wasn't finished. We thought about locking movement while a tutorial voice was playing, but we wanted to make the tutorial a part of the game and we felt locking movement would disturb the "suspension of disbelief". We chose to recommend the gamer not to move while listening to a tutorial voice.

**Powersuit**

The powersuit contains all the necessary technology to protect the avatar from hostile attacks. In addition, it provides all of the accessibility features described below. A built-in voiced PDA is responsible for miscellaneous feedback from these features. Several features use a menu system (see below).

The use of a powersuit also makes the accessibility features integrated in the game, i.e there is no special accessibility add-on needed to play the game. All gamers need to put on the powersuit.
A screenshot of the first tutorial area where you pick up your Powersuit

Menu system

We put a lot of effort into the menu system since it is used in a variety of ways. Also when we started the project, we did not know exactly what features the menu system had to be able to handle, so it had to be very flexible and modular.

The menu system is used in the game and the game shell to:

- present information as text, recorded voice / speech synthesis and sound effects
- execute commands for starting the game, use an item in the backpack, or get information from e.g the GPS
- organize information in hierarchical structures, e.g the backpack organizes different types of keys in submenus. Menu items containing a submenu are named in pluralis, while other are named in singularis.
- handle conversations with non-player characters, e.g the Communication devices.

To use the menu system, the gamer uses the keyboard. The gamer can

- step up and down in the current menu with the up/down arrow keys
- select an item with the Space key
- get back to a previous (higher) level in the menu hierarchy with the Home key
- close the menu with the End key

Since the menus are dynamically created, presenting information that varies over time, we needed to provide some simple overview of the number of items. We chose to use a soft beeping sound which repeats the same number of times as there are items in the current menu. Submenu items are presented when a submenu is selected.
A screenshot showing a menu instance used by a Communication Device (seen in the background)

Sound Compass

A 3D sound represents north, and a rough 8 direction spoken feedback is available by pressing a key on the numerical keyboard (north, northwest...).

When the avatar turns, the 3D sound beeps giving a hint of where north is. This works no matter if the avatar stands still or walks simultaneously. We chose to use a beeping sound rather than voice feedback to not distract the gamer from the immersion of the game experience.

While a 3D sound beep is discreet and still gives enough information in most situations, you sometimes want a more direct feedback telling you at least roughly what direction you are facing. We chose to use eight directions, which gives a rough direction.

We discussed to use e.g 360 degrees to give more accurate feedback but decided that the eight directions were good enough for the game, where directional precision is not a critical issue. A 360 degree feedback would also make it harder to understand how to use the compass.

A 3D sound is positioned close to the avatar as it moves, in the direction of a virtual north. The sound emits when the avatar turns, giving the player a perception of orientation.
Sonar

A 3D sound gives the gamer a rough perception of the distance to objects in the direction the gamer is currently facing. By pressing a key the gamer can also check what type of object it is (door, wall, robot...). Enemies are automatically told by the PDA voice.

When the avatar turns a 3D sound is positioned at the first object in front of the avatar, using a ray cast in the current direction. The distance to the object is represented by the volume so this gives the gamer feedback about how far away the closest object is. Using this feedback the gamer can avoid bumping into objects. Also, using the Sonar while standing still and turning gives the gamer a perception of the spatiality, i.e the size of the room or area where the avatar is standing, in all surrounding directions.

When the Sonar's raycast hits an object it checks what type of object it is. If it is an important object, the pitch of the 3D sound is set higher. This way, the gamer can scan the area for important and interesting objects to interact with while walking. I.e a wall is not very interesting, but a door or a key are important objects to find and use.

By hitting a key the gamer can get voice feedback about what type of object it is. This way the voice feedback is presented on demand, rather than distracting the gamer with unwanted voice feedback automatically. One exception is when the Sonar's raycast hits an enemy or dangerous object; then voice feedback is automatically provided (if the Sonar is on).

Proxy models are very important for this feature to work properly. I.e some game objects are so small it gets too hard to detect them with the Sonar's ray. Hence we setup proxy models, i.e larger, invisible models as a shell for all game objects. This is also good to do for optimizing collision detection and physics, since the proxy model can use very few polygons which makes the calculations of the collision detection less CPU intensive.

A schematic illustration of the Sonar: a ray is cast in the direction of the avatar's orientation, and a 3D sound is positioned at the point of collision with the first obstacle it hits
**GPS**

A global positioning system (GPS) is used to give the exact positions of objects in an area as well as the position of the avatar. A voiced menu system provides an overview of nearby objects.

The GPS, or Global Positioning System provides feedback in several ways.

- Overview of an area, through a list of important objects in the current area via the Menu System
- Information when entering a new area, through voice feedback, which helps the blind gamer to know when a transition to a new area is done. This is important feedback in order to know when to check the GPS or listen to area descriptions.
- Getting the exact coordinates of important objects in the current area
- Getting the exact coordinates of where the avatar is at the moment
A screenshot with the GPS menu showing the important objects in the area. Selecting a menu item presents the coordinates for the object.
Direct orientation

Using the numeric keyboard the gamer can orient the avatar directly in 8 directions (north, northeast...)

This feature was a direct wish from one gamer who tried the first official demo version.

The directions are mapped on the numeric keyboard where 8 represents north, 9 northeast, 6 east and so on.

A schematic illustration of the numerical keyboard on a regular PC

High Contrast Mode

With this feature turned on, you get parts of the 3D graphics rendered in black and white toon-style which enhance contrast on important objects for low vision gamers.

Table 1. Normal and High Contrast Realtime 3D Rendering

| 3D scene rendered in normal contrast for sighted gamers | 3D scene with important parts rendered in high contrast, for low vision gamers |
No 3D Graphics Mode

The 3D graphic rendering can be completely turned off for blind gamers who don't have 3D graphic hardware installed.

There is also a possibility to set the 3D engine to not use 3D graphic hardware, i.e. it the use the regular CPU only which was a way to try and workaround malfunctioning graphic drivers in Windows.

![A screenshot showing the graphic display when 3D graphics rendering is turned off.]

Backpack

Objects are accessed using a voiced menu system with hierarchies. A main menu represents "slots" where items are stored. Each slot is a submenu where the gamer selects the item to use. Sound effects are also be attached to voices dynamically.

E.g there are keys in the game, so called "sound keys" which you identify by their sound effect. In the backpack, the voice and text says "sound key" and then the sound effect is played. Naturally there are no graphical way to identify a sound key.

In the main menu, all slots (i.e. submenus) are named in pluralis, while all individual items are named in singularis.

The backpack uses the same menu system as the GPS.
Accessing Game Objects

All game objects have voiced feedback and 3D sound icons. Game objects are e.g weapons, ammunition, keys, locks, communication devices, robots, doors, walls and so on. Basically everything that the gamer can interact with.

Below is a list with a brief description of the game objects

Avatar

The Avatar is the gamer's representation in the game world, or Player Character.

Interaction: Arrow keys to move the character

Accessibility features:
  • Walk sound gave the gamer feedback when the character is moving and also about which material he is walking on
  • Sound effects when hit by hostile attacks etc

Powersuit

This suit, provides protection as well as storage in a backpack, and all accessibility features described above (compass, sonar, GPS etc).

Interaction: Picked up by walking into it.

Accessibility features:
  • 3D sound helps the gamer find it
  • Voice feedback and sound effect when putting the suit on.

Powerup stations

Powers up your power suit.

Interaction: Triggered by the Avatar walking into it.

Accessibility features:
  • 3D sound helps to gamer find it
  • Voice feedback and sound effects when trigging a power up

Sound keys

Keys to doors and other things. They emit a unique 3D sound which match a keyhole with the same 3D sound emitter. Sound keys are invisible everywhere (except in the tutorial) and are hence only detectable by 3D sound.

Interaction: Picked by the Avatar walking into it. Inserted into keyholes with Space when Avatar is holding the key, and standing close enough to the keyhole
Accessibility features
- 3D sound helps to gamer find it
- Voice feedback and sound effects when picking up a sound key
- Voice feedback when the avatar tries to put a key into a key hole which is too far away

Keyholes
Keyholes to put sound keys in. They emit a unique 3D sound which match a sound key with the same 3D sound emitter.
Interaction: Distance check to see if the Avatar is close enough to insert a key
Accessibility features
- 3D sound helps to gamer find it and match it with the correct key
- Voice feedback and sound effects when inserting a sound key

Tone wheels
Turn these wheels to a similar pitch to open doors.
Interaction: Arrow keys up/down to set pitch of each tone wheel. Distance check to see if the Avatar is close enough to use it.
Accessibility features
- 3D sound as a clear tone helps to gamer find it is also this tone which is modified by turning the wheel.
- Voice feedback when the interface mode is set to "Tone wheel mode". In this mode, the arrow keys are used to set the pitch instead of moving the Avatar

Tone emitters
These are represented be different type objects scattered around in the nearby area which gives a clue to how to set the Tone Wheels.
Interaction: None
Accessibility features
- 3D sound playing a series of tones

Code locks
Code locks need to be opened by pressing a certain code based on a sound pitch.
Interaction: The numeric keyboard 0-9
Accessibility features
- 3D sound as a clear tone helps to gamer find it is also this tone which is modified by pressing the numeric keyboard
- Voice feedback when the interface mode is set to "Tone wheel mode". In this mode, the arrow keys are used to set the pitch instead of moving the Avatar
**Doors**

Most doors merely separate areas in case of emergency and are normally automatically opened when walking close to them. Some doors must be unlocked using Keyholes or Tone Wheels.

Interaction: Distance check to see if the Avatar is close enough for the door to open automatically (if unlocked).

Accessibility features

- 3D sound. Helps to gamer find it

**Bazooka**

Fires a rocket in the direction you are facing. You need ammunition for this, otherwise nothing happens.

Interaction: Space key fires the Bazooka in the direction the Avatar is facing. Picked up by walking into it.

Accessibility features

- 3D sound. Helps to gamer find it
- Sound effect and voice feedback when picked up
- Voice feedback if target is out of range

**Ammunition**

Rocket for the bazooka, contained in a box. You can carry a maximum of 10 rockets. Rocket ammunition are respawn at the same location, so you can always get back for more.

Interaction: Picked up by walking into it. It is automatically inserted into the Bazooka.

Accessibility features

- 3D sound. Helps the gamer find it
- Sound effect and voice feedback when picked up

**Laser gun**

Fires a blue laser in the direction you are facing.

Interaction: Space key fires the Laser gun in the direction the Avatar is facing. Picked up by walking into it.

Accessibility features

- 3D sound. Helps to gamer find it
- Sound effect and voice feedback when picked up
- Voice feedback if target is out of range

**Battery**

Energy pack for the laser gun (not rechargeable, you have to pick up one for each shot). You can carry a maximum of 10 batteries. Batteries are respawn at the same location, so you can always get back for more.
Interaction: Picked up by walking into it. It is automatically inserted into the Laser gun.

Accessibility features

- 3D sound. Helps the gamer find it
- Sound effect and voice feedback when picked up

**Communication devices**

Use these to get messages from professor van Lange of what has happened and what to do.

Interaction: Space to start using it. Distance check to see if Avatar is close enough. Also uses the Menu system described above.

Accessibility features

- 3D sound. Helps the gamer find it with a ring tone.
- Voice feedback when you are close enough informing about new messages.

**Robots**

Robots used for Terraforming. In case of hostile species on the planets to be colonized, they are equipped with powerful electric weapons for self-defence.

Interaction: Distance check to see if Avatar is close enough. Attacks the Avatar by walking into Avatar (using electric shocks). Warns Avatar upon sight. Follows Avatar within the current Area.

Accessibility features

- 3D sound. Helps the gamer find it
- Sound effect for electric shock

**Parts for the main computer**

These are part of the main quest. You need to bring these to professor van Lange.

Interaction: Picked by the Avatar walking into it. Inserted into keyholes with Space when Avatar is holding the key, and standing close enough to the keyhole.

Accessibility features

- 3D sound helps to gamer find it
- Voice feedback and sound effects when picking it up
- Voice feedback when the avatar tries to put a part into a computer slot which is too far away. Computer slots works in the same way as Keyholes.

**Walls**

A sliding sound is played when the Avatar slides along walls, and a bumping sound when the Avatar hits the wall straight on. For a more detailed description, see the Accessing the Physics section.
Discussion

Below I will discuss what went right and wrong.

**Balancing the game for sighted and blind**

Getting around in the 3D space is from what I have found easier for a sighted person. In this respect we have not been able to fulfill out goal fully; to create a 3D game which is equally accessible for both sighted and blind.

We have however worked on some game design level to make the game itself equally hard to finish. Since the game is not a fast paced shooter, the getting around issue is of less importance. The game features some shooting at moving robots but they are Terraformer robots, i.e made for construction, not for fighting. Hence they have only primitive defence (close range electrical shocks) and move slow.

Invisible objects such as sound keys and mines are also more easily detected by blind. Especially the sound keys; a sighted gamer doesn't primarily use the sounds as cues or interface, easily pass by the invisible sound keys. A blind gamer home in directly to examine the object. The mines are naturally detected by both, but more easily trigged by the sighted, at least the first time!

There are also sound oriented puzzles to solve to open certain doors etc. Since you can't get passed a locked door, I not sure but from what I have seen there is no difference in the detection and solving between the two groups.

**Aesthetics versus dynamics**

When we designed the game we strongly felt the importance of high sound quality overall. Hence we chose not to use speech synthesis. It was considered for the robots, which have a primitive look anyway so it could work but later skipped to avoid the development cost. Later, Macromedia implemented a Xtra (an add-on to Director) for speech synthesis but then we were almost finished with the project and avoided changing/adding features unless very motivated.

A speech synthesis would have made our work a lot easier. The slightest change in gameplay or interface caused us to get the actors on site, record new lines of speech and then edit the samples. This took a lot of time and effort, and also money.

We actually do use speech synthesis for one specific purpose, i.e outside the game in the start menu where you can check your license key. Since the license key is personal and unique, there is no reasonable other way than using speech synthesis for accessibility; i.e the speech need to be fully dynamic.

**Writing a linear story for interactive, accessible games**

I will only briefly describe some issues that we ran into of interest since Terraformers contains quite a lot of prerecorded storytelling.

As in any game, first and foremost the storytelling must be logic. If logic fails, the suspension of disbelief is shattered. Hence actions taken by the gamer must either

- be correctly reflected by the storytelling, i.e the story has to be adaptive and dynamically changed accordingly, or

- be independent from the story which is told

We chose to make the story divided into elements that are not independent themselves from the user actions, but they are told by prerecorded messages via so called Communication Devices. Since the messages are prerecorded in the game story itself, it is logic that they do not change
according to the user actions. They simply work as leads left by the professor before he was captured. This allowed a logic but simple way of telling story elements, without risking that the gamer's actions would break the logic.

Another issue is to think about orientation. Since the gamer controls the avatar, the notions of left and right, or forward and backward are of no use. Instead you have to use directions such as north, east, west, south or in-betweens.

**Naming standards**

Having a naming standard within the project is vital for many reasons. It helps you keep files and other objects organized in structured. However, we were both over ambitious regarding this resulting in a too detailed level of prefixes. Since the project development went on for several years it required a special reference for all project members to follow, which was not the case in the end. This resulted in that the naming standard became more of a confusion than a help.

**Issues with the runtime environment**

In Terraformers we used Macromedia Director and the Shockwave player for the game development and playback in the operative system MS Windows; versions 98, 98SE, 2000 and XP. We chose this tool since at the time we started the project (2001), it featured an up to date engine to an attractive price. Just to mention a few features: DirectX/OpenGL rendering, Havok physics, 3ds Max exporter, Bones/Keyframe animation, SDS/MRM support and much more. Also, we had a lot of previous experience of developing with the tool in general.

From one perspective, developing a realtime 3D game for the MS Windows platform is a nightmare. Each PC has its own unique configuration of hardware and software, making it a very heterogenous environment:

- Different versions of the operative system (i.e MS Windows)
- Different versions of hardware drivers (software that makes graphic cards and audio cards communicate with the operative system)
- Different versions of shared software libraries (i.e DLL:s) that may or may not work well for all programs using them.

(This is different from developing for a video game platform, e.g Xbox or Playstation where the configuration is homogenous with identical setups in all machines and tested thoroughly).

With the above described context it is easy to understand that when a user runs into problems with a game on the Windows platform, there are many parameters in the runtime environment that may cause the specific issue. This combined with possible problems caused by your own programming makes it hard to track down certain problems, especially when the problems are not reproducible on all machines.

To further add to this complexity some development environments are based on propriety code, i.e in which you don't have access to the source code for the development environment. When an error occurs you must first make certain your own code and hardware/software setup is OK, then find out if the error is reproducible everywhere or only on certain machines. After this, you can contact the company who have developed the development environment to get down to the bits about sorting out the issue.

To render 3D graphics in realtime, the computer needs to render about 30 image frames per second or more. Today all new desktop computers have special hardware, a GPU (Graphic Processor Unit) that takes care of the graphic rendering, using either DirectX or OpenGL. A common issue with this is that there are a variety of vendors of graphic cards using these GPU:s with their own specialized software drivers, often available in many different (more or less updated) versions.
This makes the environment in which the game is finally rendered uncertain, and crashes are not uncommon. In most cases, updating the driver software helps but in some cases you need to try other approaches. One of those are to select a different rendering method; for Terraformers there are the options of OpenGL, DirectX5 or DirectX7. As a last resort, also for those PC:s which don't have a GPU at all, you can also choose Software, which means that the regular CPU takes care of the graphic rendering.

Similar issues may accelerate in the future since also audio cards are getting increasingly advanced with sound processors. It is also quite likely that other processor intensive features, e.g physics will use hardware acceleration, to unload the regular CPU and get more dedicated processing power for these calculations.

A special problem with this when developing a 3D game accessible for blind (as I have experienced) is that in some cases it can be hard to motivate a blind person to update the graphic drivers. In most cases though the blind gamers have been very understanding.

A problem with using a propriety scripting environment is that it may be hard to get bugs fixed. First I would like to thank Macromedia for their efforts in trying to help us, but there was a certain bug that haunted us for almost a year. I thought it was related to graphic driver issues but through feedback from a gamer I finally nailed it down. The problem is solved with a workaround, since the real cause still escapes me.

**Project resources versus scope of the game**

The amount of economical resources were roughly 1% of a regular mainstream game, or about 50,000 USD in the current rate. This financial resource was financed by the Swedish Handicap Institute, and helped us getting started, roughly 2.5 months full time work for five people. The rest was invested by us with our own spare time, which far exceeded what we originally sketched in the application (the project took three years to release). The biggest issue with this was to keep the project going and focused during a very stretched time, as we worked with other projects on our day time. Basically, I worked crunch hours on almost a regular basis for three years. This problem escalated at the months before release when we had to put in crunch hours on top of the crunch hours we had been working with during the entire project.

The project started in November 2000, and the game was released in December 2003 in an english version. However, since it was financed by the Swedish Handicap Institute we also have yet to release a Swedish version, which is under production as I write this.

To sum up, the scope of the game project and the fact that we had not much experience of previous commercial game development, and even less 3D game development experience prior to this project made it an almost impossible task to finish the project. Today I'm quite pleased that we have a released title and are working on a language pack. However, it must be stressed that developing a realtime 3D game from the ground up is something that must be taken into serious consideration regarding the time and effort it takes to finalize the project. Especially if you also have to invent new accessibility features.

**Future actions for the industry**

From an accessibility point of view, games are pretty much closed products, in the sense that assistive technology can not access information embedded in the game, e.g screen readers can not read information embedded as text in a DirectX application.

Developing games today is a demanding business that takes a lot of effort to create. It goes without
saying it is not a good approach that each game developer have to invent unique, dedicated accessibility solutions (which is how we solved it in Terraformers). Rather, the primary focus must be to open up game technology to be compliant with standards already present in regular applications, with which assistive technology can communicate.

Other features do need to be built in, i.e where there are no standard feedback such as text that can be handled by some standard, assistive technology. In this case, the game industry should work together to develop ready-made libraries that can be ready to use in any game project. E.g the methods invented in Terraformers are available to purchase as ready-made library for making a 3D environment accessible. Other libraries could be a captioning system for deaf, and so on.

It is also a question of to what extent should a game be accessible, and for who. For some users, a game will never be accessible, unless you change the game itself. E.g a fast paced shooter is a game designed to challenge you physically with speed and precision (among other things). For some groups of users this type of game it is probably not possible to create accessibility without changing the game itself. My point of view is that accessibility efforts should not aim at changing the game itself, unless it improves or adds interesting and new gameplay for the majority of gamers.

Further work

IGDA GA-SIG

To address these issues of accessibility in mainstream games, the IGDA Game Accessibility Special Interest Group was founded by me and a few others in 2003. In this SIG we try to help the game industry with accessibility issues, as well as provide information and forums for discussion.

Today recent progress in the field of game accessibility for deaf is seen in high-end titles such as Doom 3 and Half-Life 2, which we find very pleasing.

Accessibility for blind is still not done in new titles but a very interesting GPL licensed project is called Agrip where a modification of the original Quake engine have been developed, complete with multiplayer support. This project may be the break through for mainstream game accessibility for blind, and can also work as a model for new game projects when considering accessibility.

You can read more about the IGDA GA-SIG at www.igda.org/accessibility

Comparative studies of sighted and sight disabled gamers playing the same game

When we started the Terraformers project in November 2000 there were no realtime 3D graphic game available that was accessible to blind, as far as we know. Today, there are except Terraformers also the Agrip project [9] and others.

Hence a very interesting study would be to have both blind and sighted play these different games in a study to compare how well the games are balanced for differently abled gamers. This kind of study is something I would like to do in later research.

Enabling game designers

New technology enables game designers to create new, innovative games. The Sony Playstation 2 have several games, like Singstar and Guitar Hero where you compete by playing and singing well. There are dance mats where you have to keep the tempo in music. The Nintendo Wii has an innovative control where you can handle a sword for instance like you would in real life.

Accessibility technology is often innovative as it has to solve specific issues. As time goes along
these technologies become mainstream. Speech synthesis, voice recognition and keyboard interaction in operative systems such as MacOS X are examples of accessibility needs that are met but can also be used by non-disabled.

If accessibility technology is embedded or supported as external add ons to mainstream games, it opens up the door for game designers to use that as game design tools and inspiration. One example is an experiment I did together with Brainfingers Inc. I created an environment with an X-Wing fighter plane from Star Wars, and setup a Havok physics simulation. Then I connected the X-Wing to a Cyberlink device (from Brainfingers) which enables you to control a computer with biofeedback signals from your brain with a head band (EEG, EMG and EOG signals). That way you can really be a Jedi, controlling physical objects with your mind directly, instead of mapping it to an abstract interface as a keyboard.

I see this as one of the most interesting approaches ahead; by implementing accessibility features in mainstream games, game designers can come up with totally new game concepts for all gamers, while at the same time make it possible to play for people with disabilities.
Literature


2. Virtual reality rehabilitation for all: Vivid GX versus Sony PlayStation II EyeToy, D Rand, R Kizony and P L Weiss, University of Haifa/Beit-Rivka Geriatric Medical Center, Petach-Tikva, ISRAEL Downloaded January 2005. ICDVRAT 2004 Conference Proceedings

3. AudioBattleShip: blind learners cognition through sound, J H Sánchez, University of Chile, CHILE. Downloaded January 2005. ICDVRAT 2004 Conference Proceedings

4. Understanding Computers and Cognition, a new foundation for design, Winograd & Flores


9. Agrip project web site http://www.agrip.org.uk/
Appendix
This list of sites is copied from the following address: http://inspiredcode.net/GameLinks.htm

Presentations of Terraformers
In reverse chronological order

2003

March 21-23:
Independent Games Festival
Game Developers Conference
San José CA, USA

2002

November 25-27:
NetLearning 2002, Sweden

October 16-18:
ID-dagarna, Sweden

July 22-25:
Vision 2002
The 7th International Conference on Low Vision. Sweden.

June 13-14:
'Computer games in school' conference
Multimedia programme,
Stockholm University, Sweden

May 9-12:
Game tournament at URIX 2002, Sweden

Week 8:
Spanish TV, 3xl.net

February 4-8:
Milia in Cannes, France

2001

October 5-7:
Computer Game Conference,
University of Texas, Dallas, USA.

October 2+3:
Game Intersections,
Stockholm, Sweden.

September 22:
TIME in Kulturhuset,
Community web sites

Audio Games dot Net
Community Portal for all accessible gaming
http://www.audiogames.net

Game Audio Network Guild
Community for all kinds of Interactive Audio
http://www.audiogang.org/index.php

Audyssey Magazine
Dude, if you don't already know what it is, then ignore the rest of my site and Go There Now!
http://www.audysseymagazine.org/

Home Page of BlindProgramming Group
http://www.blindprogramming.com

Interactive Audio Special Interest Group
(a group for audio developers nd composers)
http://www.jasig.org/

Games Room of the UK Audio Network,
the website designed for blind and partially sighted people
http://www.yrguk.com/entertainment/gamesroom.htm
Sites with Accessible Games:

SoundSupport.net
Home of Drive, Curb Game and other delights.
Online portal where three Sound Designers present
their thoughts and projects on audio and sound.
http://www.soundsupport.net/

James North's ESP Softworks
Home of Monkey Business, Alien Outback, Raceway, Pinball and more
http://www.espsoftworks.com

Justin Daubenmire's BSCGames
Accessible Games such as
Hunter, Troupanum, Pipe and more
http://www.bscgames.com

Justin Daubenmire's BlindSoftware
Accessible shareware, freeware, and demo software downloads
http://www.blindsoftware.com

The Tactile Interactive Multimedia Project:
Computer games for visually impaired children
http://inova.snv.jussieu.fr/tim

Web page of Graham Pearce
home of Super Simon, as well as a formidable collection of GWBasic Games
http://www.anycities.com/graham43

Latest News from BPC Programs
http://www.anycities.com/mun0009/programs

Daniel Zingaro Games for Blazie Note Taker
http://www.anycities.com/user/danfreeware

American Printing House for the Blind, Inc.
Termite Torpedo Game and other programs
http://www.aph.org/tech

Ark Angles: The KChess Range
Fully Accessible Chess Games
http://www.arkangles.com/kchess

Bavisoft: Software for the Blind and Visually Impaired
Home of Grizzly Gulch
http://www.bavisoft.com
J Squared accessible gaming superstore
http://www.bestmidi.com

Code Factory: Games Especially for Blind Users
Good clean fun for all ages
http://www.codfact.com/en_index.html

Joshua's Random Page
Home of the Famous Harry Potter IF game
http://www.cs.northwestern.edu/~josha/random.htm

DisQuiet Audio Games

Fantasy Storm, home of the Savage Gamut
Please don't feed the dragon...
http://www.fantasystorm.net/mainpage.html

Games for the Blind: Accessible Games
Now have self voicing games as well
http://www.gamesfortheblind.com

DreamtechInteractive:
Layered-Audio enhanced Text Adventure Games
http://www.dreamtechinteractive.com

GMA Games: Home of the famous SOD
(Shades Of Doom), Lone Wolf, Trek2000 and Tank Commander
http://www.gmagames.com

Interactive Fiction Archive
Huge Library of Interactive Fiction
http://www.ifarchive.org/indexes/if-archive.html

The Best of Interactive Fiction
A choice collection of IF Games
http://www.igs.net/~tril/if/best/index.html

InspiredCode
4BlindMice, the singing talking VI mouse driver
Metris, accessible Musical Tetris
ChurchBells, BellTower for steeple or desktop
http://InspiredCode.net

InspiredCode's Audio Adventure Engine Project
Are you the next great game author?
http://www.InspiredCode.net/Adventure.htm

Jim Kitchen's freeware accessible Games
Yes, Free! I'll be here when you get back.
LiamWorks: Home of SuperShot
and other accessible games by the infamous DJ.
Creator of SuperLiam (accessible 2D side-scroller)
(lost in a hard drive crash, hopefully not for long).
http://www.l-works.net

PCS Games: Games that tickle your ears!
Home of PacMan Talks and thirty DOS games
http://www.pcsgames.net

Michael Whapples' Moral Mike web site.
a free Windows Othello game.
http://www.btinternet.com/~moralmike

Sonokids: totally accessible games and remixers
for kids of all ages and abilities
http://www.sonokids.com/

Terraformers
Realtime 3D game with two interfaces:
One Graphic and one Audio for VI users.
http://www.terraformers.nu/

The Blind Eye: A Game for the Blind and the Seeing;
Navigate 3D-soundscapes on a ordinary PC
http://www.TheBlindEye.com

Zform: Games with Vision
Games for the blind, games for the sighted, games for everyone!
http://www.zform.com
Glossary

**Avatar**
The gamer's representation in the game world (player character). This can be everything from a non visible box or sphere in a first person only game, to a fully animated character, car, space ship etc. (Originally the meaning of the word is an incarnation of a god on Earth).

**Game Shell**
The startup menus etc which is needed to setup, start and exit the game itself.

**GPL**
GNU Public License, a common license for "open source" software

**IRL**
In real-life; an acronym used to differ between life in digital worlds and life on Earth. ("Life on earth" is actually the term I would prefer using, since the notion of what is "real" can be discussed. However, IRL is the common term so I chose to use that acronym anyway).

**NPC**
Non-Player Character, or computer controlled character. (As opposed to the Player Character, i.e the Avatar).

**Phya**
Physical audio modelling engine

**Staccato Sound System**
Physical audio modelling engine, which is discontinued as far as I know