Research and development into an innovative computer game designed to relieve stress.

Thomas Stratford (d1191296)

Abstract

Through this project we look into possible innovative gameplay concepts and how computer programs could be used as a means to reduce stress in people's lives. Presented is research into what work is currently being produced and how it achieves its aims. We discuss our own ideas for innovative game design and analyse our implementation of them into our final product: a small game where the player must speak into a microphone to encourage a virtual plant to grow. The player can also interact with the plant by moving a light source which the plant will attempt to grow towards. The methods used to produce the program and details of how it was written are outlined.

Introduction

'While stress can never be avoided, it can be controlled. To do so requires an understanding of what causes it, and a determined effort to change those patterns of thought and action.'[1]

Computer games currently span a wide range of styles and forms, from simple internet games which are easily accessible to graphically advanced and technically sophisticated games which need the latest hardware to run. As they all require interaction between the user and machine to play, there is no doubt that they must have some sort of effect on the user's mental responses. This poses the questions:

Can a game be effectively designed with the specific role of stress relief?

What innovative means are there of achieving this?

To attempt to answer these questions I have been looking into developments and innovations in game design and the potential of these as a means of relieving stress.

Computer games and stress

There are many design aspects which can be combined to make a successful game, most obviously the graphics, sound and the way the game is played i.e. the controls and the rules. Each of these components will have some impact on the user and so the way they are handled is very important. In the book Stress Care, Hans Selye, a pioneer in the study of human stress response, defines stress as,

'the non-specific response of the body to any demand made on it.'[2]

When thinking about the player and 'any demand made' on him/her, it can be argued that the majority of the most popular games currently available such as

first person shooters and driving games are very demanding on the player's responses. Games of this nature feature intensive graphics, elevated sound levels and fast paced game play with imposing deadlines such as the incentive to complete an objective within a time limit. Although games vary in intensity and realism, many popular games push all these elements to the limit which as a whole is sure to induce stress in the player even on a subconscious level. This isn't necessarily a negative factor about games as a more intense experience could make a game more engaging and exciting. However at an early stage we realised we would need to carefully consider how we used aspects such as graphics, sound effects and controls if we were to fulfil our brief of a game designed to relieve stress.

Innovations in computer games

I began the process of exploring the links between stress and computer games by researching some unique and innovative approaches to computer games design.

Although the popular market for games can often look quite stale in terms of innovation with endless franchise sequels and new releases looking like clones of existing titles, I believe there is still much innovation to be found in the world of games and interactive media.
These innovations are not necessarily available on the commercial market. A good example of this is a very original game which is being developed in medicine. The game uses 'biofeedback' technology to detect stress in the user and is designed to cure irritable bowel syndrome of which 60% of cases are thought to be stress induced. In his book Biofeedback: A New Training Technology, the author Winfield writes,

'Biofeedback is a technique for recording physiological or bodily states, and after suitably processing the information to a readily understandable form 'feeding it' back to the subject.'[3]

The technique the (currently unnamed) game uses for recording the patient's physiological state is called Galvanic Skin Response. This is found in lie detector technology and simply measures the electrical conductivity of the skin which varies with levels of arousal, tension and stress in the body. The game feeds back the information to the user through animated computer graphics and only through achieving a state of deep relaxation is the 'player' rewarded with images of open fields and running streams. The results show that there was some success in the idea as it worked on over half the patients tested, teaching them relaxation techniques they were able to use away from the game. Leahy and Colleagues who conducted the research into the biofeedback game at the Royal Free Hospital in London conclude in their analysis,

'This computer biofeedback game taught deep relaxation rapidly and effectively…Therefore, computer biofeedback games may offer a simple, inexpensive strategy for managing other stress related medical conditions.'[4]

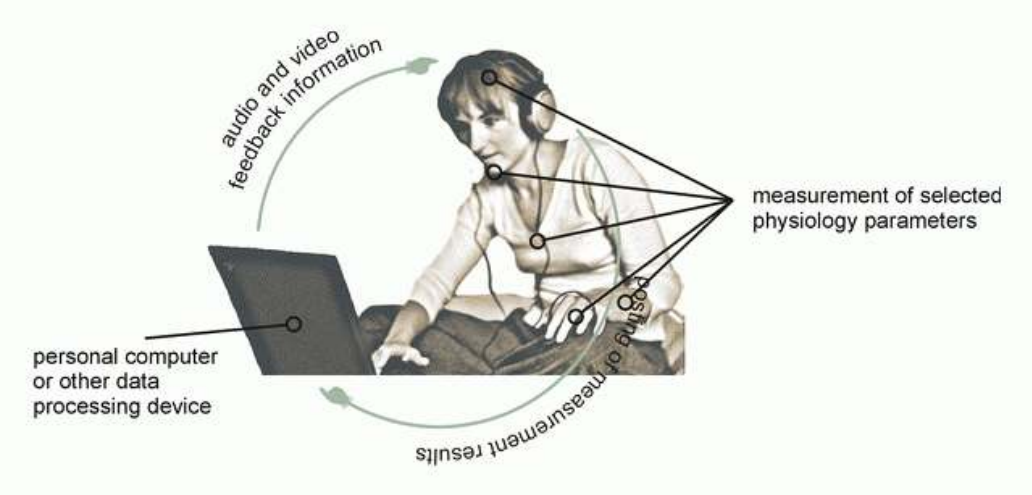
The positive results of their stress relieving game proves that games of such a nature do work and sets up much potential for development in the field. Another series of biofeedback games, inventively called Bio-Games, use biofeedback to drive more recognisable games such as a golf game where different levels of the player's muscle tension control how hard the golf ball is hit. Originally designed to train NASA astronauts in biofeedback techniques, the games are now being developed for more mainstream use. The games creator, Patrick Doyle, an associate professor of psychology at the University of Houston comments on his games,

'The computer games offer stress management in an enjoyable format so that players can learn to handle life events effectively. As players interact with Bio-Games, there are continuing challenges and new levels of difficulty to keep interest high.'[5]

The 'continuing challenges and new levels of difficulty' Patrick Doyle mentions is interesting in regards to a stress relieving game. Whilst these factors keep 'interest high' I believe there is the danger of a game becoming too challenging which could frustrate the player and unintentionally reverse the effect of relieving the stress.

[Figure 1] Screen shot's from Patrick Doyle's 'Bio-Games'. I believe the games are very primitive in appearance and functionality and don't fully exploit the potential of biofeedback.

As well as Galvanic Skin Response to measure sweating, other forms of biofeedback measure bodily processes such as heart rate, temperature, muscle tension and even brain waves. With these techniques in mind I believe biofeedback could be an interesting and potentially successful means of controlling a more commercial game than the medical related games they are currently associated with. For example if a game were to measure a player's heart rate and sweating, it could reduce the difficulty of the game if an increase in the player's stress levels was identified. Conversely if a game was designed to be exciting, an increase in the player's stress could trigger an event that notches the gameplay up another level such as increased sound and faster enemies to intensify the action experience.
I would very much liked to have explored the possibilities of biofeedback as a means of game control in our final product. Unfortunately the direction of the project took us away from this area and time and money (biofeedback equipment tends to be quite expensive!) were against us.



[Figure 2] (above) A simple diagram outlining the basics of biofeedback (from Wikipedia).

[Figure 3] (left) The game 'Wild Divine' markets itself as an 'Amazing Stress Relieving Game'. It calls it's biofeedback device the 'Light-stone' and features cheesy visuals and sound effects. At a price of £99 I don't believe it will be the game that launches biofeedback into the popular games market.

Whilst biofeedback is an unique means of controlling a computer game, perhaps the most unconventional control system I have discovered is in ‘On Target’ by designer Marcel Neundorfer. This game features a pressure-sensitive display screen located in a urinal which when ‘used’ triggers images and sound! The player’s progress is also displayed on a screen above the urinal so others can see how well he is playing. Although this approach is never likely to be taken on as a serious means of controlling a game, it is a good example of how there are still innovations to be seen in the world of interactive media which has the potential to liven up the games industry.

[Figure 4] Photos of ‘On Target’ and the unique display screens.

This doesn’t mean that the games industry void of innovation however. Perhaps the best example of a development currently making an impact on the way commercial games are being played belongs to the Nintendo console the ‘Wii’. Braking away from the traditional controller design it looks and is held more like a television remote; the real revolution in its design being that it is motion sensitive. This means that to play a racing game you need to hold the ‘Wii Remote’ on its side like a steering wheel and physically tilt it to turn the car. This is just one of a vast number of possibilities of what can be done with the control mechanism.

I believe designs such as this have a positive impact on the way games are designed and the audiences they are targeted at. Through changing the way games are played radically, some long time gamers can find themselves at the same level of ability as complete beginners which should level the playing field, increasing the fun involved and broadening the accessibility of games for everyone.

The Wii’s controller is just one of many inventive ideas for controlling games that have made it into the popular gaming market across the history of computer games. These ideas range from the earliest light gun games to products such as the ‘Eye Toy’, a camera which reads the player’s movements to control a game. Whilst some critics may throw these ideas away as gimmicks, I believe that ideas such as these are healthy for the gaming community and help drive progress and change through continually rethinking the way games should be played. In my

opinion this is a true form of innovation, as tried and tested formulas are often destroyed in order to explore new possibilities.

[Figure 5] Publicity shots of the Wii Remote in action.
(above) A screenshot from 'Wii Sports' showing how the remote can be swung like a baseball bat.
(left) A screenshot from 'WarioWare: Smooth Moves' in which the player must move the remote in a number of pre-determined was to complete mini games.

[Figure 6] A few of many examples of fun and unique methods for controlling a game. (left to right) Donkey Konga (2004), Guitar Hero (2005), Atari XG-1 Light Gun (approx. 1984)

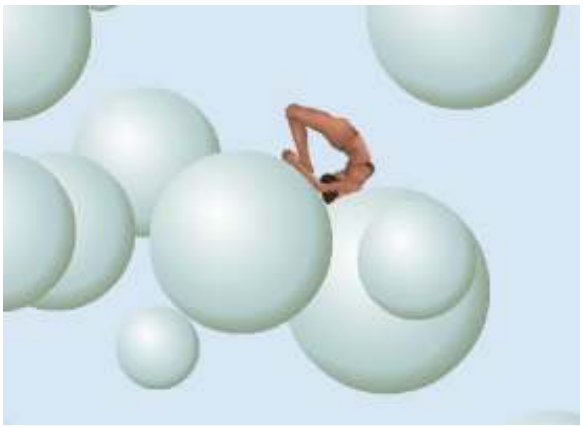
Having looked at innovative games, in particular their inventive means of control, I believe there are endless possibilities for potentially successful, fun and innovative ideas still to be realised. I learnt that an important design aspect of our stress relieving game would be to make it as accessible and fun for the player as possible. As I discovered this could require abandoning preconceived ideas of how a game should be controlled in order to explore new ideas.

Casual games

Many of the successful titles on the Wii revolve around games which are short, simple and fun. One such title simply named 'Wii Play' features several small games which each take only a few minutes to complete and are stripped to a minimum in forms of controls, sound and visuals. These games are essentially 'casual games' which have become massively popular over the last few years largely due to their ease of distribution over the internet. Windows' 'Solitaire' is a classic example of a casual game and is thought by many to be the first successful game of this kind. Casual games aren't just limited to card games however, as a huge range of genres and styles can be found on the internet. Wikipedia defines casual games as,

'…a category of electronic or computer games targeted at a mass audience. Casual games usually have a few simple rules and an engaging game design, making it easy for a new player to begin playing the game in just minutes. They require no long-term time commitment or special skills to play, and there are comparatively low production and distribution costs for the producer.'[6]

I feel the reason for the success of casual games is summed up well in this definition. As they only feature 'a few simple rules', the design of casual games must strive to be fun and engaging to keep the player entertained. Furthermore they require little commitment from the player and can be easily distributed across the internet which means they aren't limited only to players with the latest hardware or games consoles.
The fact that casual games are being sold on consoles goes to further prove the extent of their growing popularity. Not only does the Wii sell casual games bundled into packages such as 'Wii Sports' and 'Wii Play', the Xbox 360 and Playstation 3 are selling titles produced specifically for their platforms which can be downloaded from the internet. These games are low in cost and casual in nature, requiring no long term commitment from the player but still focusing on being highly playable and fun.



[Figure 7] 'Ragdoll' (left) and 'Line rider' (above) have been a hugely successful casual games. Internet distribution means they are easily available to everyone and their simple but fun concepts keeps players entertained.

Casual games were essential in our research on how to develop a stress relieving game due to their perceived links with stress relief. An online search for ‘stress relieving game’ will produce thousands of results and many online articles on how to manage stress suggest playing casual games as a means of relaxation.
One such article features the results of a survey carried out by a leading casual games developer who looked into the gaming habits of 2,100 of their clients.
I believe the most interesting statistic gathered in the survey is that 88 per cent of the participants noted that they played casual games to relieve stress and 41 per cent noted stress relief as being the most important reason to play casual games. This is strong evidence that casual games are considered by the majority that play them to be stress relieving. Furthermore it suggests that casual games contain design aspects that are key to creating a game that is successful at reducing stress. In the article, Dr. Carl Arinoldo, a psychologist and expert on stress management expresses his ideas as to why casual games are stress relieving,

“Unlike traditional videogames that tend to over-stimulate while they engage our minds, casual games have a calming effect while still providing an acceptable level of distraction and entertainment.”[7]

This further backs up my conclusions that a key to the stress relieving qualities of casual games is that they are fun and engaging enough to distract the player to a state of mental rest.

Game design

Having researched innovations in game design and the links between games and stress, we developed a much clearer idea of the aspects we would need in our game. Using casual games as a model, I have listed what elements I believe a successful stress relieving games requires,

- Innovative or interesting control system.
- Simple, abstract graphics.
- Ambient sound effects.
- Pick up and play functionality.

I believe a large factor of the success of casual games is the fact that they can be picked up almost instantly by a player and forgotten about just as quickly. This helps relieve stress as the player is able to forget about anything that is troubling them and focus on the game even if it is just for a few minutes at a time.

- Simple game-play.

One important aspect of a stress relieving game is ensuring it's easy to win or if there is no target at all which should remove negativity toward the game and make the player feel positive. This is possibly the hardest factor to get right in a game as is summed up on 'deepfun.com',

'When the challenge is greater than our abilities, we become anxious and potentially dead. When the challenge is significantly less than that of which we are worthy, we become bored, and potentially dead.'[8]

This is a simple summary of psychologist Dr. Mihaly Csikszentmihalyi's 'flow theory' which basically relates the amount of fun somebody is having to that person's ability to complete a task. There is a great deal of documentation on this subject which is becoming popular as a model for game design.

Product development

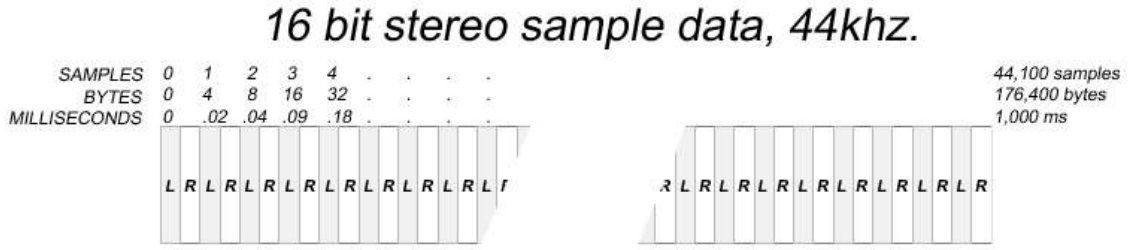
Our final product is an interactive plant growing game in which the player speaks into a microphone to encourage a virtual plant to grow. The game analyses the player's sound input and varies the way the plant grows depending on the signal's volume and pitch. This sound input could be anything from a song played into the microphone to the player shouting to let off some steam. The idea of this is to provide an innovative control mechanism which will hopefully relax the player as he creates the plant. As well as speaking into a microphone to make the plant grow, the player can control a light on the surface of a dome surrounding the plant. Using the idea of phototropism (the growth of a plant towards a light source) the virtual plant will lean and grow towards the user controlled light. This gives users freedom to make interesting and unique shapes from the plants they have created.

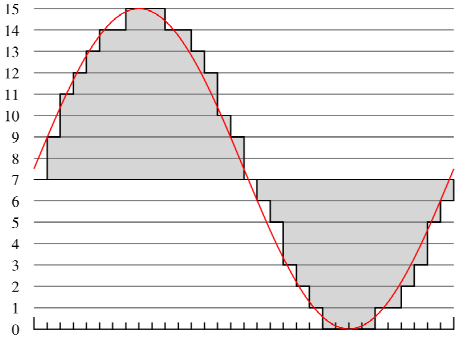
From our initial idea of creating a stress relieving game and drawing upon our combined research, we began the task of developing our final product.
After some brain storming we decided upon the idea of interactive plant growth. This would provide us with the stress relieving qualities of nature and give the user something to manipulate and play with without setting a specific challenge. As an incentive to make the game controls innovative we decided to make sound an important aspect of making the plant grow, using a microphone as a means of getting sounds into the program.
With the idea on paper we split the project into two areas of development with Patrick using his experience with L-systems to create a system for the actual plant growth and me focusing on creating the sound input for the game.

After looking into some available audio libraries I was advised to use 'fmod', a widely used and very powerful music and sound effects system which supports a variety of platforms. I was advised to use fmod for reasons of its ability to efficiently stream sounds which would allow me to sample sounds from the microphone and quickly and seamlessly analyse the sound data for use in the game.
Initially I found the process of using fmod quite daunting. I don't consider myself to be the most adapt programmer and I found some of the technical aspects of how fmod handles sound difficult to grasp. As I spent more time focusing on fmod and how it worked however, I was able to develop some chunks of code which performed simple tasks such as playing an mp3 and eventually displaying the wave data of music being played in a window.

From here I needed to use the knowledge I had gained to produce the code that would feature in the final program. My first step was integrating microphone input into my simple fmod setup. To work out how to do this I needed to understand how fmod actually handles sounds.
I discovered fmod uses PCM (Pulse Code Modulation) samples which is the standard for digital audio in computers. It is the digital representation of an

analog signal where the magnitude of the signal is sampled regularly at uniform intervals and converted into digital code. The below diagram from the fmod documentation shows how raw PCM sample data is stored in the fmod buffers:





[Figure 8] (above) Details of how fmod uses PCM to sample audio data.

(left) A simple graph outlining how an audio signal is encoded using PCM (from Wikipedia).

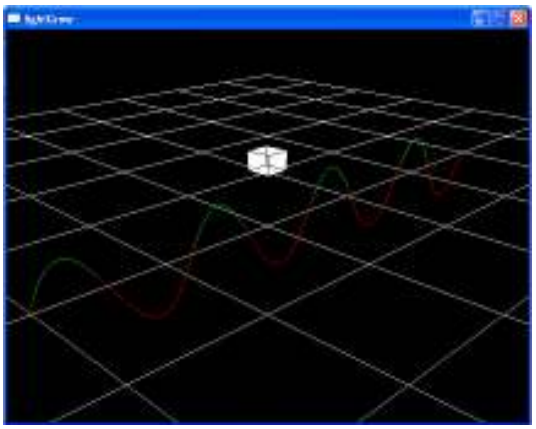
As can be seen from the diagram a pair of left and right sounds make 1 sample which is made up of 16bit data and therefore 1 sample is equal to 4 bytes. As the default playback rate in fmod is 44.1khz, fmod is by default handling 44100 samples per second.
Early lack of understanding on how fmod uses PCM to sample audio led to my first problems in creating microphone input. Although I was able to set up a sound which could be recorded by fmod and played back through the speakers, I found there was roughly a couple of seconds delay between speaking into the microphone and fmod playing the sound back. This was a problem as although the delay was a relatively short amount of time it was very noticeable and would have made the game seem slow and unresponsive if it had featured in the final version.
I found the cause of this problem was that when I was creating the sound to be recorded to I was using the default settings. The sound I was creating didn't yet have a specified length as it contained no data and therefore I needed to specify how much data I wanted fmod to record to.
The solution to the problem was to use the fmod structure FMOD_CREATESOUNDEXINFO which allowed me to define the length of the sound I desired. This was a simple case of reducing the size of the sound to 11025 bytes. As can be seen from the PCM diagram 176,400 bytes stores one second of sound at a playback rate of 44100 samples per second and as 11025 is sixteen times smaller than this, I specified the length of the sound to 1/16th of a

second. This is small enough to give the impression that what is spoken into the microphone is fed back through the program instantly.

The process of creating efficient microphone input took a significant amount of time and I had to fix multiple small errors in the code before I had a working model. The final steps of how my code handles the microphone input are listed below:

1. Create fmod system.
2. Initialise the system object.
3. Set up the structure FMOD_CREATESOUNDEXINFO to provide extended information to fmod about the soon to be created sound.
   - Format of the sound. (I set this to 16bit integer PCM data).
   - Default frequency of the sound. (44100Hz)
   - Size in bytes of sound to create. (11025)
4. Create sound for streaming using the specified fmod exinfo. Set the sound to loop so it will repeat until fmod shuts down.
5. Start the recording engine recording to the created sound. Tell the recording engine to start recording from the start once it has reached the end. This means the sound is continually being overwritten.
6. Whilst the sound is being recorded to, immediately play the output through a specified channel. Although I don't want the sound to be heard by the user, the sound must be playing within the fmod mixing engine in order for fmod to analyse the sound's output signal.
7. Set the volume of the channel to 0. This means that the player will not hear the recorded sound being played back through the speakers. The fmod engine analyses the sound's signal before reducing it's volume so no information is lost.

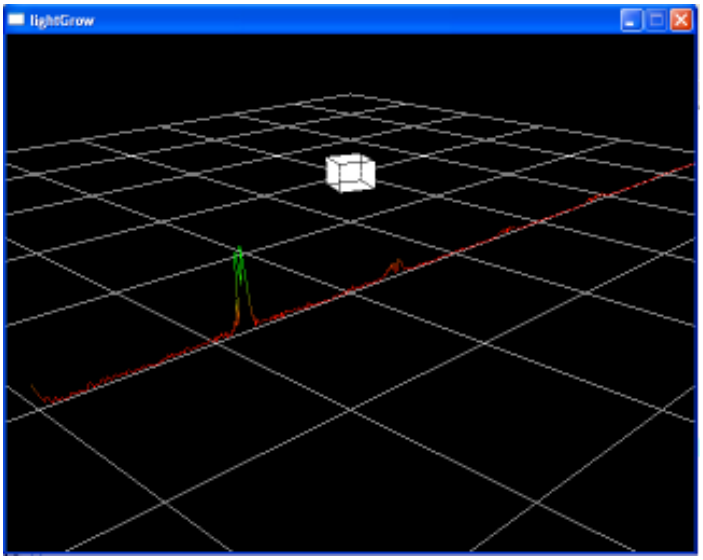
Now that I had sound input to the program from a microphone, my next challenge was to use the sound's signal to retrieve data that could be used in the final application. The two aspects of the signal I focused on analysing were the sound's volume and frequency.
To calculate the volume of the sound I used the fmod function Channel::getWaveData. To retrieve the volume data from this function an array is created which gets passed into the function. With the array holding the data of the sound's wave the program performs a simple comparison operation in order to retrieve the maximum value in the array which corresponds to the maximum volume of the sound.
In early tests of the code a cube was set to respond to the maximum volume of the sounds being input by the microphone. The max volume of the sound was set to the cube's Y translation which would make the cube raise higher the louder the sound. Although this worked the cube tended to move quite erratically as a sudden loud sound would make it briefly get drawn at the top of the screen before appearing stopped at the bottom again with no movement in between.
We decided to make the growth of the plant smoother by calculating the average maximum volume of the sound across a period of half a second. The old volume average is then linearly interpolated over the course of the following half second to equal the new volume average. What was the new average now becomes the old average, a new max volume average is calculated and the process beings again. This makes the growth of the plant very smooth and as it samples the average volumes over half a second of input a brief loud noise doesn't make as dramatic an impact.



[Figure 9] The wave of the sound is drawn here show the varying values in the array. This wave displays a sung note and the wave of a person's voice speaking is not usually this clean!
The maximum values from this array are averaged out every half second to get the average volume.

Calculating the frequency of the sound works in a similar way but uses Channel::getSpectrum to get data from the sound. The spectrum is fed into an array which is stepped through and the size and location of the prominent frequency is stored. The size of each entry in the array is then calculated (using the macro BINSIZE in the code) and multiplied by the value of the location of the dominant frequency in the spectrum array. I had troubles calculating the frequency myself but using the fmod example code 'pitch detection' I was able to get a clearer understanding of how the sound's frequency could be calculated. Whilst the final values calculated using this method aren't precisely the frequency

of the sound, they are a good enough estimate to get usable results from. Finally the average frequency is also sampled over a period of half a second and the values are linearly interpolated between in the same way as the volume data.



[Figure 10] The spectrum of the sound is drawn here to show how the values in the array are stored.
The dominant frequency can be clearly seen. The value obtained from the array and calculated appropriately.

Once the code was completed I moved onto the second form of controls for the game; controlling the light. I got a copy of Patrick's growing plant program and stripped the code down to just what was necessary to add controls to the lights. Working in this way was helpful as it gave me an understanding of how Patrick's half of the program was working and also meant that my code would integrate back into the main program seamlessly once complete.
Creating controls for the light was a case of learning the SDL_PollEvent loop to check for any key presses. In my first implementation of the light controls the code would check for a key press and add or subtract a unit to the appropriate position vector of the light. Whilst this worked in moving the light around it wasn't very useable as the key needed to pressed every time the user wanted to move the light one unit. I discovered the answer to this problem was that only one event occurs when a key is pressed down or released in SDL. This meant if the key was held down the SDL_PollEvent loop would see the corresponding SDL_KEYDOWN event had already occurred and would continue through the loop without altering the light's position more than once.
The solution to this problem was simply adding a SDL_KEYUP event and a 'bool' variable corresponding to each key. When the key is pressed the bool relating to that key is set to true and when the key is released it is set to false. The final solution to the problem was to update the position of the light outside of the event loop. This means that a simple function call changes the position of the light accordingly when a direction key's bool is set to true and stops changing the light position when it is set to false.
As a final touch to the controls I added acceleration and deceleration to ease the light into and out of movement and used SDL's time functions to make the movement frame rate independent.

Critical analysis and conclusion

In our project proposal we stated we wanted to produce,

‘a small but sophisticated stress relieving game the player can pick up any time and enjoy.’

Looking back at our work I believe we have achieved our criteria and am pleased with the result. However, I also believe there is still much room for development. Whilst I personally believe the game is somewhat relaxing to play, it lacks enough incentive to keep the player interested for a very long time. As a game for a quick stress relieving break I would personally like to be kept occupied for five to ten minutes whereas I feel interest in the game is hard to keep for much longer than two or three.
An obvious way to solve this problem would be to include more of an objective to the game. However this then leads to the danger of the game becoming more competitive to play and could infringe on it’s stress relieving qualities.
The way around this problem could be instead of adding an ‘objective’ to the game, giving the user more of an ‘incentive’ to play the game. The incentive could be to grow the plant to produce relaxing music and create nice visuals which could be saved out as pictures, audio tracks or even video files. This could add an extra dimension to the player’s experience as a sense of achievement could be gained out of having made something. Another interesting addition would be more customisation over the colour of the light, the colour of the plant, the types of sounds being played along to the plant’s growth or more complex ideas such as being able to control the weather in the plant’s world to affect its growth.
As well as gameplay issues the game is still a prototype of a larger idea and therefore hasn’t been pushed very far with graphics and sound design. These elements could be improved upon to add an extra layer of gloss to the game and make it feel more complete.

As I am not a confident programmer, I found the process of creating the final product very challenging. This was a positive challenge as it meant that I had to think about problems I faced and how they could be solved rather than jumping into problems with no solution which would have produced poor results.

Having looked into innovations in computer games and what is currently available I don’t believe our product is truly innovative but is quite individual and fits in nicely with the shape of computer games to come. Although features such as microphone input to drive a game have been used before now, they have mostly revolved around repeating pre-determined orders into the microphone to drive an event in the game. I feel our game is quite unique in the sense that the user is able to decide what sounds they would like to use to make the plant grow. Personally I find the game is most fun if I put the microphone next to a speaker

and play music through it. This makes the plant move in an entirely different way compared to voice input and the music makes the process feel more relaxing.
As the numbers of people that play games increases and the market broadens, products of a similar nature to ours have been growing in popularity. The rise in casual gaming is an indicator of this. There is definitely a future market for these fun-centred, less intense games which I believe can only add more positive variety and adversity to the computer games market.
I feel that innovation is important and my research has assured me that within the computer games industry (as with most other industries associated with computer animation and visualisation) there is always innovation taking place which keeps the industry fresh and exciting to work in.

Notes:

As part of my hand-in I have included two demos of my individual work using fmod.

The first demonstrates my microphone input to give a clear understanding of how it works. In the demo the larger the volume of the sound input the higher the cube will rise and the higher the pitch of the sound (i.e. the greater the frequency) the faster the cube will rotate.

The second demonstration is my research into the potential of fmod's 3D sound and virtual voices. In the program the red disc represents the 'listener' and the green dots each represent a sound in 3D space. Each sound is assigned one of seventeen audio files at random which it plays on repeat. There are 50 sounds in the scene playing simultaneously but only the closest five to the 'listener' are audible and the rest are 'virtual'. This is a feature of fmod that means there can be many more sound effects playing in a game then there are available channels on the player's soundcard.
The concept behind the demo is that the seventeen audio clips which are assigned to the sounds at random are all notes of the C major chord (except for one slow drum loop). This means that as the user moves the listener around the space no matter what five sounds the listener happens to be closest to, the combination of notes will always create an ambient harmony.
Unfortunately I didn't have time to integrate this into the final product but I believe it would have really added to the stress relieving qualities of the game. It could be implemented into the final product in many ways. One idea I have had is that the 'listener' could be attached to the tip of the plant. This would mean that as the plant moves to follow the light it would travel through the sounds creating different musical textures as it went.

Although I created the source code on my own, it could not have been completed without the help of James 'Jimmy' Whitworth the demonstrator, Rob Bateman's online code tutorials, the fmod documentation and the fmod code examples which were included in the downloaded fmod package.

References

[1] Evelyn DeLunas from the book ‘Stress Care’ edited by Li McDerment. Social Care Association.

[2] Evelyn DeLunas quoting Hans Selye in ‘Stress Care’ edited by Li McDerment. Social Care Association.

[3] Li McDerment quoting Winfield in ‘Stress Care’. Social Care Association.

[4] Leahy A et al. Computerised biofeedback games: a new method for teaching stress management and its use in irritable bowel syndrome.
Article links -
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=9881312&dopt=Citation
and -
http://www.positivehealth.com/research-view.php?researchid=1707

[5] From http://www.sti.nasa.gov/tto/spinoff1997/ch1.html - Dr. Patrick Doyle.

[6] Wikipedia, 2007 (online) - http://en.wikipedia.org/wiki/Casual_game

[7] From http://biz.gamedaily.com/industry/feature/?id=13799 – James Brightman quoting Dr. Carl Arinoldo.

[8] From http://www.deepfun.com/funflow.htm - Bernie DeKoven on the theories of Dr. Mihaly Csikszentmihalyi.

Final quote from our Innovations project proposal, Thomas Stratford and Patrick Anderson.

Title page image – Managing Stress – The practical guide to using stress positively by Ursula Markham. Illustration by Peter Till.

Websites

http://news.bbc.co.uk/1/hi/health/227970.stm - Article on biofeedback games.

http://www.yankodesign.com/product_info.php?products_id=866 – Description of game ‘On Target’.

http://wii.com/ - Official website of the Nintendo Wii.

http://www.meditations-uk.com/landing/l_stress_game.html?kw=wild%20divine&fl=2132 – The 'Wild Divine' website.

http://en.wikipedia.org/wiki/Pulse-code_modulation - Detailed description of PCM.

http://en.wikipedia.org/wiki/Biofeedback - Detailed description of biofeedback.

http://pekkasandborg.com/portfolio/?id=2 – Location of the 'Rag doll' game.

http://www.deviantart.com/deviation/40255643/ - Location of the 'Line rider' game.

http://ishi.blog2.fc2.com/blog-entry-85.html - Not referenced in the report but is included as the games featured helped inspire the project idea.

Additional reading

Open GL Programming Guide (the 'red book') – Addison Wesley 2005.
Fmod documentation – available from http://www.fmod.org/