EE3K1 Interactive 3D Design for Virtual Environments and Serious Games

Written Report

18/11/12

Yousef Amar (1095307)

Contents

Abstract

Within this document, the EE3K1 oil spill project is reported on, its adaptability to be ported to a multi-touch form factor explored, and the human factors issues involved discussed. This is done with the aid of research and experience and ultimately a mock-up concept design is outlined. The oil spill project was the design of a simple game in which a player has to save animal life from an oil spill that aims to educate and disconcert the player. This game concept can be projected to work on a multi-touch device, such as a tablet PC, itself and the design and appropriate human factors are discussed.

Introduction

In the wake of the smart phone revolution, it is clear that multi-touch interaction has become deeply ingrained into our collective psyche. Through interfaces such as the Metro UI and products such as the iPhone/iPad, and all of their substitutes, scaling images with two fingers or turning a page with a swipe has become so intuitive that it is almost naturally expected. It has reached a point where even our toddlers try to interact with magazines through tapping the images within[1].

It only makes sense that multi-touch interfaces should be harnessed to educate and entertain. As such, when a system, for instance the oil spill project, is to be adapted to a multi-touch form factor, the most important aspects to consider are the human factors involved when implementing the system. Not only are there fundamental features to be reflected upon dealing with multi-touch interfacing in general, but human factors issues are much more divided when it comes to platform specific implementations on smart phones, tablet PCs, or a surface table for instance.

This report will look specifically at the system and how it can be applied aim to a tablet form factor specifically. The reasoning behind this is multifaceted. Primarily, the design fundamentally aims to educate as many people as possible and tablet systems, such as the iPad, are currently existing, accessible, mass produced consumer devices that are not conceptual, complicated, or bleeding edge. Additionally, the style of the game is not uncommon (a first person view, casual game) on handheld devices while surface tablets would be more suitable for a top-down view, strategic game or simulation, such as simulating the spread of oil over time. One could apply the same arguments for smartphones but due to the trade-off between abundance and processing power as well as human factors regarding such a device that are discussed in detail later, a tablet PC is decided upon.

[1] "A magazine is an iPad that does not work" - http://youtu.be/aXV-yaFmQNk

Project Achievements

(See appendix for any and all screenshots)

The oil spill project in BGE is a casual yet addictive single-player game in which a player attempts to rescue as many animals, which are being affected by an oil spill, as possible in a limited amount of time. Through the mechanics and visuals, the game aims to evoke long-lasting emotion by presenting the player with dramatic events, build a sense of urgency through the countdown timer tied in with the game's goal, bring forth feelings of despair in the player as some creatures inevitably do not survive, and associate all these experiences with the core goal of the game which is to educate the player on the effects and repercussions of large tanker oil spills and drive the player to sympathise with people and creatures affected by oil spills ultimately making them more environmentally friendly.

The implementation and interfacing on a normal PC is simple; the player moves using the standard WASD key mappings and mouse movement to change their pitch and yaw. These controls usually need no prior explaining to a new player. The player uses the left mouse button to save a creature and the right mouse button to leave and enter their boat. The player is introduced to these controls in-game as they become relevant.

As far as human factors go, there are not many issues on a PC platform as it has been tried and tested again and again. When there are issues, they are usually UI problems or in-game inconsistencies. This game however, barely has a UI as the HUD is just a score counter for how many animals you have saved and a countdown timer to hopefully keep the player engaged.

At the time the work plan was written, the player could be completely controlled, the scene included an animated water plane with normal mapping for lighting, a low-poly oil tanker, a skydome with a hemi light source, and some basic smoke particle effects. Since then blender was abandoned for Unity. The same progress was made in a matter of minutes and looked even better. The argument can be made that using engines such as Unity or UDK allows easy porting to iOS devices as Blender Game Engine only supports Android.

A number of animals have been added, namely fish and seagulls, with simple animation and AI. For example, the seagulls will fly straight in a random direction that incrementally rotates randomly but slowly with the direction more likely to point downwards. The only thing they can perceive of the environment as agents is their height relative to the water level which is fixed. If they fly below the water level, they switch to the drowning state which has its own animation and movement behaviour. The seagull's texture is also changed to being covered in oil slick. Once the player is near enough to save the animal, it moves into the boat. This is purely reflexive but the behaviour could be expanded upon in the future.

Additionally, after the player is done, they can view statistics or light information panels pertaining to the animals they saved. On a PC, this information can potentially be displayed on mouse-over, while a tablet could have a scrolling list on the side or icons.

The geometry is kept minimal for performance reasons though visually, it would look better on a tablet device. Furthermore, lighting in modern PC game engines often very high quality and with more and more shaders replacing fixed pipeline functions, algorithms such as deferred shading (where the entire scene is drawn without shadows then shadows are overlaid as each light source is computed) allow for an infinite number of light sources and great graphics. Shadow mapping and fancy lighting is however not essential for the goals of the application to be met. Although they do help with perceptions of depth, they can be replaced by "blob" shadows (a simple dark circle with feathered edges underneath an object that may or may not change in size depending on how high they are). It is more important that the animals look realistic to meet the goals of the game.

Realism is accomplished through intelligent use of textures and materials. For instance, the edited texture of an oiled bird may not be enough to show that it has been oiled. As normal mapping it would be slightly overkill, it is sufficient to darken the materials and increase specular lighting to give it a wetter and waxier feel. This would also mean that less computational resources would be needed and would make it more feasible to work efficiently on any mobile device.

A few point light sources have been added at the centre of particle effects with a reddish hue to convey the atmosphere. In conjunction with this, the water, which as aforementioned is a simple plane with animated textures and normal mapping, has its own texture for reflection and does not actually reflect objects above it. This behaviour however, is impossible to tell as the illusion of tiny waves and ripples makes it seem like all light reflected is scattered and only the aliasing in the distance starts to reveal that it is not through the rendering of a slight Moiré pattern (the optical illusion of interference patterns).

The aliasing could be fixed through mipmapping the water textures (using pre-scaled down versions of the textures as they are rendered further away) such that the texels seem to blur in the distance, however this is not done due to the positive side effect that makes the water with the aliasing effect look like it is reflecting the sky.

Interaction Possibilities and Human Factors Issues

One might think that the game objectives are independent of the platform and interfacing method but that could not be further from the reality. The interface alone contributes a great deal to the player's immersion and even the fidelity of the design. For example, a simulation designed to train surgeons for minimally invasive surgical skills may have no realistic graphics and textures at all but accomplishes its goal perfectly through the mechanics and the human interface, which is very similar to the surgical equipment (Stone, 2012).

Beyond the standard keyboard and mouse interface, or an intuitive Xbox controller, there are a number of ways in which the player can interface with the game. Given the nature of the game and the fact that it is in a first person view, the first way that often immediately comes to mind is a head mounted display. This can however be almost immediately disregarded as it would directly clash with the objectives. HMDs have yet to progress to a point where they lose their air of novelty and the technology reaches a point where the wearer can completely forget what they are wearing and be completely "immersed". This can already be achieved by a simple screen; a well composited movie can lock people in a solid trance without the need for any sort of augmentation.

There have been products however that did look very promising that do not involve a lot of technical steps to match the system but still do increase immersion. An example of this is the "jDome"[2], a hemispherical canvas on which the game can be projected providing a 180° field of view that works off the shelf in a variety of areas.

The jDome being used for simulation and training; Source: jdome.com/simulation

[2] Official jDome website - http://jdome.com/

The game would still be interfaced by keyboard and mouse but would possibly feel much more immersive. We all have experienced the automatic storage of certain actions into muscle memory, whether they are game controls or driving a car, to a point where interaction does not become an issue anymore as our minds have already mapped it out. The real issue is *learning* to use a new interface and, as WASD + mouse is so well known, that issue would be solved. Ideally such an interaction method would be used as it is comparatively cheap and simple. Of course, realistically, to have as many people play as possible, a tablet device is a much better platform as, due to the casual nature of the game, it would appeal to the users of such devices.

Future Implementation Using a Multi-Touch System

The most important aspect of adapting a game to a multi-touch system is to not detriment the fidelity. It can be anything from delivering training to entertainment as long as the implementation does not hinder them.

As far as performance is concerned, if the game was designed properly and optimised, it should not have to be streamlined even more to work smoothly on a tablet device. In this case there is very little geometry, unlike a simulation where relatively high detail is a must and a large number of objects exist such as a submarine true to its real-life counterpart.

Several techniques such as normal mapping and drop shadows were already mentioned however more can be done to improve performance whilst not affecting the goal and quality of experience. Firstly, there are standard graphics optimisation techniques that engines often have built in. For example, with a land/seascape environment like this, level of detail can be changed depending on how far an object is from the camera. Objects could either be replaced with versions of themselves with a lower polygon count or culled out completely beyond a certain distance smoothly with the use of fog. This would reduce the number of polygons that need to be rendered and can be incredibly useful for weaker platforms. It would be up to the designer to set the fog distance depending on the requirements for example.

Other techniques are usually already handles. These include backface culling to only render front faces, frustum culling to throw out anything outside the frustum, the use of display lists and vertex buffer objects to cache vertex data on the graphics instead of sending it from CPU to GPU every time among many others. What can be done by the designer though, is for example pre-baking shadows on static objects, improve lighting with techniques such as ambient occlusion, or using just ambient and diffuse lighting. This is not needed for this particular implementation of the oil spill game however.

Textures may have to be scaled down, but this would be to a degree where it is still unnoticeable since the resolution on tablet devices is inherently less.

One of the disadvantages of multi-touch interfaces is the complexity involved. This manifests itself in the designing of the system as well as the user's interaction. The user had many more possibilities and potential actions. The user can use more than one fingers or gesture over time all on the same screen which requires unique classifiers. Fortunately most devices have built in APIs for detecting gestures and in some cases (Android) game engine map touch events to mouse events 1:1. For instance tapping would be interpreted as a mouse press and swiping as a mouse drag.

Furthermore, the fact that the input device is directly on top of the display means that controls would have to be placed very carefully as to not obstruct the viewport. This, coupled with the limited size, is another disadvantage. For this reason a tablet device is preferred to a smartphone for this kind of implementation as the extra space would be needed to display text and a scene with enough detail. This has already been proven possible with e-reader apps and the like.

Another large con is the strain and fatigue people experience after sustained interaction. Either one holds the tablet device at eye-level and maintains a correct but unnatural posture, or holds it like a book slouches. The size of the screen might also cause eye strain and the involvement of gestures that require the user to remove their hands off of the sides completely and move their arms near the centre of the screen can become quite tiring. The same goes for any tilting or movement actions. An ergonomic holding position would be if both hands are on the sides with the thumb pointing upwards and if most controlling is done with the thumbs.

Furthermore it is not uncommon that taps are misinterpreted especially if buttons and other features are close together or the user has generous fingers. This issue is not as prevalent on tablets as it is on smartphones but must still be considered when designing GUIs; buttons and text must be made large and separated. Fortunately for the reasons discussed earlier and how they match the goals, the pros outweigh the cons when it comes to adapting the system to tablet device.

The WASD keys and mouse-look controls can likewise be intelligently translated. One might think you external controls work best, such as the interface on the Xperia Play for example but on-screen implementations can comfortable to the user.

Xperia Play – Source: Gameloft

An example of a successful interface is Minecraft Pocket Edition, based on the popular PC title Minecraft adapted for a myriad of handheld devices. As it is similar to the oil spill project in its controls, the multi-touch interface can be just as achievable.

Minecraft Pocket Edition for the iPad – Source: GameBlaster64

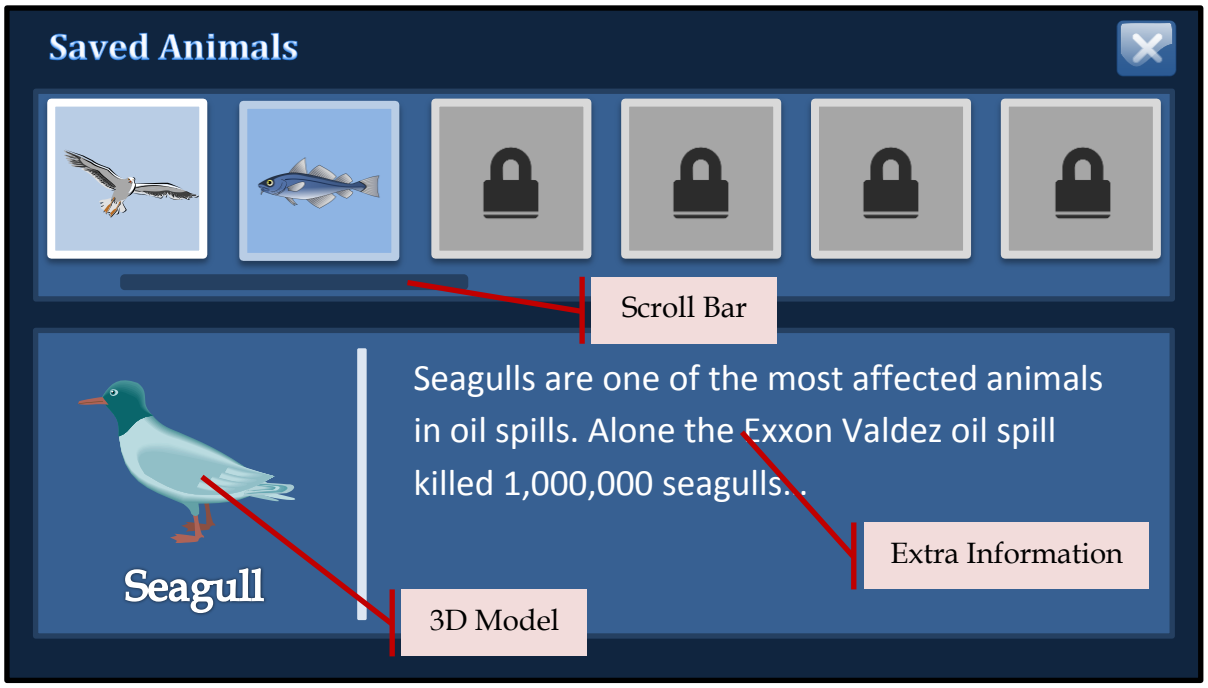
The movement buttons, ideally joystick-like, on the bottom left corner would match the player's hand position as well as allow for movement in any direction and tapping for jumping while at the same time being semi-transparent (perhaps even completely disappearing if the player lets go for a while) to not obscure the view. At the same time, dragging anywhere else on the screen, though probably on the lower right, alters the players pitch and yaw changing the view. Actual physical joysticks could even be attached to the screen if the player preferred it.

A further possibility is using the well-known two-finger pinching gestures to zoom in and out when reading the full texts on the affected wildlife and dragging horizontally and vertically to rotate the models along their x and y axes when being viewed. Another potential interfacing medium is tilt and movement utilising the accelerometers and gyroscope within most tablets. For the purpose of this design, these are not used as they do not contribute to the goal and perhaps even make the user misinterpret them.

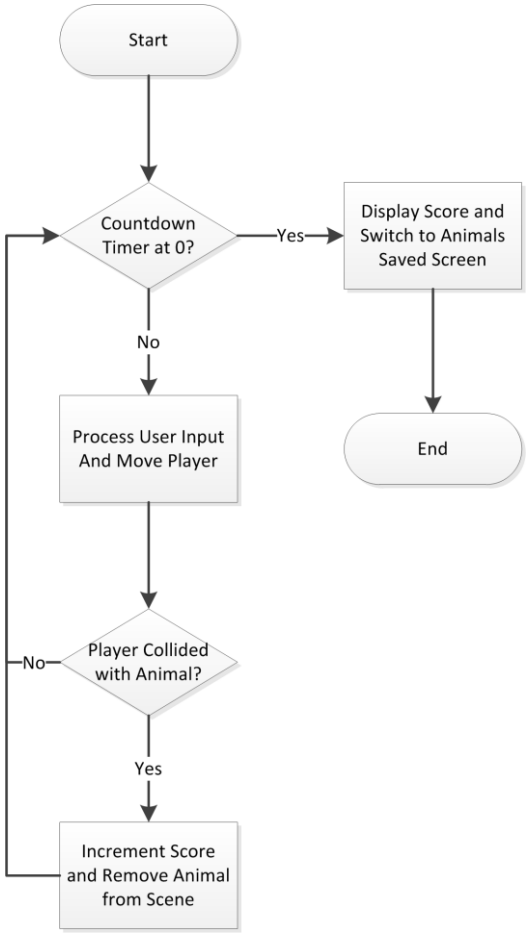
Concept Design

Based on all the concepts and considerations discussed previously, a multi-touch tablet adaptation can designed. The player would start with an almost identical menu to the PC version. The difference is that the buttons and text contained within them is much bigger. Then the player begins the game and proceeds to bring their hands to the sides of the tablet. They would use the two joystick buttons to look around and move and tapping on the left button or right button once to convey the equivalent of mouse clicks on the PC version.

The player would then proceed to play the game normally as if it were the PC version with only minor difference to account for performance as discussed in depth in previous sections. UI differences would only start to arise once the main game is over and the player can see what animals they have saved and read about them in more detail. This is shown as buttons with image icons within them that display more information when tapped. The buttons are aligned in a single row on the top of the screen and can be scrolled from left to right with a swiping motion, that is particularly intuitive to tablet users, or with a scroll bar underneath. It would look something like the following.



The gameplay itself may also be reduced slightly in complexity in that the player can save an animal simply by colliding with it to avoid having to make the user have to switch between using their thumb for looking and for saving. Other than that the same basic design is used that can be represented with a basic, non-UML flowchart.



Once the player taps the "X" button on the "Animals Saved" screen, they return to the menu screen where the highscore value can be seen as updated if they beat it.

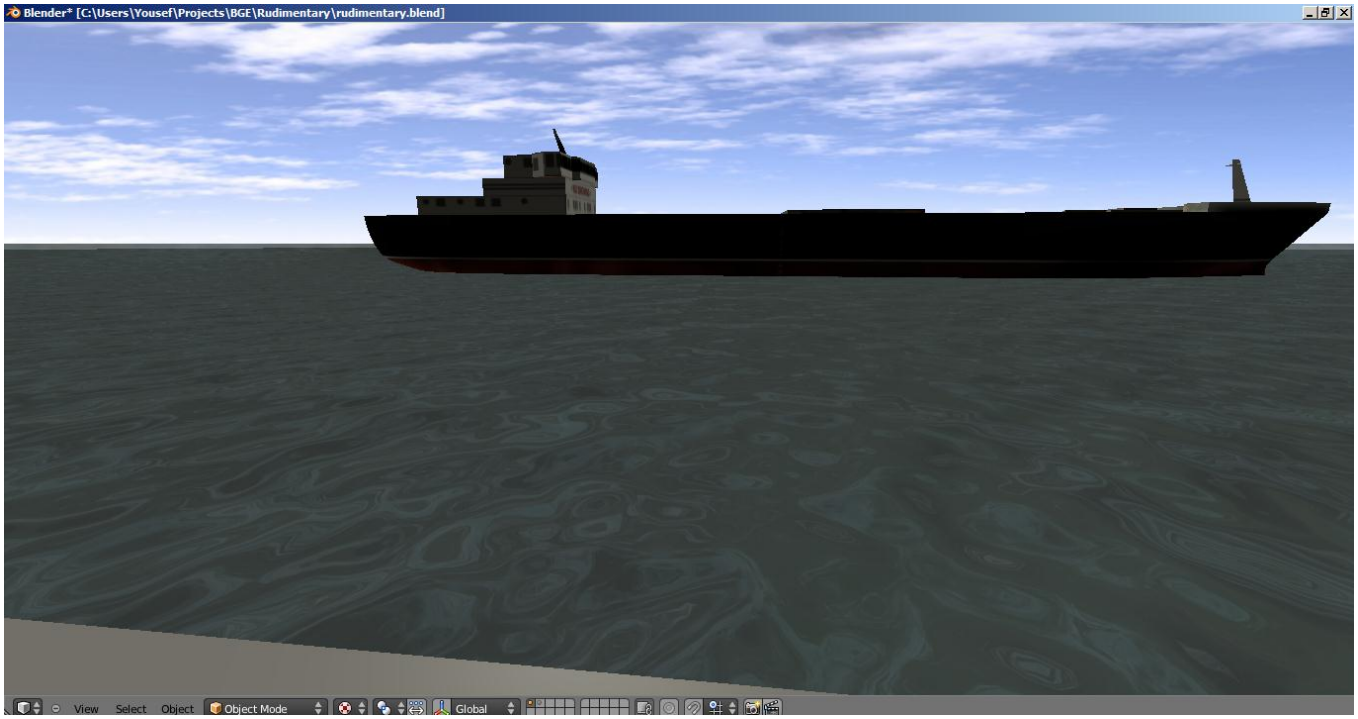
Conclusion

Ultimately, if all went to plan, the player should emerge more educated and sympathetic towards the consequences of oil spills. The iPad can then immediately be passed on to somebody else to have their two minutes of trying to beat the highscore and bettering themselves. Although a tablet-specific multi-touch concept was designed and significant human factors discussed, one can only speculate what the future holds in the world of handheld devices.

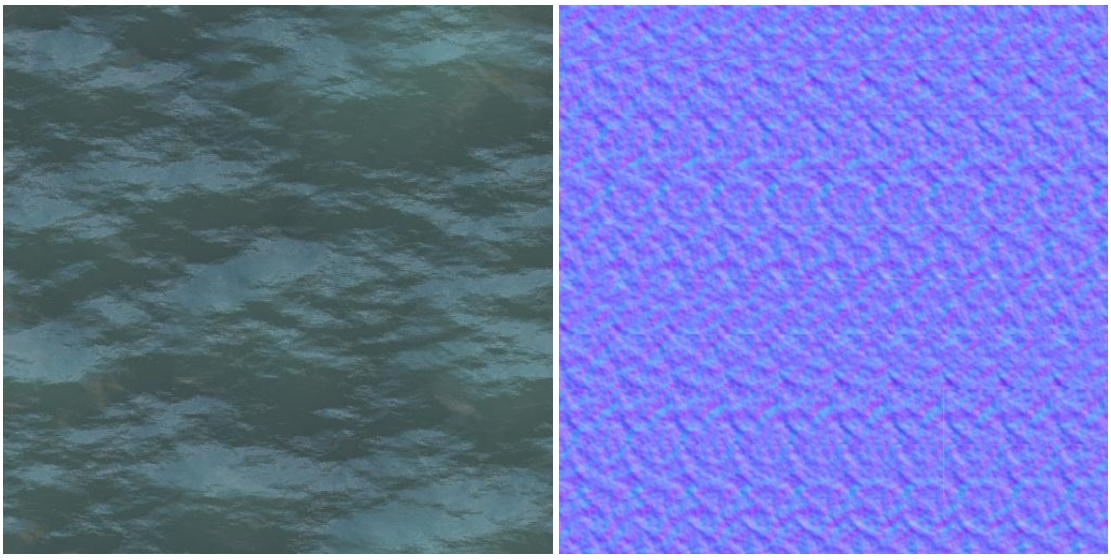
Bibliography

Stone, R. (2012). *Human Factors Guidance for Designers of Interactive 3D and Games-Based Training Systems.*

Appendices



Original BGE Player view of intact oil tanker, ocean and sky



Original reflection and normal map frames, courtesy of Sam Dearn (YouTube tutorials)

Use of resource-intensive water vertices in the mesh actually move)

Use of normal mapping and no shaders (the water is an animated flat plane)