Final Year Project: Specification

# Research Question

In the modern day games studio, artists and designers are often found using keyboard and mouse input to create scenes, art assets and such; for games. However, creative people have a tendency to work better with their hands. The keyboard and mouse input may limit their ability to do this.

I aim to create a simple tool (in the form of a terrain editing system), where the input is based upon the user within their 3D environment as well as using other inputs such as the users’ voice, creating an interface more in tune with its users’ tendencies this would result in the exploration of the users’ potential productivity gain and a potential higher quality of work. Where by the main complication in implementation will be finger tracking and hand gesture recognition, due to limitations in the Kinect hardware.

Posing the question, is current computer hardware limiting usability with its non-natural interface?

# Rationale for Project Choice

My inspiration for this project was found whilst on work placement at ‘Blitz Games Studios’. Whilst there I spent time working on their tool system (‘Blitz Tech’) as well as working closely with game teams and at points the Microsoft Kinect.

With this, I have first-hand experience of how an artist works and how a programmer creates software, for how they think an artist works, as well as experience with user interfaces, tools graphics/rendering and the Microsoft Kinect.

# Areas of Investigation

The main areas of investigation fall into the area of practicality within the workspace and getting useful and logical information from the user into the tool.

The problem with using the Kinect is that the user requires a given amount of workspace, most likely greater than the space a keyboard and mouse require. This means that the tool itself will need to work within confined spaces yet offer the freedom to the user to do what feels ‘natural’ to them.

As well as this, a keyboard and mouse offer more input states (buttons and keys) than the Kinect, which only offers visual and audio inputs (though, different words/phrases and physical gestures could be classed as the buttons and keys of the Kinect).

# Background Research

Whilst on placement at ‘Blitz Games Studios’ I was introduced to the Kinect hardware, its features and limitations. This along with working on Blitz’s tools/engine system and the Blitz team as a whole gave me the initial idea for the project. Whilst at Blitz I spent a few months working very closely with a game team whom where using the Kinect on an unannounced project. I also worked with the lead programmer on an unannounced sub-section of the tools.

# Literature Review

The Kinect will be required to work with finger detection, something nonnative to the Kinect API. The following papers explore different approaches to hand and finger gesture detection;

**Yuan, Junsong**. 2011. Robust Hand Gesture Recognition with Kinect Sensor

**Tang, Mattew**. 2011. Recognizing Hand Gestures with Microsoft’s Kinect

**Tang, Mattew**. 2011. Hand Gesture Recognition Using Microsoft's Kinect

**Li, Yi**. 2010. Hand Gesture Recognition Using Kinect

An abundance of techniques and theories exist on the <http://www.kinecthacks.net/> website. The website accepts Kinect applications and demonstrations from all over the world.

The API I will most probably use is [libfreekinect](http://openkinect.org/wiki/Main_Page) to avoid using the Microsoft Kinect SDK due to limitations in its implementations, however using the Microsoft Kinect SDK is easier to setup and distribute, so some investigation will be required.

As well as requiring specialized Kinect features, the terrain system itself will need to have some enhancements. Such as [real time terrain LOD (Level of Detail)](http://www.gamasutra.com/view/feature/3434/continuous_lod_terrain_meshing_.php) and other such real time terrain techniques like [terrain height based texturing](http://www.cosc.canterbury.ac.nz/research/reports/HonsReps/2008/hons_0801.pdf).

# Research Methodology

The project consists of two primary areas, the user input/interface and the terrain rendering. The user input/interface will benefit from surveys etc from people likely to use the software. Tools systems are created to improve productivity, however without the correct user interface or features available to the user they can easily become more of a hindrance.

The importance of user data is extrapolated by the fact the input devices is so physical and relies on human interaction. Due to this, surveys which explore words associated with given tasks, gestures and or movements associated with the tasks and what the user expects to be able to do within the tool will need to take place. The tool will work best with a large collection of data to best recreate the mean average of how a user thinks the tool should work.

Whilst at ‘Blitz Games Studios’ it was mentioned that some members of my team wanted to test and give feedback on the project, so some external feedback will also be collected.

# Project Plan

The project splits nicely into 2 main phases, as well as each phase splitting into easy to manage tasks. I will be using an online system called [asana](http://www.asana.com) for my project management.

Phase One – Terrain Renderer

The terrain system itself shouldn’t be too much work as it isn’t the main focus of the project as well as it already having the minimum required features to be used with the Kinect, so it makes sense to use my own rather than an external system. A simple terrain system with the ability to deform the landscape and paint vertices should service. However, should time allow a LOD (level of detail) system would be beneficial, as well as the ability to load in 3d meshes and position them about the scene. This phase should take no more than 4 – 6 weeks to complete.

Phase Two – Tool User Interface (Kinect)

The input system relies on making use of the Kinect in a manner it was not originally designed for, ‘finger tracking’. There are multiple techniques available for preforming such, each with its own pros and cons. Some with less accuracy but with a gain in performance, others with a near perfect accuracy, with the overhead of a large hit on computational time. Since I am not creating a game, more time can be spent on computation of a more accurate gesture recognition systems, however, the application should still be as near to ‘real-time’ as possible. Exploring this area and creating the best implementation that fits may take some time. As well as implementing the hand and finger detection, testing and gathering information about what gestures should be used and how successful/logical they are will take time.

# Project Deliverables

The end product will be a single exe along with any required assets (shades, models, textures etc…) which, with the use of the Microsoft Kinect peripheral, will allow the user to deform a flat terrain and texture said terrain.