Digital Portfolio

**HATtrick Games**

**Perth, W.A.**

**5/1/2013**

HATtrick Games

A collection and summary of the games created by the three founding members of HatTrick Games; Arran Ford, Hamish Carrier and Timothy Veletta.

Table of Contents

[Blizzard the Wizard 2](#_Toc356550198)

[GoldCrest 4](#_Toc356550199)

[Sanguine Wanderer 6](#_Toc356550200)

# Blizzard the Wizard



Developed as part of a four person collaboration (with the additional help of Liam Doherty), this game was created as a submittable assignment for the university course ICT207 Games Design and Programming. It is a top down third person arena shooter, where the goal is to simply survive as many waves of enemies as possible, using your spells to help you survive and kill monsters.

The game was created in C++ from scratch, using OpenGL 2.0 to handle all the graphics calls.

All assets and levels (barring sound and music) were custom created for this project by group members.



## Skillset Required

OpenGL 2.0 graphics API programming

3D modelling skills using Autodesk 3dsMax 2011

Texture Mapping skills using Adobe Photoshop

C++ programming

Knowledge of finite state machines

Ability to break down a complex task into different sections manageable by a single party

Teamwork and the ability to work collaboratively over a long distance

Time management skills

## Lessons Learned

That it is essential to have a proper breakdown of the work that will be required, mapped to a timeline, before starting. This allows the team to know when and if the project is slipping from the assigned deadline, without this we didn’t realise until shortly before the deadline and had to put in the extra amount of hours in a much shortened period of time.

Teamwork and communication need to be occurring continuously; this way if anyone is falling behind on their workload it can be shifted around the group and accommodated into the expected timeline. Without this communication it is hard to realise if the group is still on target or not.

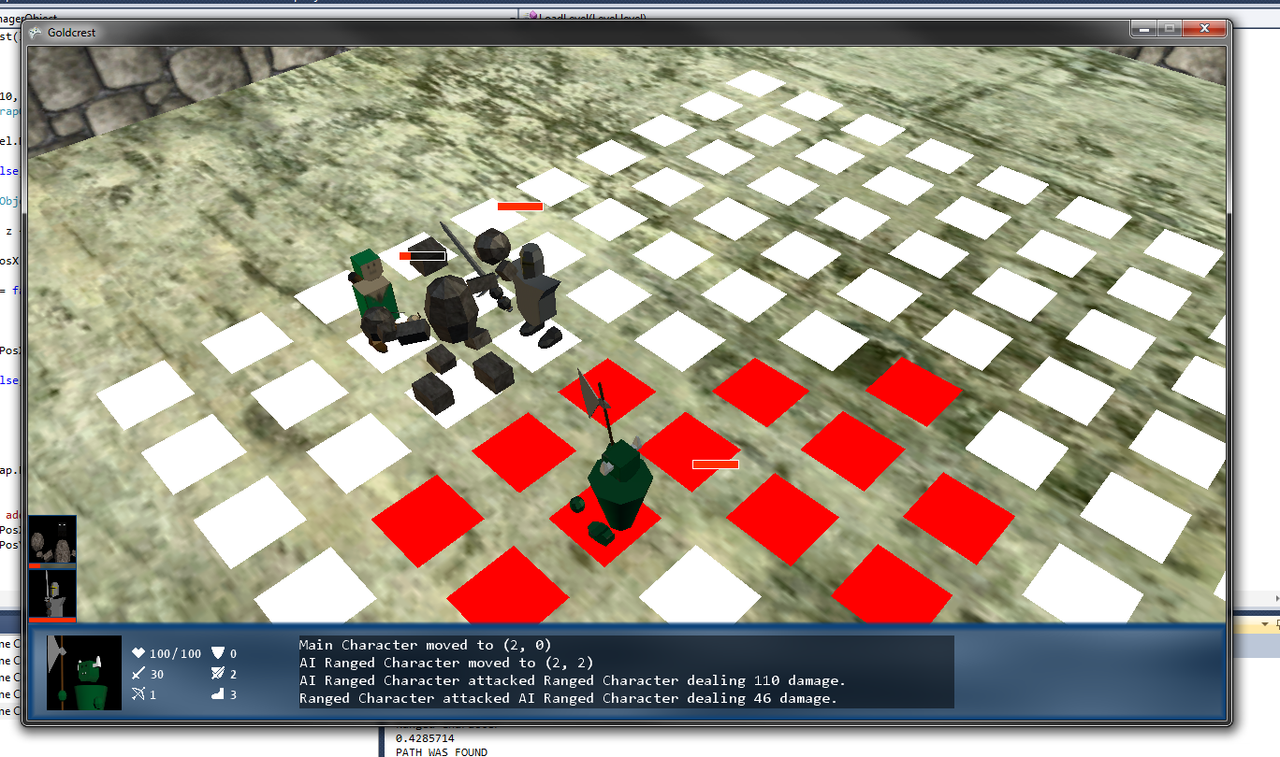
Creating our own assets while challenging, can save time in certain circumstances. Rather than attempting to deal with multiple different types of files and importing them, simply creating quick and easy custom made textures / models can save a lot of hassle on the programming side.

# GoldCrest



Developed solely by the three members of HATtrick Games, this game was created as a submission for university course ICT309 Games Development. This game is a turn based arena strategy game, where the goal is to defeat all the enemy units on the board without losing all of your own.

This game was created in C# using the Digital Rune engine, it was created with custom built assets made in 3ds max and Photoshop.



## Skillset Required

The ability to make use of existing libraries and engines

3D modelling skills using Autodesk 3dsMax 2011

C# programming

Texture Mapping skills using Adobe Photoshop

Ability to break down a complex task into different sections manageable by a single party

Teamwork and the ability to work collaboratively over a long distance

Time management skills

Video editing skills

Keyframe Animation editing

## Lessons Learned

That it is much quicker to work with an existing engine, than creating one from scratch. This allows for rapid development of a prototype and reduces the number of bugs that have to be dealt with.

Coming up with a simple modelling style can look better than an attempt at realistic modelling if the modeller is fairly unskilled at modelling. It also allows for much cleaner animations and texturing.

# Sanguine Wanderer

Developed solely by the three members of HATtrick Games, this game was created as a submission for university course ICT311 Game Engines. This game is a first person RPG, where the goal is to finish the story.

## Skillset Required

OpenGL 3.3 graphics API programming

The ability to make use of existing libraries in developing an in-house engine

3D modelling skills using Autodesk 3dsMax 2011

C++programming

Texture Mapping skills using Adobe Photoshop

Ability to break down a complex task into different sections manageable by a single party

Teamwork and the ability to work collaboratively over a long distance

Time management skills

## Lessons Learned

Multiple members should be accomplished in each area of the project, that way if the workload gets too large for one, it can be shifted to more members.

Time management needs to be a priority from the very start of the project, rather than only closer to the end.

Creating an engine from scratch provides for much more flexibility over the individual aspects of the game, though it comes at the price of large amounts of debugging and learning.