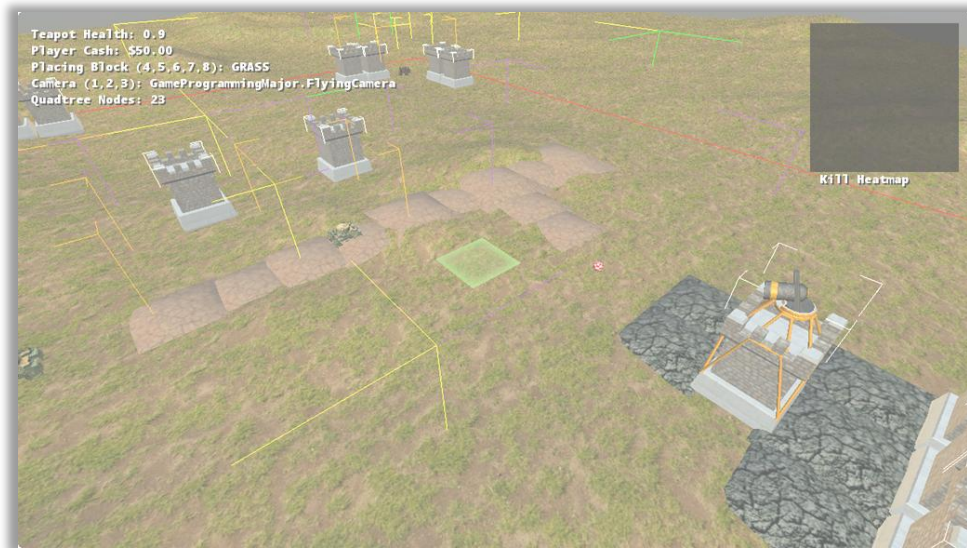


Team Cool **Bananas**

Assignment Two: **3D Game (Gold Master)**



Team Members:
11688025 Deinyon Davies
11685740 Daniel Nelson
11697822 Matthew Wale
11729631 Jesse Walker

Video

<http://youtu.be/NrISi22vXB0>

Introduction

Welcome to **Teapot Defence Simulator 2014 (TDS2014)**! TDS2014 is a fantasy-themed tower-defence styled game in which the player must stop enemies from destroying the elusive **Plasmatic Teapot**. Waves of enemies spawn in and make their way to the player's base, finding the most efficient path to get there. To find the most efficient path, these enemies navigate around walls and turrets, and have tendencies to stick to specific ground types that allow faster travel.

Controls

Operation	Action
Mouse Movement	<i>In FPS and Flying modes:</i> Rotate the camera
Left Mouse	Fire a projectile or place a Tower in the game world
Middle Mouse	<i>In Top-Down mode:</i> Press and drag to rotate the camera
W,A,S,D Keys	Translate the camera about the world
Shift	Increase the amount of parallel transfer of the camera or player
1 Key	Use an FPS Camera
2 Key	Use a Top-Down Camera
3 Key	Use a Flying Camera
4 Key	Use a Wall Tower for tower placement
5 Key	Use a Turret Tower for tower placement
6 Key	Use Grass for tower placement
7 Key	Use a Path for tower placement
8 Key	Use a Tar Blob for tower placement

How to Play

In the pre-designed level, the player is supplied with \$200.00 funds to set up defensive towers. The player is awarded funds when a turret destroys a tank. The player should use their funds to purchase towers, in order to kill and steer tanks away from the Plasmatic Teapot. Red (aggressive) tanks appear in later enemy waves, and will attempt to destroy turret towers that they find threatening.

Defend the Plasmatic Teapot until the tanks surrender.

Technical Features

The game exhibits a three-dimensional world composed of multiple static models, non-static entities, and NPC instances. The player can navigate the world using the keyboard and mouse to control the camera in one of three distinct camera behaviors.

The game performs bounding-sphere, AABB and ray collision detection for projectiles and tower placement. Ray-AABB collision detection was prototyped for turret line-of-sight detection, though the overhead resulted in low frame-rates.

The project exhibits over **45** distinct classes, of which some contain nested classes. A great deal of polymorphism has been employed to reduce code repetition and to greatly enhance maintainability. Development of new components is made easy by the high level of abstraction and comprehensive framework.

The game provides a sound queuing and storage system for modular and maintainable sound resource playback. The game plays a background loop, and various sound effects which are triggered by many types of dynamic events.

The program implements a **Quadtree** for two-dimensional spatial partitioning, which is used by each NPC's Projectile Manager to find collisions with Tank entities, and was tested with Turret Towers to locate nearby tanks.

An advanced **A* Pathfinding Algorithm** has been implemented, and is demonstrated in the waves of enemies that navigate through the level. The algorithm uses weighted search nodes to traverse the terrain using the fastest and most logical route. It is possible to re-form the search nodes by placing new Towers with the left mouse button.

Two XML configuration file structures are now integrated with the game: **Level Descriptors** and **Finite State Machines**. These configuration files allow the player to easily find and edit XML tags that will modify the layout and metadata (game parameters) of the level, as well as NPC behaviors. The Level Descriptor XML documents allow the player to modify and create new level layouts and enemy wave sequences. The Finite State Machine files define enemy states and transitions, allowing for **AI scripting**.

Subsequently, there exist **two** enemy types (passive tank and aggressive tank), which both exhibit multiple states through a **Finite State Machine** described in XML. Aggressive tanks target threatening Turret Towers, and fire at them to destroy them.

A custom written HLSL shader program has been composed that generates a plasmatic per-pixel effect with the trigonometric functions, and projects the effect onto a custom model using its texture coordinates.

The game boasts a **Game Analytics** feature, which displays an enemy "Kill Heatmap" as the game executes. The ground is a **Variable Height Terrain** that is defined by a bitmap. The program performs bilinear interpolation to find the height at any point on the terrain surface.

Peer Evaluation

Deinyon: Lead Programmer, Graphics & Documentation Expansion

Score: 5.0

Contributed the majority of the codebase and framework, implemented a Quadtree, terrain and Finite State Machine system.

Daniel: Lead Design, Programmer, Debugger and Balance

Score: 5.0

Researched and developed prototype implementations for many core game components, including the Wave Manager system and level loading. Contributed greatly in debugging and game design.

Matthew: Design, Documentation & Administration

Score: 5.0

Organised, wrote, and compiled several elements of the read-me document. Recorded and edited the video for the assignment, along with uploading to YouTube. Participated in group design meetings.

Jesse : Documentation & Designer

Score: 4.0

Participated and helped with group discussion, contributed lab code to the project, and helped with elements of the read-me document.

References

Tutorial & Reference:

Amit's A* Pages 2010, *Pathfinding*, viewed 3 September 2014, <<http://theory.stanford.edu/~amitp/GameProgramming/index.html>>.

Gamedev Tuts+ 2013, *Understanding Steering Behaviors : Collision Avoidance*, viewed 4 September 2014, <<http://gamedevelopment.tutsplus.com/tutorials/understanding-steering-behaviors-collision-avoidance--gamedev-7777>>.

Patrick Lester 2005, *A* Pathfinding for Beginners*, viewed ~25 August, 2014, <<http://www.policyalmanac.org/games/aStarTutorial.htm>>.

Textures:

Jacobo Cortés Ferreira, *Grass0139_33* (resampled), viewed 3 September 2014, <<http://www.cgtextures.com/texview.php?id=44371&PHPSESSID=pegmiou3bj4bkrbgvc54jvna25>>.

CGTextures, *BrickOldRounded0298* (Resampled), viewed 3 September 2014, <<http://www.cgtextures.com/texview.php?id=52593&PHPSESSID=j6fcl84hlqk61c723qjsdstsq4>>.

CGTextures, *SoilCracked0157* (Resampled), viewed 13 September 2014, <<http://www.cgtextures.com/texview.php?id=66844&PHPSESSID=eimacm8d4ftmecu9p3642eo1b4>>.

R. Boel, 2006, *Sun altitude of 5 degrees above the horizon*, Roel z'n Boel, viewed April 19 2014, <<http://reije081.home.xs4all.nl/skyboxes/>>.

Pokemopolis Forum, *Awesome Smiley*, viewed 13 July 2010, <<http://captionsearch.com/pix/thumb/nqfzkzcwd-r.png>>.

Audio:

Valve Corporation 2004, *metal_barrel_impact_2*, sound effect, Half Life 2 Resource.

Valve Corporation 2004, *shotgun_fire6*, sound effect, Half Life 2 Resource.

Valve Corporation 2004, *smg1_fire1*, sound effect, Half Life 2 Resource.

Valve Corporation 2004, *towm_ambience*, sound effect, Half Life 2 Resource.