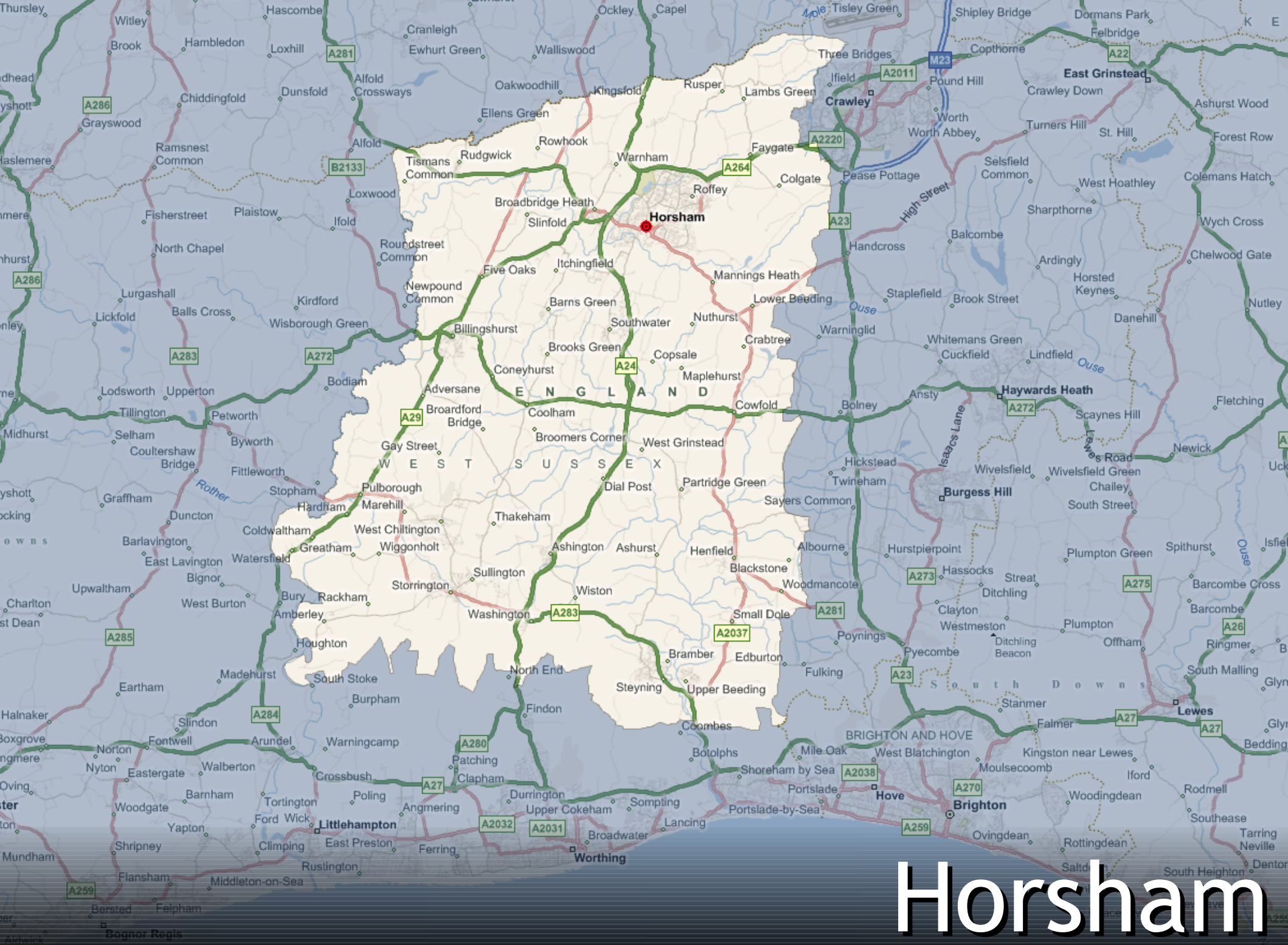




Industrial Placement 2010



Outline



Horsham



The Workplace



The Company



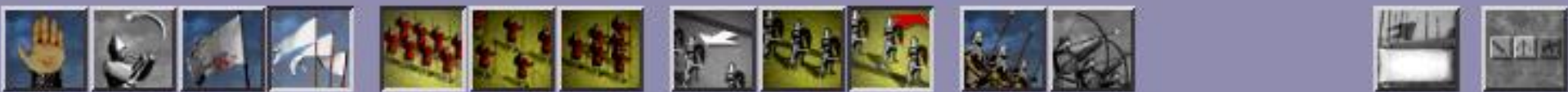
The Games



Shogun: Total War (2000)



Shogun: Total War (2000)



37 CHIVALRIC KNIGHTS
CAVALRY POOR IN WOODS
FIGHTING (WINNING EASILY)
IMPETUOUS (EAGER AS THE ENEMY ARE FLEEING)
QUITE FRESH

Medieval: Total War (2002)



Medieval: Total War (2002)



Rome: Total War (2004)



Athens - Large city

 Governor: Sextus Marcius, Age: 37
Command: ★
Management: 
Influence: 

Settlement Details

Income	4620	Population	14148
Public Order	+110%	Population Growth	+15%

Population required for next level of settlement: 24000
Automanage ☐

 High tax rate  Yearly Games

Construction	Recruitment	Repair	Retrain	
				
				

Construction Queue

Army City Agents

 Large city Athens
6215
2195

Rome: Total War (2004)



Medieval 2: Total War (2006)



Medieval 2: Total War (2006)



Empire: Total War (2009)



Empire: Total War (2009)



Empire: Total War (2009)



Napoleon: Total War (2010)



Napoleon: Total War (2010)

將軍

SHOGUN

TOTAL WAR™

2

中島巖

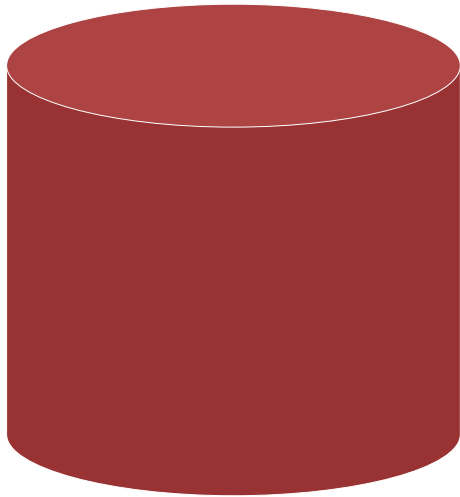




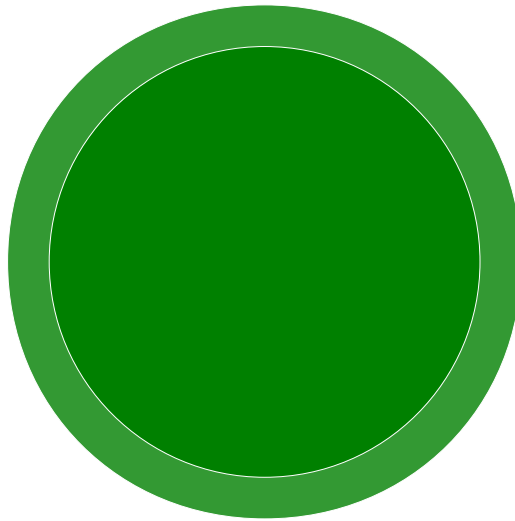
Infrastructure



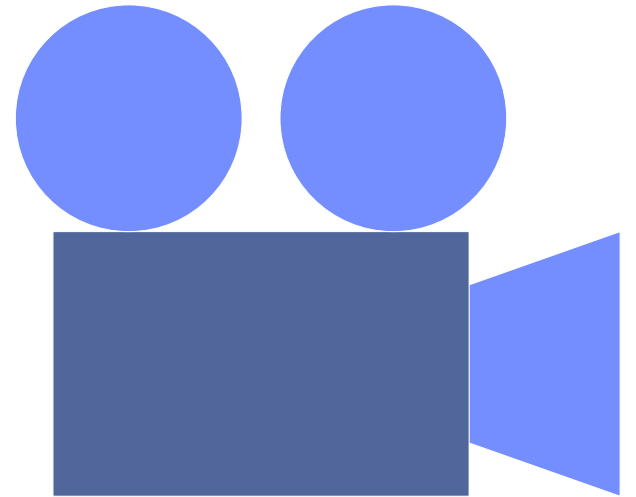




DaVE



BOB



Cindy



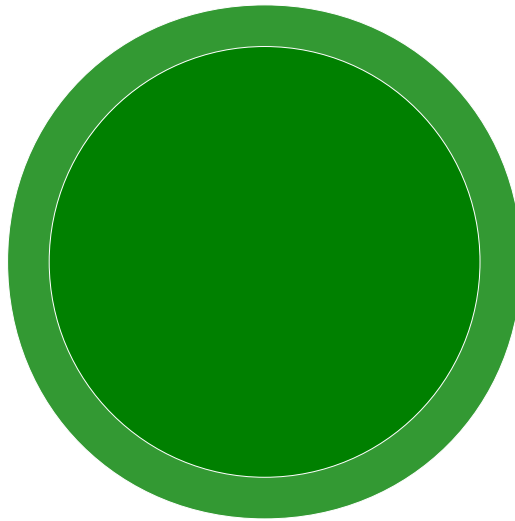
XML Database & Editor



Battle



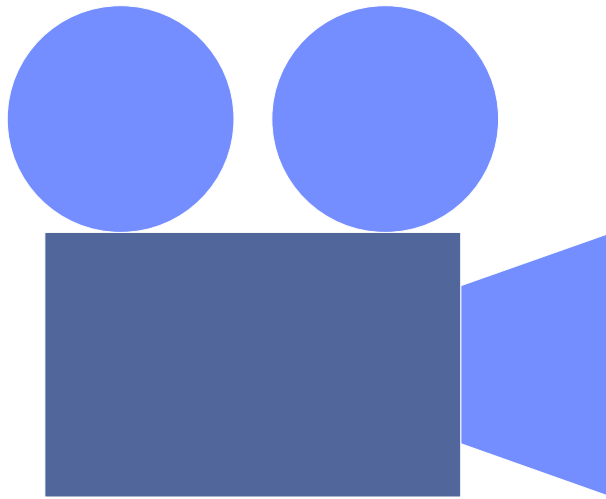
Campaign



Build on One Button



Ships



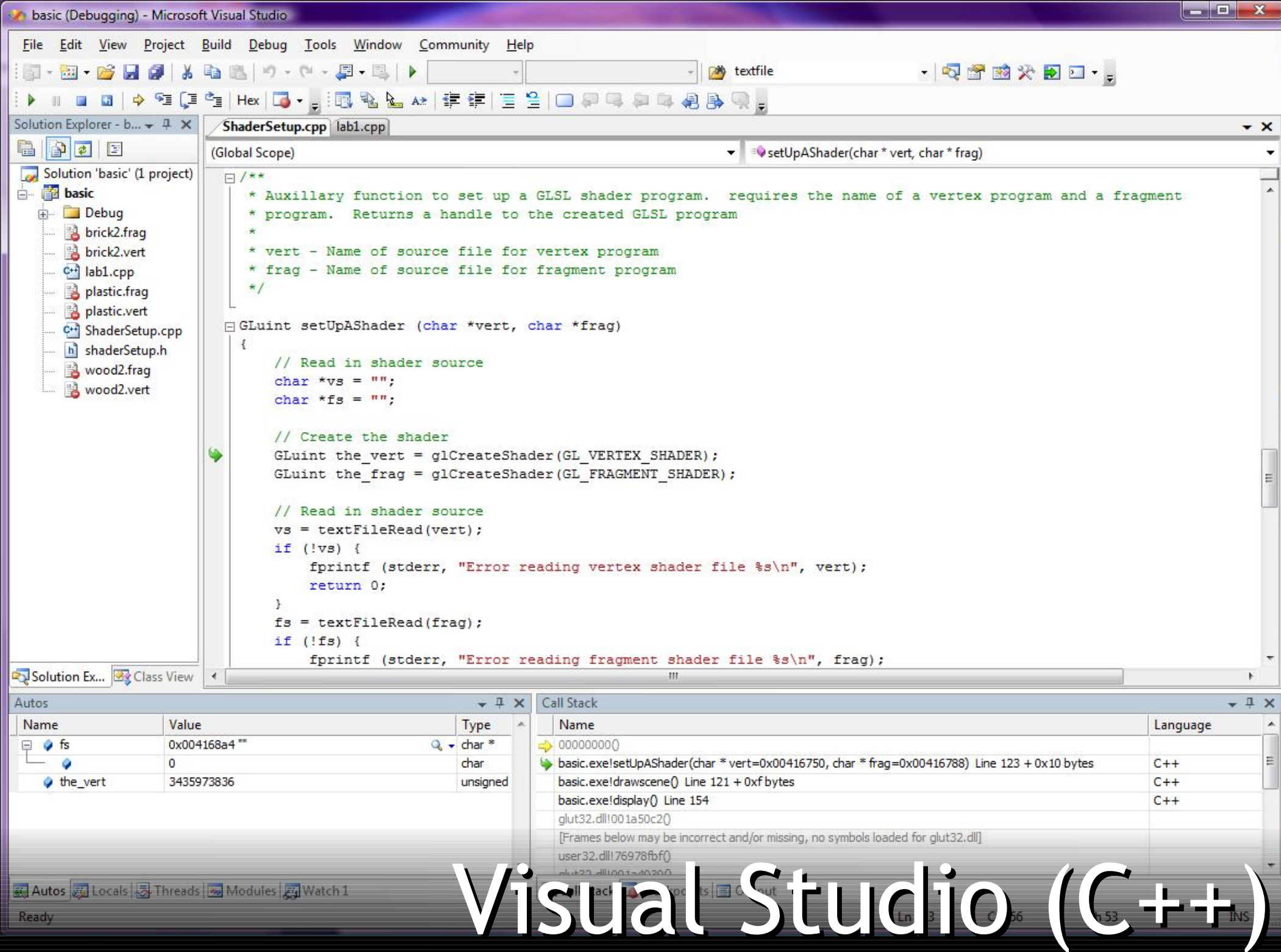
Cinematic Editor



Gameplay Trailers

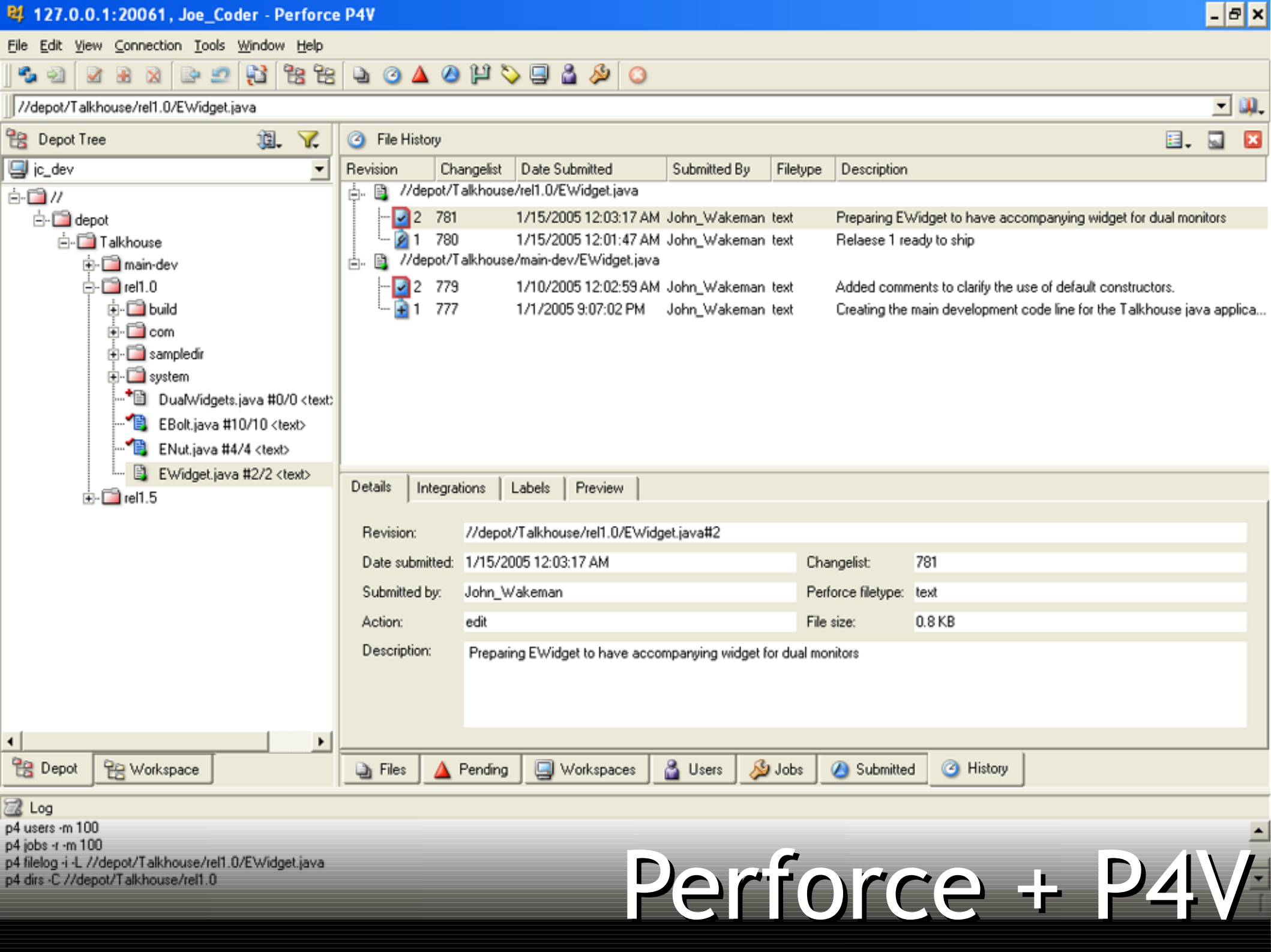


何か問題
がありますか？





IncrediBuild



//depot/Talkhouse/rel1.0/EWidget.java

Depot Tree

jc_dev

//

depot

Talkhouse

main-dev

rel1.0

build

com

sampldir

system

DualWidgets.java #0/0 <text>

EBolt.java #10/10 <text>

ENut.java #4/4 <text>

EWidget.java #2/2 <text>

rel1.5

File History

Revision	Changelist	Date Submitted	Submitted By	Filetype	Description
//depot/Talkhouse/rel1.0/EWidget.java					
2	781	1/15/2005 12:03:17 AM	John_Wakeman	text	Preparing EWidget to have accompanying widget for dual monitors
1	780	1/15/2005 12:01:47 AM	John_Wakeman	text	Release 1 ready to ship
//depot/Talkhouse/main-dev/EWidget.java					
2	779	1/10/2005 12:02:59 AM	John_Wakeman	text	Added comments to clarify the use of default constructors.
1	777	1/1/2005 9:07:02 PM	John_Wakeman	text	Creating the main development code line for the Talkhouse java applica...

Details Integrations Labels Preview

Revision: //depot/Talkhouse/rel1.0/EWidget.java#2

Date submitted: 1/15/2005 12:03:17 AM Changelist: 781

Submitted by: John_Wakeman Perforce filetype: text

Action: edit File size: 0.8 KB

Description: Preparing EWidget to have accompanying widget for dual monitors

Depot Workspace

Files Pending Workspaces Users Jobs Submitted History

Log

p4 users -m 100

p4 jobs -r -m 100

p4 filelog -i -L //depot/Talkhouse/rel1.0/EWidget.java

p4 dirs -C //depot/Talkhouse/rel1.0

Perforce + P4V



Hey guys, thanks for coming - my name is Hok Shun Poon. I worked at games studios called **The Creative Assembly** for my 3rd year industrial placement from April to September of this year.

<< pretend to switch slides >>

BTW, if you're wondering why I have a picture of Himeji castle up, I can tell you now that's not where I worked, though I'm sure many of the guys would like that idea."

[0:20]



So, in the next 12 minutes, I'm going to show you

Where I worked,
Who I work for,
What we do,
What we're working on,
How we work,
And finally
What I thought about it all.

(By which time you'll understand why all the Japanese references!)

[0:35]



We follow a map with another! Here is the location of Horsham in West Sussex. The highlighted area is Horsham District.

Just for reference, here's Brighton; and above Crawley there is Gatwick Airport. That'll probably give you a rough enough idea.

[0:50]



And somewhere in Horsham you will find this this building by the name of **Spire Court**.

The **second floor** is currently home to the main UK offices of **The Creative Assembly**, where I worked. I say the UK offices, because we have an Australian branch as well in **Fortitude Valley** in **Brisbane**.

We hire about 140 developers at the moment, but are seeking to expand. In the foreseeable future, we will take the **first floor** of the building as well.

[1:40]



So a bit about the company then.

The Creative Assembly is wholly owned by Sega as of 2005. So strictly speaking, that actually made me a Sega employee.

It generally goes like this: Sega **gives us money**, we **make the games**, and Sega **markets, publishes** and **distributes** them all over the world.

Part of the revenue that Sega makes comes back to us, in the form of our salaries or funding on future projects.

We make both PC and console games, but we are probably best known for a PC game series called **Total War**.

[2:30]



There are 6 major releases of **Total War** as of today.

Since it'll be relevant to what I worked on, I'm going to show you a Brief History of **Total War** via a **blitzkrieg** of screenshots, and hopefully that'll **illustrate** the evolution of the series over its 10 year history.

So, let us take a step back in time!

[3:00]



It really all **started with this guy**.

One half of every **Total War** is essentially an epic battle simulator – one that supports **thousands of men** on the battlefield at one time, even if our earliest game does so with the help of **sprites** on a **3-dimensional landscape**.

[3:30]



The other part of **Total War** is the **campaign** map.

This is a turn based element of the game that gives **context** to the battles, where the player may conduct army movements, stage invasions etc.

[3:40]





While **Shogun** was set in Japan, the next installment was **Medieval** used pretty much the same game engine that **Shogun** was using, but was set in Medieval Europe.

[3:55]



Rome: Total War was the first of the Total War game to support thousands of **3D units**.

The BBC even made the game show '**Time Commanders**' using a special version of this engine.

[4:10]



Rome: Total War (2004)

The campaign map moved onto using a **grid based system** rather than Risk-style provinces.

[4:20]



Medieval 2: Total War (2006)

Medieval 2: Total War built upon what **Rome** accomplished with many graphics enhancements.

[4:30]



We began the use of matched combat animations, adding more **detail** and **realism**.

[4:40]



Empire: Total War (2009)

Empire takes a little **stride** forward in history, and follows the **early modern era**, where warfare was conducted more with **mortars** and **guns** than swords and arrows.

[4:50]



Empire: Total War (2009)

Here you see enhancements such as volumetric cloud and smoke effects, a bit of depth of field was in **Empire** too.

[5:00]



The **first naval battles**... since rendering nice looking water became a lot more feasible on graphics hardware of recent times.

[5:10]



And finally, we get to Napoleon - our latest release. It covers the Napoleonic Wars, and reportedly, some schools in France have been using it as an educational tool.

[5:20]



Napoleon: Total War (2010)

This really shows that we're now making games which **render out** pretty **photo-realistic** battle scenes in real-time.

[5:30]



So, that's the status quo of **Total War**.

We **don't intend to stop there, though**.

[5:35]



Shogun 2: Total War. It's the official sequel to the first **ever** Total War game, to be set in 17th century Japan.

I joined CA in April, and I couldn't have asked for a better time to join the project, because **The Creative Assembly** started work on it soon after release of Napoleon in February. This means the office was full of **energy**; everyone was working **full steam ahead** to ensure we were ready to show a **pre-alpha** gameplay demo for E3 in June.

The Creative Assembly have effectively accumulated **10** years of game development experience to **revisit the era** that began the **Total War series**. So in short, for me, and for the company, it's a very **exciting title**, and a very **time** indeed.

[6:40]



I joined the ranks of the **Infrastructure team**. We are one of five other programming teams that form the the **programming effort** behind **Shogun 2**.

They are namely **Battle, Campaign, Graphics, Sound** and **User Interface**.

The job of the **Infrastructure team** is to support vital software tools used by our **artists <switch slide>!, animators, game designers <switch slide>!, quality assurance guys**, and **media people**, facilitating automated data pipelines.

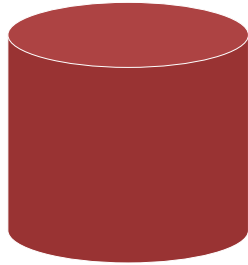
[7:00]



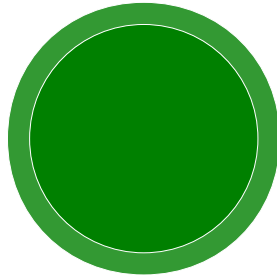
[7:10]



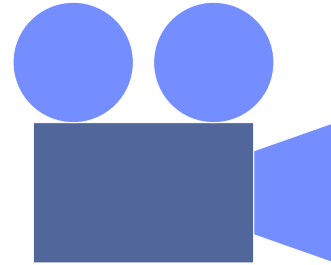
[7:20]



DaVE



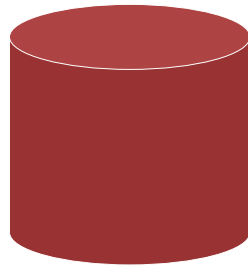
BOB



Cindy

I personally worked on **three** particular projects.

[7:25]



XML Database & Editor

The first is the **XML relational database**. I'll show you later why we need this in the development of **Shogun 2**.

The **main reason** we use XML as a data format is because it allows for merging of data changes across different development branches, and **versioned data replication** for multiple branches and developers is relatively simple to achieve, in contrast to client-server centralized database systems such as **M\$ Access**.

The Database Visual Editor (which we affectionately call DaVE) is a GUI frontend to access the database, hiding the **complexities of editing XML** for our **game designers**.

[7:30]



On the battlefield, there's quite a number of different types of data that needs to be stored:

unit statistics

unit special abilities

Types of projectiles available to which units, and the **particle effects** that they fire,

Locations of buildings on the battlefield

Types of trees to be displayed for any given **climate**,

You get the drift. I helped improve the visual editor so that designers can change these things more efficiently, such as improving filters function better, making useful tooltips about fields with relationships.

[8:00]



Another **example** of where the database is used in-game is the campaign:

initial starting positions of units

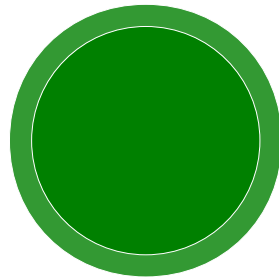
missions that the player will receive

building technology trees for **settlements** etc.

The campaign guys very often ask for **changes to table schemas** (adding of fields etc.) with new features they're adding to the game. But unfortunately, when I joined CA, DaVE was lacking a schema editor.

I learned that this process was very time consuming to do manually because of the number of files that needed to be edited. I started automating the process incrementally, initially using Haskell to aid my XML editing, and later port all of my program's functionality into DaVE as a rudimentary visual schema editor. This removes any human error, and ensures that the integrity of the database is maintained through edits.

[8:30]



Build on One Button

Then there's the problem of **building game data**.

In the company, **models**, **textures**, **animations**, **XML data files**, sounds, language localised strings that our game designers, artists, animators, produce are NOT the files that the game reads.

That's because these **raw data** files are inefficient for the game to read, and therefore should be converted into what we call **working data** before it can be used by the game.

But sometimes these conversions are not trivial. Sometimes there are complicated **data dependencies**. BOB automatically builds a dependency tree of raw data, and it ensures the data is built in the right order, simplifying the data building of the game to the invocation of **one button**.

BOB is built with extensibility in mind, by means of **plugins**. Different plugins handle different data types.

A simple example is the database processing plugin. The XML database we just saw needs to be converted into a binary format before the game can use it, because reading and parsing XML is too time consuming.

[9:00]



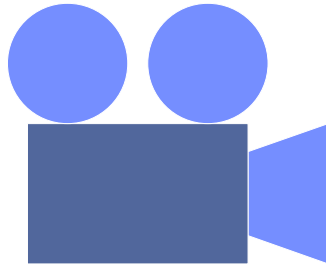
One good example of a plug-in is that of **ships**. They are complex game asset, because of their many components:

The
basic ship model,
oar positions,
rowing animations
destruction animations depends on the mesh
path finding waypoints might depend on the mesh
logic of how the men move on board the ship depends on pathfinding
waypoints etc.

And with the help of some metadata, BOB is able to automate each of these conversion processes, avoiding human error.

I have helped write and maintain a number of these plugins, including the ship plugin, unit equipment processing, audio processing and packing, and rigidmodel progressing for when the raw data file format changed.

[9:40]



Cinematic Editor

The game doesn't have its own built in system for making movies. However, Cindy uses hooks in the game to controls various values in the scene.

These values maybe the position of the camera, the position of its look-at target etc.

[9:00]



So in that sense, you may think of it as a virtual camera rigging system. But actually you can do **much** more with other values, like changing the colour of the ground, or water, or the sun.

I made a few enhancements to Cindy, made it **draw some lines** to indicate the bounds of the output video with 2.35:1 aspect ratio and added **real time colour picking** capability, after which I had a bit of fun changing the colour of the sun very rapidly. The effects are psychedelic.

[9:15]



Movie
[11:00]



So, that's that. Within 10 years, **The Creative Assembly** went from being just a logo at the back of the retail box for Shogun: Total War to becoming my employer of 6 months.

Even if I didn't really work at Himeji castle this summer, the experience of being accepted into my favourite games company and being part of the legend is probably just as surreal.

I thank CA for realizing a little slice of my childhood dream, and thank you all for listening to me talk about it.

[12:00]

何か問題
がありますか？

Movie.

[11:30]

