Dominic Ashill

A-Level Computing Project

Contents

Section 1 – Analysis

* 1.1 - Problem Definition
* 1.2 – Research into the topic
* 1.3 – Requirements Gathering
* 1.4 – Hardware and Software requirements

Section 1 – Analysis

1.1 - Problem Definition

Using the Panda3D game engine, I will be developing a realistic but accessible tank combat game, based on previous games in the genre, but with improvements made derived from interaction with my intended end users. Tank games fall on a spectrum of simulation, detailed in the research section (1.2) below.

1.2 - Research into the topic

Several different video games feature tanks. However, they simulate them to different levels of accuracy. Most games of this genre are arcade in nature, and simulate vehicle systems only to a slight degree. A vehicle’s state is abstracted to a hitpoint pool, guns often only have single ammunition choices, and there is no representation of the crew’s interaction with the vehicle. Often they are afterthoughts in the game’s design; such as the many vehicle sections in the Call of Duty series games. Notably in Call of Duty 2 (see figure 1), the player is turned out all the time, something which would not be recommended in real life!



*Fig. 1. A player commanding a Crusader tank in Call of Duty 2. The player commander has control over all functions of the tank, but the functions are limited to driving the vehicle, manipulating the turret and firing the gun. This particular section is a good example of a vehicle section in most games, with not nearly as much depth of gameplay as the infantry combat. (1)*

However, some games go into more detail about tanks whilst still retaining infantry combat. They may simulate different ammunition choices, armour angling, shell ballistics and multiple mannable crew positions, but they often only select a few of these things. One good example is the Armed Assault series of games. Multiple crew positions can be manned in a tank, with the other positions being occupied by either AI units, other players when online, or empty. Additionally, “turning out” of the vehicle is simulated, different ammunition choices for different situations are present, and different parts of the vehicle can be damaged to the point of non-function, like the tracks and turret. However, the damage modelling is still simplistic, with the status of the vehicle still mostly being decided by hitpoints, and the vehicles not having modelled interiors. Whilst the ballistics of the rounds themselves are simulated to a good degree, the interaction between the vehicle and incoming projectiles is quite simple – if the round can penetrate the tank, which has a single armour value, then damage will be dealt, with module damage perhaps occurring if the round strikes lucky. It is to be noted in figure 2 that the gunsight is shown – it is a flat 2d plane, with no way to actually interact with the interior of the vehicle.



*Fig. 2. Looking down the gunsight of a Merkava main battle tank in Arma 3. The screenshot is taken whilst in the gunner’s position, with a player commander and driver potentially being in the vehicle as well. Notice the icons on the top left representing the damage state of various parts of the vehicle like the tracks and gun. (2)*

In contrast to the games listed above, some games are specifically designed to be based around tanks. These games mostly discard the infantry aspect of warfare and focus exclusively on vehicles. War Thunder does exactly this. It has realistic armour and vehicle modelling, well-simulated round ballistics, suspension, engine and gearbox simulation, crew simulation, and extremely good interaction with the game world (shown in figure 3). Unfortunately, War Thunder does not model the interior of vehicles, and the player also controls what can only be assumed to be the entire crew, and not from a commander’s perspective.



*Fig. 3. A kill camera screenshot from War Thunder. This shot illustrates a shell penetrating the turret of an unfortunate KV-85 driver. The kill camera shows the module modelling of War Thunder, with spall fragments striking fuel tanks, ammunition and the gun’s breech. Additionally, the excellent round modelling is shown, as the shell fragments inside the tank. (3)*

Despite the overwhelming majority of games featuring tanks being survey-style (meaning simulating a large amount of vehicles to a limited degree of accuracy), some tank games have been study-style (simulating one or a couple of vehicles to a very high level of accuracy). The leading example of this type is Steel Beasts Pro PE. It is the commercial version of simulators used by real-world militaries to train tank crews. Everything is simulated, from vehicle optics to round penetration, and most importantly, the player is inside a vehicle with a modelled interior (as shown in figure 4). The player interacts with the vehicle by using the buttons, cranks, joysticks and viewports that would be present on the real vehicle, and also has the option of “turning out”.



*Fig. 4. In the commander’s seat of a Leopard 2 in Steel Beasts Pro PE 4.0. Note this is the first time we have seen proper interiors; all vehicles intended to be playable in SB have modelled interiors. Most things that can be interacted with on the real vehicle can be interacted with in this digital rendition. (4)*

However, whilst Steel Beasts is a very good simulator, it has drawbacks. For one, it is not very accessible to newcomers, with the controls being obscure, and in some cases having to have the actual vehicle’s manual to be able to interact with the various systems. Additionally, it is extremely expensive, with one year’s license costing £100. Clearly, there is room for improvement here.

With this, I have identified a niche in the tank game market. The only games that offer meaningful crew interaction with the vehicle are high-end simulation games. The only games that have a somewhat casual approach to the tanks are too unrealistic to scratch a simulation “itch”. Thus, there is a demand for a casual tank game, where the operation of the vehicle is modelled, but to an accessible degree. This will be the main focus of my project. But, the degree of simulation to which other parts of the vehicle are modelled needs to be decided upon. How will be ballistics be modelled? How complicated should the interior of the tank be? For this, I will need to ask my end users, and this is what is detailed in section 1.3.

1.3 – Requirements Gathering

My end users will be anyone who wants to play the game. They are likely to be interested in Tank games. Thusly, I will be corresponding with players of tank games from around Europe. I will be contacting them to ask for their input as it concerns to user requirements and feedback throughout development of my project. Many different levels of simulation of different systems can be included in tank games, so I will be asking my end users what they liked from their previous experiences with tank games, so I can take this into consideration when programming my project. The interviews conducted to gather this information are presented below.

First Interview – Experience with other games

My first interview will take the guise of a questionnaire, based around the end user’s prior experience with tank games and what they enjoyed about them. The questions follow:

1. Have you played any games involving tanks?

-Yes

-No

1. Please list the games below.
2. Did you particularly like any of the features of the games you played as they pertain to tanks?
3. Did you particularly dislike any of the features of the games you played as they pertain to tanks?
4. Are there any features you would want to particularly see in a game based around operating tanks?
5. Would you prefer the game to be more realistic or more arcadey?

-Realistic

-Arcadey

Results of the first interview

Here all the responses are compiled into results. Frequency of results for yes or no questions are shown in parenthesis, and all long answer questions are shown. The four tank game players whom I asked to fill out this questionnaire have agreed to be my user group for this project, and their names follow: Tom Key, Julius Salonen, Kalle Toukonen, and Anders Svensson.

1) Have you played any games involving tanks?

Yes (4)

No (0)

2) Please list the games below.

War Thunder, World of Tanks, ARMA Series, Red Orchestra

Armored Warfare, World of Tanks, Arma & Operation Flashpoint

Arma series, World of Tanks, Armored Warfare, War Thunder, Tokyo Wars, Battlefields, Steel Panthers 2

Red Orchestra 2, ArmA 2 and 3, War Thunder, World of Tanks

3) Did you particularly like any of the features of the games you played as they pertain to tanks?

Vehicle Handling & Sound

Cookoffs are very satisfying, both the turret flip kind and the slower burn kind. Smoke clouds billowing out of burning vehicles for some time like in Wargame: European Escalation is great. Tanks having more weak spots than just general front/side/back armor is nice, as is being able to damage subsystems and individual crew.

Red Orchestra 2 and ArmA are nice because they expect you to interact and communicate with your crewmen (actual players) in order to perform well. RO2 (and some vehicles in ArmA) also have vehicle interiors which is great for feeling claustrophobic (adding up to the immersion). War Thunder and World of Tanks lack what RO2 and ArmA may provide, but it's still enjoyable by the amount of detail there is on the tank models.

4) Did you particularly dislike any of the features of the games you played as they pertain to tanks?

Nothing.

Actual tank controls (controlling both tracks separately) is not very fun, "spot the pixel" gameplay is not very enjoyable.

Health bars are iffy in tank combat (module-based damage is much more visceral), reconnaissance is nonexistent in asymmetric scenarios, lack of feeling of power

RO2 has great tanking mechanics from the little I've played and I've heard about the mod Darkest Hour for RO1 which apparently is even better when it comes to tanking. Can't say I disliked anything about it. Tanking in ArmA isn't as exciting as RO since it's not WWII anymore. The tanking experience can easily be abrupted by ATGM's fired from either other armored vehicles or infantry. It can't be helped I guess.

5) Are there any features you would want to particularly see in a game based around operating tanks?

Loader Position as Playable, First Person interior control from all positions.

Cookoffs, if multiplayer then multiple people in the same vehicle.

Can't think of anything.

6) Would you prefer the game to be more realistic or more arcadey?

Realistic (4)

Arcadey (4)

Analysis of the results of the first interview

The main thing I took away from this questionnaire is that the players I questioned preferred realism in their tank combat to arcade features. This is primarily shown by the overwhelming realism vote in question 6, but specific responses to the preference questions also show this. “First Person interior control from all positions” is an example of this, but other answers show that realism in modelling of the vehicle as well is a desirable feature for the program to possess. For example, “damage subsystems and individual crew” is something that is thoroughly in the camp of realism. Thusly, I will build an emphasis on realism into the design specification.

However, there is some doubt about absolute realism in the project. “Actual tank controls (controlling both tracks separately) are not very fun” points to specific aspects of realism being unwarranted and unfun in a game context, so this points to a second interview being needed to narrow down the required realism.

Additionally, there was conflict over the desired time period that the game would be set in. Elements of modern warfare like anti-tank guided missiles (ATGMs) were stated as being unfun, so I will ask another question in the second interview regarding the time period that the game should be set in. This also has impacts on the internal systems of the vehicles.

One desired feature that will most likely not be feasible is multiplayer inside vehicles. It was desired in the initial interview, but it will most likely not be time-efficient to include LAN support in the game.

In review of these answers, I drafted a requirements specification for my project:

* The player will be able to take control over several different crew positions inside a generic ww2 era tank.
* AI crewmembers will occupy crew positions that the player doesn’t.
* The tank should have accurate systems modelling IE armour thickness, gearbox, powerpack, cannon and optics modelling.
* The tank should have basic vehicle controls, without overly detailed steering modelling.
* Good vehicle ballistics should be present IE shell penetration, different shell types and accurate ballistics.
* The player should be able to interact with objects inside the vehicle, IE hatches, optics, movement of ready rounds, and turning in and out with binoculars
* A basic terrain of around 2x2km2 for the player to move the vehicle around in
* AI enemy vehicles to act as opponents to the player.
* A basic mission to complete.
* A menu screen and basic game settings.

Second Interview – detailed requirements

In order to determine if my user group was happy with the requirements I had derived from the previous interview, I composed a second questionnaire. The user group was given the requirements specification I drafted, and was given a questionnaire based around the requirements specification. Extra detail was given to the points detailed in the analysis of the first interview. Additionally, some technical questions were given.

The questionnaire given is shown below:

1. Were you happy with the requirements specification given for the project?

-Yes

-No

1. Would you make any changes to the requirements specification?
2. Would you prefer a more or less realistic modelling of the engine and gearbox?

-More realistic

-Less realistic

1. Would you prefer interaction with the interior of the vehicle by direct mouse control or keybinds?

-Direct Mouse Control

-Keybinds

-Both

1. Would you prefer the game to run in fullscreen, windowed, borderless windowed, or selectable?

* Fullscreen
* Windowed
* Borderless Windowed
* Selectable

1. What time period would you prefer the game to be set in?

* WW2
* Early Cold War
* Modern

1. How should the AI be controlled by the player?
2. If an AI moves to the commander’s seat of the vehicle, should they or the player retain command of the AI?

* Player
* AI

Results of the second interview

The results of the second interview are compiled below. The frequency of responses are detailed below.

1. Were you happy with the requirements specification given for the project?

Yes (2)

No (2)

1. Would you make any changes to the requirements specification?

Make the map larger (5x5km2)

Turning in and out should be done by a keybind, not mouse interaction.

1. Would you prefer a more or less realistic modelling of the engine and gearbox?

More realistic (3)

Less realistic (1)

1. Would you prefer interaction with the interior of the vehicle by direct mouse control or keybinds?

Direct Mouse (0)

Keybinds (0)

Both (4)

1. Would you prefer the game to run in fullscreen, windowed, borderless windowed, or selectable?

Fullscreen (0)

Windowed (1)

Borderless Windowed (0)

Selectable (3)

1. What time period would you prefer the game to be set in?

Second World War (2)

Early Cold War (1)

Modern (1)

1. How should the AI be controlled by the player?

The player should have a command rose available, with context-sensitive commands over whatever the player points at. Vehicle movement should be by WASD commands to the driver.

AI commands can be done by a command menu, with map interaction

1. If an AI moves to the commander’s seat of the vehicle, should they or the player retain command of the AI?

Player (4)

AI (0)

Analysis of the second interview

50% of users were happy with what I detailed in the project requirement. The 50% that were unhappy appeared to differ on minor technical points. However, their input must be recognised. Below follows the final requirements specification:

* The player will be able to take control over several different crew positions inside a generic WW2 era tank.
* AI crewmembers will occupy crew positions that the player doesn’t. The player will keep command of the AI, and will be able to issue orders via a context-sensitive command rose.
* The tank should have accurate systems modelling IE armour thickness, gearbox, powerpack, cannon and optics modelling.
* The tank should have basic vehicle controls, without overly detailed steering modelling.
* Good vehicle ballistics should be present IE shell penetration, different shell types and accurate ballistics.
* The player should be able to interact with objects inside the vehicle, IE hatches, optics, movement of ready rounds, and turning in and out with binoculars. The method of control will be by both keybinds and a mouse interaction option.
* A basic terrain of around 4x4km2 for the player to move the vehicle around in
* AI enemy vehicles to act as opponents to the player.
* A basic mission to complete.
* A menu screen and basic game settings. The game’s resolution, window type, and important keybinds will be changeable.

1.4 – Hardware and Software requirements

Based on the preferences that my end users had in their interviews, I have compiled a complete specification detailing what my game needs to do.

* The player must be able to sit inside, interact with and command the crew of a fictional WW2 period medium weight class tank. The specifications of this vehicle follow.
* The tank must have semi-realistic engine, gearbox, transmission, Christie-style suspension and steering simulation. The tank must be able to turn left and right via keyboard keys or mouse input by selecting the direction to orient the tank, modelling differential steering. It must be able to drive forward through different gears with optimal RPMs and speeds, up to a maximum speed on level terrain of 65km/h, and with a maximum reverse speed of 5km/h. The tank must have four forward gears and one reverse gear. The tank must be drivable from the driver’s position.
* The tank must have a simulated turret, able to traverse left and right, either by power assisted traverse or manual (crank) traverse. This will be controlled via either keyboard keys or mouse interaction with the traverse controls, both from the gunner’s position. The tank’s gun must be able to be elevated and depressed upwards and downward to a maximum of 23 degrees and a minimum of -7 degrees. This must be controlled by a crank, either controlled via keyboard keys or mouse interacton with the traverse controls.
* The gun must be able to be loaded with ammunition from the tank’s ready rack. This will be accomplished by clicking on the shell with the mouse cursor and moving it to the breech of the cannon whilst the player is occupying the loader’s position. The player must also be able to fire the gun. This will be accomplished by a left mouse button click or keyboard key. This action will only be possible when occupying the gunner’s seat.
* The player must be able to cycle between the commander’s, gunner’s, loader’s and driver’s position in the vehicle. Each position will have a complement of systems assisting the player in their task in the vehicle.
* The commander must be able to open his hatch and look outside the vehicle. He must also be able to use a commander’s periscope.
* The gunner must be able to look down his primary gunsight, as well as a backup lower-magnification gunsight.
* The loader must be able to turn out of his hatch and look around.
* The driver must be able to look out of his periscope.
* The vehicle must also have an array of systems available to inform the crew as to what their tank is doing.
* The driver must have an RPM meter, speedometer, and gear indicator.
* The gunner must have a turret/hull azimuth indicator.
* The commander must have a turret/hull azimuth indicator.
* The loader must have an indicator on the gunbreech to indicate that the gun is loaded.
* As the game will not have multiplayer, the player must be able to command AI crewmembers who will occupy the positions that the player does not. They must be able to complete all tasks that the player would be able to. The player must be able to issue commands to all crew members from any crew positions in the tank. The player will do this with a communication rose, brought up by a keyboard key, from which the player will be able to issue commands.
* The AI crewmembers must also supply the commander with information that they gather, such as contact reports on enemy vehicles, and reporting damage to the vehicle. This will be done by text popups.
* The game must be able to simulate the ballistics of armour piercing and high explosive tank rounds.
* The rounds also must have realistic effects on armour, with armour thickness, impact angle relative to the armour and penetration of the round taken into account.
* Any penetrating hit must be able to damage key components of the vehicle if they hit in the correct place, such as the engine, gearbox and gun breech.
* The game must have a small 3x3km map, with some props, for the vehicle to move around in.
* The game will also use this map for a basic mission. The player will have to find and destroy an enemy truck.
* Enemy tanks will be protecting this building. They will use a different model and have simple AI, armour thickness modelling, and module placement, but they will not be drivable.
* The game must also have a menu system ready. It will be displayed when the game is first loaded, it must allow the user to select the mission, and when escape is pressed ingame, the player must be brought back to the menu.
* The player will not be able to rebind keys.

My game will use the Panda3D engine for its graphics. Coding will be done in Python, with the IDE being used being Microsoft Visual Studio 2015. The requirements for hardware will be based off the minimum requirements for Panda3D. Additional hardware needed will be the memory space necessary to store the game, a keyboard, and a mouse. My game should be able to run in its own executable file, so it will not need any additional software.

Minimum specifications:

CPU: dual-core 2.4ghz

RAM: 4gb

GPU: Nvidia geforce 460

Memory Space: 512mb

OS: Windows 7

Peripherals: Mouse and Keyboard

Design

Data Dictionary

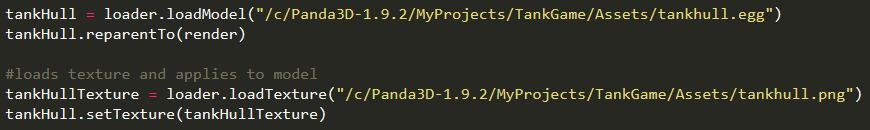
Do commenting

Do example algorithm

Development

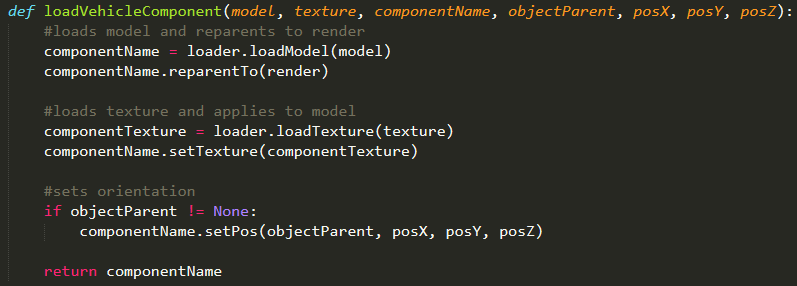
I broke the software development stage into four parts: creating the initial version of the feature in question, finding any bugs that existed in the initial version, fixing those bugs to perfect the feature, and then giving that version to my end users to provide feedback on the specific feature in question that would then go into refining the feature. This cycle would then repeat until the end users were happy with the feature.

The first feature I had to develop was the system used to initialise the tank and the tank’s interior. Originally this was done by using Panda3d’s inbuilt rendering commands inside the main program itself, as shown below:



However, after a few models, this would have swollen the size of the program significantly, especially when my single vehicle is made up of 20+ components. So, I decided to create a function inside my external utility library that would allow me to just supply a function with all the necessary information to create the component.

Shown below is the function inside the utility library:

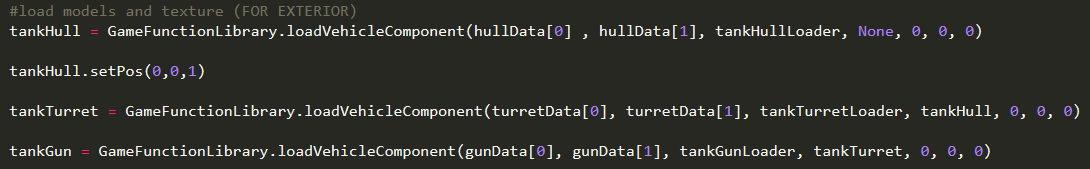


The function takes the model filepath of the component, the texture filepath of the component, the name of the component (this is just used for external referencing), the model that the component’s position will be set relative to, and the X, Y and Z coordinates of the component relative to the parent object. The function loads the model and parents it to panda3d’s render node, bringing it ingame. The function then loads the texture and applies it to the model. After that, it checks if the component has a parent object, and if that evaluates to anything other than true, then the model is positioned relative to the object

This function is used inside the main body of the program to initialise all the components that the tank the player is in uses. The model and texture filepaths are held inside arrays for each component, shown below:



And the function from the external utility library is shown below:



(this is not all the functions

Bibliography

1. Screen capture from https://www.youtube.com/watch?v=XNyWW7p62U8
2. Found at http://www.securitronlinux.com/arma-3-tips-and-information-for-new-players-of-this-war-simulation-game/
3. Found at https://forum.warthunder.com/index.php?/topic/187188-143755/
4. Found at http://forum.americasarmy.pl/showthread.php?t=15950