MIDDLE EAST TECHNICAL UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

REQUIREMENT ANALYSIS REPORT

EDU3D

AlturaTech

TABLE OF CONTENTS

1. PROJECT DEFINITION, SCOPE AND GOALS

1.1. Project Definition

AlturaTech's LETS (license education and traffic simulation) project aims to develop a tool for driver courses. It will include a driving simulation, a scenario editor and a test for students. System will have two different user types: teacher and student. Student will be able to use simulation part that will teach him\her real traffic environment. This part will be developed with graphical, physical, audio and artificial intelligence support. Teacher will be able to use scenario editor and map editor part to arrange a test for the student and measure his\her knowledge and performance on traffic.

1.2. Project Scope and Goals

AlturaTech aims to develop a system that will collect all beneficial parts of current traffic simulations and driver education tools. In addition to these parts, we will add some parts that are needed and never used before. Project scope:

- LETS will use real physics engines that will be used for auto movements, weather conditions and accidents. While a student is using simulation he\she must be aware of real physical laws are still valid for the system.
- LETS will use advanced graphic engines that will allow us to render our visual parts realistically and enjoyable. Our system will be much better than most current systems in 3D graphics. We hope it will be like Need for Speed Underground rather than moving boxes.
- LETS will use audio support for the realism of the simulation. It will be both noise of the traffic and entertaining ones. For the proper use of audios we will use some programs that will be like audio engines.
- LETS will be supported with an advanced artificial intelligence support. It will be most specific part of our system that adds a lot for usability and realism of the system. In the system pedestrians, other drivers and environment will be very close to natural ones. We will work over and implement artificial intelligence and use some type of engine for integration.

Main goals of the LETS are:

- Education: System should be educative for students and teachers could be able to measure and test them.
- Usability: System should be usable for all type of users and it should have shortcuts and easy links.
- Realism: System should be realistic with the help of supports mentioned.
- Performance: System should be fast and run accurate enough although it is a complex one.
- Interface: System should have interfaces that are user friendly and enjoyable. Colors and other main opponents will be chosen accordingly.
- Network: LETS will be support many users in same time so that the teacher control many students on same time on same exam.

2. THE PROCESS

2.1. Team Organization

In software engineering process, team structure has a vital importance. Since we are all relaxed and easy-going persons, we choose a team leader for coordination and interaction between our group members. This structure exactly fits into Democratic Centralized (DC) model. Decisions are taken by all project members and final approval is given by team leader.

2.2. Process Model

While we are choosing our project team organization, we planned to move in a direction called Component-Based Development which is spiral model variation in which applications are built from prepackaged software components called classes. Object oriented approach is a must in developing our project so we are focusing on creating components and packages.

2.3. Major Constraints

2.3.1. Project Schedule

We have to finish the project until end of May 2006 so this is one of the main major constraints. We have to design, build and test the project during this period.

2.3.2. Language constraints

We are going to implement our project in C++.

2.3.3. Data constraints

Models, textures, sound, videos will require huge amount of storage and processing.

2.3.4. Execution Speed

We should provide at least 24 frame per second rendering speed in order the game to be smoothly playable.

2.3.5. User Interface

User friendly menus will be created. Easy to use and simple menus will be taken into consideration.

3. LITERATURE SURVEY

3.1. Current Popular Systems and their Features Search

3.1.1. Driving and Car Simulation Softwares

A) STSoftware

STSoftware (Simulation Technology Software) is a company which develops traffic simulators. We have found its web site while we search on google.

www.stsoftware.nl

Now let us investigate their program.
(There are videos in http://www.stsoftware.nl/demos.html)

Graphics

Its graphics are unsophisticated, but not like traveling boxes around. Cars are fine, but pedestrians are out of intelligence concept. There are several mistakes like one can pass his/her own car through a tree or pedestrian, you will not have a crash.

Artificial Intelligence

Artificial Intelligence is very low, cars travel on their way, and pedestrians don't walk through road on red lamp and so on. One can easily drive car in this simulation.

Usability

Left and right signals are very well integrated and the other cars see them and behave through it. According to videos we can't easily say that its physics motor is bad or not, because there were no clue about rainy roads or other difficult situations on videos.

System Analysis

Developer Definition of the system:

ST-RoadDesign

Graphical road databases can now be created in a minimum amount of time with the ST-RoadDesign package. This designer allows the user to build databases in a few hours, or if large databases are required, in a few days time. The designer generates both graphical and 'logical' databases for the intelligent traffic in ST-Traffic. Driving simulations can be run with ST-Traffic and visualized directly in ST-Render, using these databases.

ST-Scenario

The driving simulations are generated with the scripting language ST-Scenario, developed by ST Software, for which a handy script tool is included. This scripting language is very versatile, and easy to learn for people with a technical background. The scripts results in fast runtime simulations with high frame rates in ST-Traffic, even when more than 100 intelligent agents (vehicles or bicyclists) are simulated simultaneously. Script may even be used to add as many external computers as desired, for special tasks that communicate with the simulator.

ST-Control

With the graphical user interface ST-Control the system can be configured and as many renderers added as required. It is possible to add external vehicle models or use an internal vehicle model. Users

may add their own vehicle cabin or control the driving simulator with a game set. This allows for full scalability: the driving simulators can range from low to high end. In addition, of course, ST-Control lets the user control the simulator. See Appendix E1

ST-Render

The rendering package, ST-Render, is a configurable real-time renderer in which viewing angles, viewing positions and viewports can be defined. It is easy to add as many renderers as required. Systems may then range from simple 1-channel simulators with only the road view ahead on a single monitor, to a full 360° field of view with multiple channels and projection screens. See Appendix E2

ST-CentralControl

If the user wants to control a number of simulators simultaneously with one operator, ST-CentralControl is the choice to make. This package offers common control, print and database facilities for up to six simulators. This greatly increases the cost-effectiveness of your driver training courses. In addition, this package contains a sophisticated driver assessment system which, together with a set of scripts that come with this package, can be configured and changed by the user. The user may add driver tasks that need to be monitored or change the criteria or driver feedback for driver errors easily. A large number of driver tasks are presently included in the driver assessment system.

B) Hank Driving Simulation

Hank is a University of Iowa project which investigates through driving behaviors of children. It is web page is:

http://www.cs.uiowa.edu/~hank/

We have not found so much information about Hank because there are no videos and demos available on internet. But they use projections to simulate the world and it was really interesting.

C) Meteksan – Trafikent

Graphics

- The general car modeling is above average, but there is no damage modeling.
- The world detail is not enough to create a realistic look.
- The frame rate is high enough for a game.
- It can simulate day or night at a sunny, foggy or raining day.
- The map is small for a simulation like this. The university campus can be bigger. See Appendix E3

AI

- The artificial intelligence consists of driving forward. We have tried some tests to understand the artificial intelligence, for example; when we stopped in the right lane with 2 cars behind us, none of them gave any response to us; they just stopped there for 2 minutes when we gave up stopping. There is nearly no lane changing in the computer controlled cars.
- There is a bar simulating the heat of the motor. It reminds users to change to the next gear and prevent overheating.

Usability

- The general simulation of the car is average, the turnings and the acceleration seams realistic.
- They have implemented a 3-pedal system with a clutch, but the feedbacks from the clutch are nowhere close to the real life. The motor did not stop even if you stop in the first gear.
- Users can choose the starting location in the map
- They have implemented the wind wipers.
- It replays the traffic accidents in order to see the mistakes and provide better training.

System Analysis

- Meteksan says that this is the most realistic traffic simulator in Turkey.
- There is not a Turkish alternative for Meteksan; the foreign alternatives are more expensive.
- The camera turns a little to left and right with the turning of the wheel, which gives the user a better view while creating a sense of reality.

3.1.2. Car Games

A) Midtown Madness

Graphics

In midtown madness we see that graphics are not so bad looking to other games. In car camera shows us that indoor design of the cars are done very carefully. (Image 1) But we see a lack of shadows and reflection of lights on the outside, like on the police car. Also the damages on the car are very unrealistic. We have cracks on the glasses after a huge crush. (Image 2) The outdoor elements like buildings and trees are good looking. (Image 1)

Artificial Intelligence

When we look at artificial intelligence of other cars in midtown madness we see that only the police cars have some extra movements. Other cars are always going in a low speed, stopping at red light or when a car is stopping in front of them and moving at green light. In addition to these cars police cars are moving faster when they are trying to chase another car. Also these cars have poor a.i. because they only go with you and they try to stop in front of you. It is very easy to escape from them. So we can say that there is not traffic

like real in the game, and it can not be an example to a simulation because cars are not intelligent.

Usability

When playing the game I controlled the car via keyboard and I saw that steering the wheel is made by holding on right or left button. This is better than some games which do full steering when tapping to the right or left button. But when we drive much we see that turning on a direction is not made like real. Although you are going very fast you can steer left easily and nothing happens like drifting. You can turn very rapidly. Reality factor is not very good for tires and drifting.

Car Physics

When we look at a car editor for the midtown madness we see that game developers have variables like these:

-Engine
- horse power
- idle rpm
- max rpm
- optimum rpm

-Transmission
- number of gears
- reverse gear ratio
- low gear ratio
- high gear ratio

-Brakes:
- Front brakes
- Front hand brakes
- Rear brakes
- Rear hand brakes

-Body:

-Mass
- Med damage
- max damage
- Impact Threshold

-Handling
- Front static friction
- Front sliding friction
- Rear static friction
- Rear sliding friction See Appendix E4

B) Need For Speed - Underground 2

Graphics

- Need For Speed series has always been in the bleeding edge of the graphical technologies.
- It supports all the new features of DirectX, and it has particle system, light trails, road reflection, car reflection, over bright, enhanced contrast, rain

splatter and texture filtering display options in order to customize the gameplay.
- The map of the game is massive, with lots of details in the cities.
- It has more then thirty realistic car models for the user to drive.
- The car models can be modified or upgraded, which creates nearly billions of car combinations available to gamers.
- The vision is blurred with the increasing speed, to create a more realistic driving.
- The game has a daytime which circulates every 48 minutes, which will enable gamers to race at dusk or dawn.
- The neon and shadow effects are very realistic, and changes between day and night. See Appendix E5

AI
- Computer controlled cars has two types;
- Other racers, which tries to go between two points in the maximum speed
- Normal city cars, wander of the streets aimlessly, they can switch lanes, and can make turn in the intersections.

Usability
- It can be played from both the keyboard and a joystick (including steering wheels).
- The physics of the car is fabulous, they have also implemented the differences between front wheel drives and rear wheel drives.
- The grip and the handling of the car changes with the weather conditions of car racing game. See Appendix E6

C) Grand Theft Auto San Andreas

Artificial Intelligence

When we look at artificial intelligence of the San Andreas we see a new thing in the traffic: The cars react to your acts. For example when you crush to another car. Sometimes they try to escape from the event location whatever the lights or the police. Sometimes when you crush to a taxi driver they start to go faster to catch you and they crush you many times and they push you through the wall. Also in the highways the cars don't like to go slowly. So they horn and shout to you. Also the cars don't go until there's an obstacle in front of them. When they see another car coming in a junction they try not to crush to other cars.

Usability

I tried to test the game with keyboard so I don't know much about steering with a console but with the keyboard you can control easily the car. For turnings with speed I saw that when you go faster you can't turn with the angle that you turn slowly. If you try to do so much you start to spin at the road and like the real cars you can stop hardly. Also when it rains the control of the car becomes much difficult and you can not go very fast until you are on the highway.

Car Physics

I tried to find an editor for San Andreas and find out one for all the cars. When we look at this software we see the following options for the cars:

-Mass
    -Mass
    -Turn Mass
    -Drag
-Center of Mass
    -X
    -Y
    -Z
-Traction
    -Multiplier
    -Loss
    -Bias
-Brakes
    -Deceleration
    -Bias
    -Abs
-Transmission
    -Number of gears
    -Max. Velocity
    -Engine Acceleration
    -Engine Inertia
    -Drive Type
    -Engine Type
-Suspension
    -Force Level
    -Damping Level
    -High Speed Damping
    -Upper Limit
    -Lower Limit
    -Front and Rear Bias
    -Anti Drive Multiplier

See Appendix E7

When we look at these properties we see that San Andreas has better reality in cars physics. I think that they have enough properties also as a car simulation software. So we can take these variables as an example to our software, so that we can simulate variety of cars easily instead of one type of car.

Graphics

- GTA: San Andreas has very good graphics, but the most interesting part of GTA is the size and the level of detail of the map. The map includes 3 big cities, and countryside between them. It can take a few minutes to travel between two cities.
- The graphics are not superb but considering the number of models to be drawn, it is quite good.
- Although it is not a driving game, the car models are very good and realistic.
- The game supports light effects and shadows, but with an average system it creates frame losses.

See Appendix E8

3.1.3. Map and Scenario Editor Example

Although I searched much the internet I couldn't find any suitable editors for car simulation software. Some of examples are very difficult to use (for coders) and some of them are not suitable for road mapping, so I decided to explain and show sim city classic as an example editor because it is an easy editor to provide, and it is a 2d map editor so that user is not problems with 3d buildings textures or their heights. Only roads and cars are important to us and we can edit them easily in this editor.

A) SimCity Classic

When we look at SimCity classic game we see that we have a menu on the left. This menu is not fixed so we can drag it to elsewhere. In this menu we see a bulldozer, road, railroad, buildings and some other buildings which don't interest us. To build a road we must select the road button and we click somewhere on the map. The map is formed of squares so the roads are always on a line or they are turning left or right. To place a building also we use same menu.

The advantage of this editor is it's simple to use but it has some disadventages too. For example the roads are not always on a line. So we have to do hexagonal mappings. Also these maps don't have height, they are flat but the world is not always flat. Also we must add some traffic elements like traffic lights, traffic signs and also type of roads (highways, one way roads, etc.)

B) GTA 3 Map Editor

As we mentioned in 2.1, STSoftware has a map editor and we talked about it. While on the search we come up several map editors on a popular computer car game, GTA 3.

http://gtaworld.nfscheats.com/utilities.php

The link above consists of nearly all tools to control and produce new maps and scenarios on GTA 3. (GTA3 Map Editor 3D, GTA3 Admin Console, Garage Edit, DFF Viewer, Collision File Generator, Car Stat Editor, Audio Editor, GTA3 Res Hack, GXT Editor, IMG Editor, MagicBoots, Mod Installer, Mod Manager, Parked Car Editor, Weapon Stat Editor, ZModeler)

http://www.estetiksoft.de/gta3console/collisioneditor.htm

The link above is a 3D map editor on GTA 3.

C) Scenario Editor

As we want to add a scenario editor to our project, we searched for similar scenario editor supported systems. By scenario editor support

AlturaTech aims to give opportunity to students and teachers that they can arrange the roads and streets, weather conditions, artificial intelligence and difficulty levels of the simulation. By that way tests would be more appropriate and more specific.

Warcraft III – World Editor

Usability

- It enables the users to create scenarios to be played in the game.
- It has a simple graphical interface that can modify or create maps.
- Putting units or buildings are simply done by selecting the type, then clicking the place to be put.
- Each unit can have a number of checkpoints and a stance (aggressive, defensive, no attack).
- User can arrange units to be spawned at a later stage of the scenario.

Artificial Intelligence

- The AI of particular units is defined by the stance and the order given. Units will try to visit each of their checkpoints, if an enemy is encountered while walking it will react according to its stance.
- The control of the units in general is done by scripting.
- It has a building order for each race and tactic.

3.2. Meeting with Driving Teachers

We are developing a system that will work as a driving teacher of driver's license courses. In order to be appropriate and realistic we met with three driving teachers (our potential users):

- Ali MERAKİ 3159690
- Bünyamin SANLAV 3253141
- Mahmut BOZ 05436618570

Meeting was held on 27.10.2005 with those minimum 15 year experienced driving teachers. They told us about their main responsibilities and important point of driver education. They also answered our some specific questions which enlightened us about the system. You can find the notes related to the results of the meeting in the Appendix, part F.

3.3. Literature Survey and Technical Analysis

3.3.1. RenderWare Products

In our project, if we use RenderWare, the compilation of different libraries will be no longer such a bad disease. Graphics, Physics, AI and Audio will have been put together without giving too much effort.

Some background information about renderware:

Criterion Software's main product portfolio is RenderWare®, the world's leading middleware technology for the games industry.

The RenderWare portfolio of game development solutions includes:

- RenderWare Platform, the game development middleware of choice, that consists of an integrated technology suite providing world leading, Graphics, Physics, AI and Audio.
- RenderWare Studio, the game development framework that harnesses the power of the development team from prototype to gold master.

RenderWare is empowering developers to be creative within a mature, professional software engineering environment. There are over 500 current generation RenderWare games in development or published world-wide. Today 1 in 4 console titles in pre-production or development is using RenderWare technology.

RenderWare is used extensively among the game development community, including clients such as Activision, Atari, EA, Konami, Midway, Namco, Rockstar Games, Sammy Studios, Sony Computer Entertainment, Sony Online, THQ and Ubisoft.

3.3.2. Other Libraries and Engines

Physics Resources

Open Source

- Open Dynamics Engine (ODE) and odejava

  It is platform independent with an easy to use C/C++ API. In our perspective, it is useful for simulating vehicles.

  http://www.ode.org/
- Tokamak Game Physics SDK
- Newton Game Dynamics
- Bullet Continuous Collision Detection and Physics Library
- OPCODE Optimized Collision Detection

Commercial

- Havok Game Dynamics SDK
- RenderWare Physics
- Meqon Game Dynamics SDK
- 3Impact dynamics-capable game engine
- CMLabs (real time, but oriented toward engineering and academic use)
- NovodeX Rocket (free for non-commercial use)

AI Engines

Open Source

- OpenAI

  It is still an under development. The project's primary goal is to create configuration and communication standards for AI tools.

  http://openai.sourceforge.net/
- FEAR

A generic framework for AI development with a collection of building blocks, and methodologies to extend the system

http://fear.sourceforge.net/

- OpenSteer

A C++ library and tools to help construct steering behaviors for autonomous characters. http://opensteer.sourceforge.net/

Commercial

- AI. implant from BioGraphic Technologies
- Massive (crowd system used in Lord of the Rings, see also)
- Softimage Behavior
- RenderWare A.I. ("powered by Kynogon")
- SimBionic from Stottler Henke Associates see also.
- Pariveda
- Louder Than A Bomb! Software
- Virtools AI Pack (formerly NeMo?)
- DirectIA
- emotion ai
- Realtime Drama

Sound Libraries

Open Source

- OpenAL

The main advantage of OpenAL over DirectSound is that OpenAL is designed to be a cross platform API. It is possible to use Windows, Mac and Linux binaries of it so it can be used on many Operating Systems.

- OpenAL++

OpenAL++ is an object oriented API for spatial sound, built on top of OpenAL. It also uses OpenThreads and PortAudio to give easy to use, portable support using input devices. By using smart pointers, memory allocation problems are held down to a minimum development.

http://www.openal.org/

Input Libraries

SDL Simple Direct Media Layer

Simple DirectMedia Layer is a cross-platform multimedia library designed to provide low level access to audio, keyboard, mouse, joystick, 3D hardware via OpenGL, and 2D video frame buffer

http://www.libsdl.org/index.php

3.3.3. Integration of Libraries

Game Engines

Open Source

- Delta3D

Delta3D is an Open Source engine which can be used for games, simulations, or other graphical applications. Its modular design integrates other well-known Open Source projects such as Open Scene Graph(Written

entirely in Standard C++ and OpenGL it runs on all Windows platforms, OSX, GNU/Linux, IRIX, Solaris and FreeBSD operating systems.), Open Dynamics Engine, and OpenAL. Rather than bury the underlying modules, Delta3D just integrates them together in an easy-to-use API.

Delta3D renders using OpenGL and imports a whole list of diverse file formats (.flt, .3ds, .obj, etc.).

Delta3D can be compiled in Microsoft Visual Studio .NET (7.1) and Fedora Core 4 using gcc 4.0.0.

http://www.delta3d.org/

4. PROJECT REQUIREMENTS

4.1. Functional Requirements

4.1.1. Menu Requirements

Login Screen will come before entering our system so that a teacher or student can be distinguished. Furthermore, teacher's main menu will include a scenario tool in addition to students' main menu.

a) Student Menu Requirements

- ***Edit Profile***
  It will have student's personal information such as name, surname, address, telephone number.

- ***Set Options***
  Students can change their preferences like audio, video.

- ***Select Vehicle***
  Features of cars can be displayed and student can select a color for the car.

- ***Exit***
  Student will return to the operating system.

- ***Select Scenario***
  Scenarios already created by teachers and put into database can be selected by student.

- ***Read Help***
  Read Help item will have Traffic rules, frequently asked questions, a tutorial about our software usage.

- ***Enter Training***
  Student will enter either a default scenario or a selected scenario.

- ***Logout***
  Student will return to login screen.

b) Teacher Menu Requirements

This part includes the menu requirements of teacher type users. We arranged this topic in 4 main parts.

Scenario Editor:

- New/Open Scenario; lets teacher to create a new scenario or modify an existing scenario using the pool of addable objects and traffic elements, also they can arrange artificial intelligence and difficulty level of that scenario.
- Constructor Tool: This tool will help adjusting the landscape, like inclination of the map, adding water elements like sea, lake ...
- Brush Handler: There will be a toolbox, where teacher can select the object type he/she wants to add to the map such as roads, buildings, trees, traffic elements and cars.
- Object Properties Window: Each moving object in the scenario will have a properties window, where teacher can select general behavior of the car and set checkpoints for the car to follow.
- Help: This selection will aid the teacher on how to use the scenario editor, giving tips and showing key shortcuts.

Option Setting:

- Set Weather Conditions: The teacher can set and adjust the weather conditions at where the student will be driving. This option can be changed during simulation.
- Car Settings: There will be an option where the teacher can modify the settings of each individual car. This option will include front/rear wheel driving, tire friction, etc …
- General Test Options: The teacher has the privilege to set the general settings of the driving course, like selecting the scenario, setting the duration, starting location, day and night settings ...

- **Student Tracking:**
    - Reading Error Reports: The teacher will have an easy to use environment to keep track of the little and big traffic mistakes of the student while driving.
    - Reading Crash Reports: The simulation is supposed to create a crash report including why the crash happened, where the mistake was made, and how to avoid it.
    - General Student Data: Each student will have his/her own record of previous tests. The teacher will be able to see the overall performance of the student, where he/she can see the student's mistakes and warn him accordingly.

- **Manage Student Account:**
    - Add New Student Account: Teacher will be able to record general information of the student on our system, enabling the student to login.

- Delete a Student Account: Teacher will be able to delete a student account, disabling the student to login.

4.1.2. Traffic Flow Requirements

Student uses this simulation in a traffic flow. Student will try to drive a car in a world of cars, trees, buildings, atmosphere and earth surface like real life.

There are 4 main parts of requirements in traffic flow.

a) Environmental Requirements

Buildings

- Static Buildings. No way to enter them.
- They will have different kinds of textures and height.

Cars

- Dynamic cars moving using car physics.
- Cars behave according to climate changes.
- Cars will have artificial intelligence which we will discuss in artificial requirements part.
- Cars will have lamps (brake, left signal, right signal)
- Cars will have weight, speed and sort of characteristics.
- Cars include trucks, light and heavy weighted cars, jeeps etc.

Trees

- Trees are static.
- Trees have different size, texture.

.

Signs

- Traffic signs will change according to scenarios and maps.
- Traffic signs will be available according to scenarios and maps. Student has to obey the signs.

Traffic Lamps

- Traffic lamps will change according to scenarios and maps.
- Traffic signs will be available according to scenarios and maps. Student has to obey the signs.

Climate Changes

- Morning, evening, night will be available for scenarios and maps. Sunlight will change according to time of a day.
- Snowy, rainy, sunny, cloudy climates will be available and objects (cars, road...) will behave according to climate.
- Teacher can change climate.

b) Living Simulation Requirements

Mistakes

It records statistics about mistakes.

Control

It is going to control mistakes real time.

Report

It is going to report student's mistakes to teacher.

Student Tracking

It is going to track student's speed, mistakes, overall situation and acquaint teacher.

c) Artificial Intelligence Requirements

Smart Cars

- Each car will have some kind of intelligence.
- Each car will use cars properties (signals, brake etc…).
- Each car have a characteristic (aggressive, calm) and these can be set overall (effects all cars).

Randomness

- Cars behave randomly.
- Climate can change randomly or can be set by teacher separately.
- Teacher can adjust randomness factor.

Flow of Traffic

- Similar to real traffic.
- Flow of traffic can be controlled (like start and end points).
- Randomness has a great effect on flow of traffic.

d) Student – Car Interaction Requirements

Scenarios

The simulation behaves according to a scenario. So student must obey the rules (set by teacher), signs, traffic lamps and conventional.

Hardware

The student uses hardware equipments to interact with the simulation. Detailed information can be found in 4.4 Hardware Requirements.

4.1.3. Operational and Structural Requirements

a) Graphics Engine

This engine will render 3D objects such as cars, trees and roads with textures.

In addition, it will coordinate with other engines while running.

b) Audio Engine

Audio engine will let player control sound options such as volume level. Sound effects will be played in appropriate times during traffic flow. In addition, some audial instructions from teacher will be created.

c) AI Engine

Our system will have an advanced artificial intelligence and scripting engine. This will be organized by teacher. AI engine will coordinate all cars and their drivers' aggression level. Also, AI engine will affect traffic flow.

d) Physics Engine

Physics engine will display real life object behavior. Car physics and environment physics such as weather, road and wheel drive can be arranged by teacher.

e) Data Module

Our system will have a database control module that will interact with other engines. Distribution and collection of data will be held by this module.

f) Main Engine

It provides interaction and coordination between each module and engine.

4.2. Non-Functional Requirements

4.2.1. Usability

In developing a simulation, one of the most important issues to be considered is similarity to reality, so that the user can use real cars easily after training with our software. The car will respond to user's movements with realistic feedback. Also other cars behaviors in traffic must be similar to the real life. The controls in software simple and identical to real car like wheel, three pedals and gear. Also the outputs for the user like speed, revolutions per minute will be shown with huds like in real cars.

4.2.2. Reliability

Our software must be free of bugs and errors because the area we are working in is traffic which can be lethal in wrong use.

4.2.3. Portability

Most of car simulation softwares are sold with their hardwares but these are not portable and can not be used in homes. Our software can be installed and used with any personal computer.

4.2.4. Education

Since our main aim is educating the new drivers, our software must contain required functions to create an easy platform to use for education by teachers and students.

4.3. Software Requirements

User Requirements:

- Windows

Development Requirements:

- Microsoft Visual .NET 2003
- Windows
- OpenGL(Open Graphics Library)
- RenderWare Platform
    - RenderWare Graphics

    - RenderWare Physics
    - RenderWare AI
    - RenderWare Audio
- ODE(Open Dynamics Engine)

4.4. Hardware Requirements

We plan to purchase steering wheel, but according to our searches through shops and internet, we could not find a steering wheel which includes clutch and 5-way gear box. As a result we plan to build them if possible.

We have met with Asst. Prof. Buğra Koku from Mechanical Engineering Department. After the interview, we found different ways how to build the clutch and gear box. Some of them are our scope and beyond the education we have had so far.

4.4.1. Clutch

The best and possible way is to get a mechanical clutch from junkyard. After maintaining the clutch, we plan to build its electronic parts from a game pad. At first we are going to learn the structure and input/output of the game pad. Later on, we are going to implement necessary mechanical parts between the clutch and game pad.

We are going to put a vibration device (from a mobile phone or a game pad) under the clutch, so according to a specific input, clutch may react to user by vibrating.

4.4.2. Gear Box

We are going to try to construct our gear box. We have learned that building a gear box requires more than building a clutch and hard to implement.

One possible way to build a gear box is to maintain a mechanical gear box from the junkyard and merge it with a joystick's electronic part.

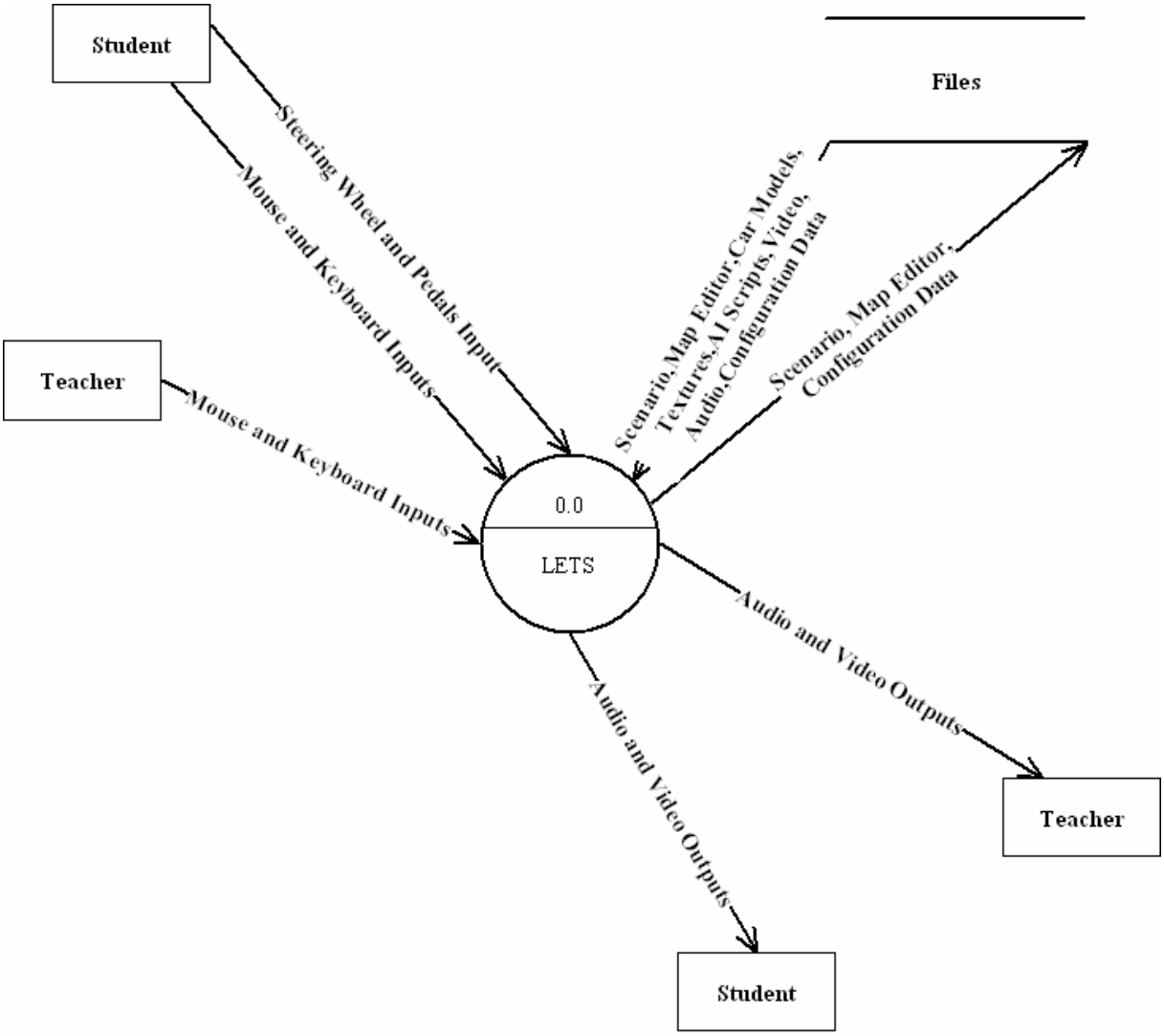
4.4.3. Drivers of the Clutch and Gear Box

Since we plan to build clutch and gear box from game pads and joysticks, drivers of game pads and joysticks will be used. So we will not have to implement special drivers.

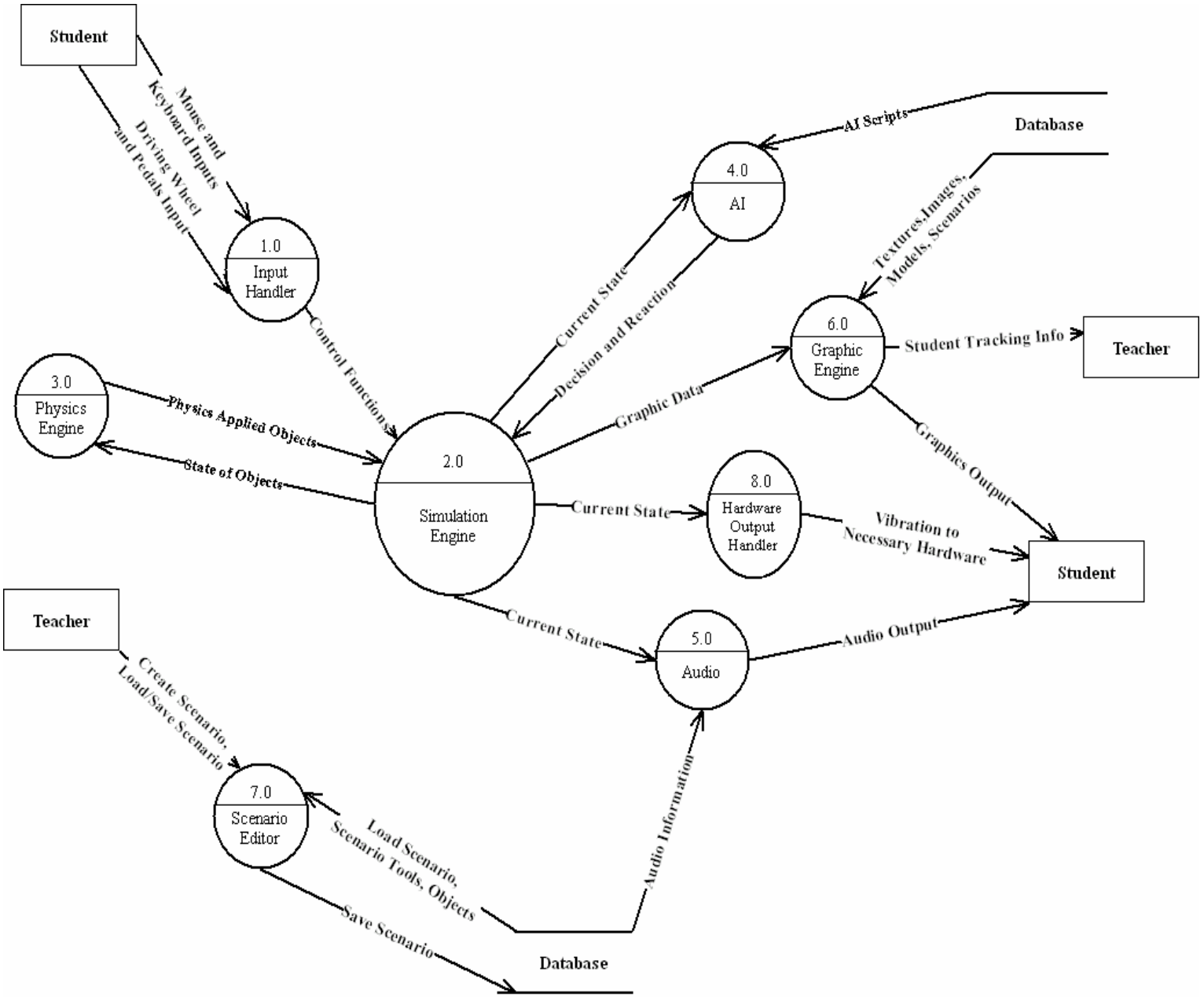
5. STRUCTURED ANALYSIS - FUNCTIONAL MODEL

5.1. DFD

5.1.1. Level 0 of DFD

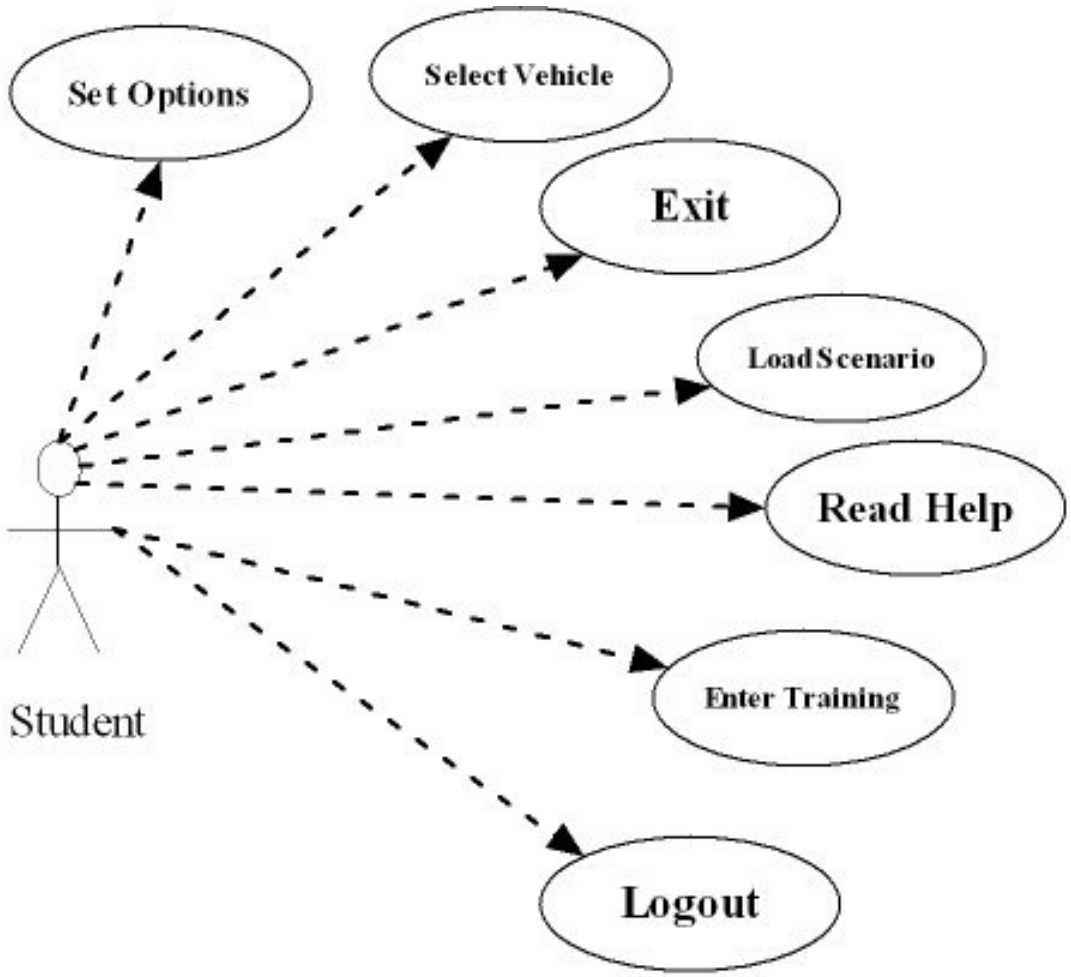


5.1.2. Level 1 of DFD

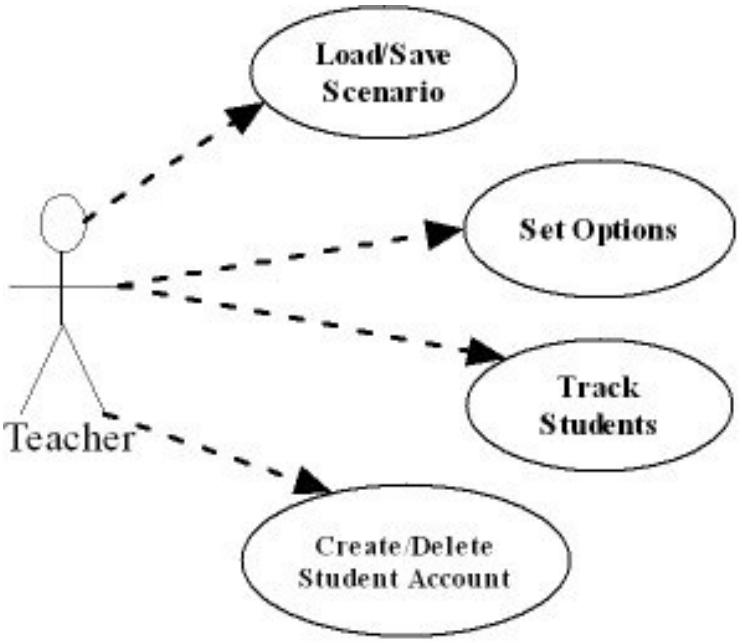

Student
Mouse and Keyboard Inputs
Driving Wheel and Pedals Input
1.0
Input Handler
Control Functions
3.0
Physics Engine
Physics Applied Objects
State of Objects
2.0
Simulation Engine
Current State
Decision and Reaction
4.0
AI
AI Scripts
Database
Graphic Data
6.0
Graphic Engine
Textures,Images, Models, Scenarios
Student Tracking Info
Teacher
Graphics Output
Current State
8.0
Hardware Output Handler
Vibration to Necessary Hardware
Student
Current State
5.0
Audio
Audio Output
Teacher
Create Scenario, Load/Save Scenario
7.0
Scenario Editor
Load Scenario, Scenario Tools, Objects
Audio Information
Save Scenario
Database

5.2. Use Case Analysis

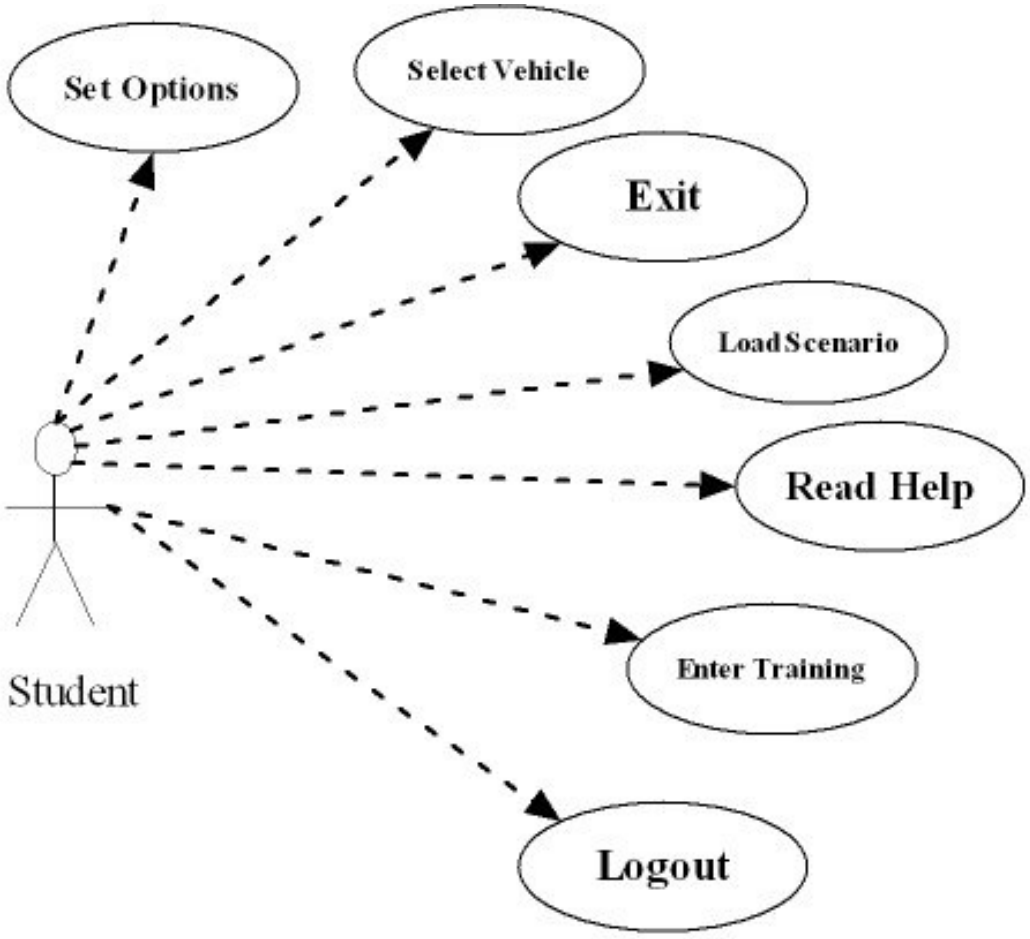
5.2.1. Menu Student Use Cases



5.2.2. Menu Teacher Use Case



5.2.3. Car Use Cases



6. RISK MANAGEMENT PLAN

It's wise to consider possible risks before they happen, because these risks generally cause in project failure or late submission. In order to prevent or decrease these risks effects we must do a risk management plan for our software project.

6.1. Project Risks

We have different subsets for risks which are sorted in priority:

Lack of Authority: Project team leaders authority can be decreased by some critical decisions. This may result in lack of organization in team, resulting in latency in some due dates in project.

Lack of Responsibilities: Project team members must obey to the project rules and their due dates. It has a vital importance in our progress.

Overloading team members: Project team members can have many tasks because of their homeworks and other projects. Team leader can assign some tasks to them not considering their other works.

Postponing tasks: Project team members are used to do their tasks at the last minute. This can be a problem in project schedule.

Hardware constraints: We consider designing hardware to use clutch and gear in simulation. But we are unsure about the feasibility of these hardware components and their drivers.

AI Implementation difficulties: We are planning on a realistic artificial intelligence for other cars in traffic but, this can be more difficult than we thought to be.

Graphical Implementation difficulties: Although we are considering making environment realistic, we are not experts on computer graphics subject and this can cause a step down on the predicted software graphics.

Compatibility of libraries: We are planning to use many libraries in order to create a realistic simulation. But we don't have experience in coordinating those libraries. So this may take longer time that we are expecting.

6.2. Risk Table

Risks	Probability	Impact
Lack of Authority	25%	1
Lack of Responsibilities	30%	2
Overloading team members	40%	2
Postponing tasks	30%	2
Hardware constraints	70%	3
AI Implementation difficulties	40%	3
Graphical Implementation difficulties	20%	3
Compatibility of libraries	10%	4

Impact Values:
1. Negligible
2. Marginal
3. Critical

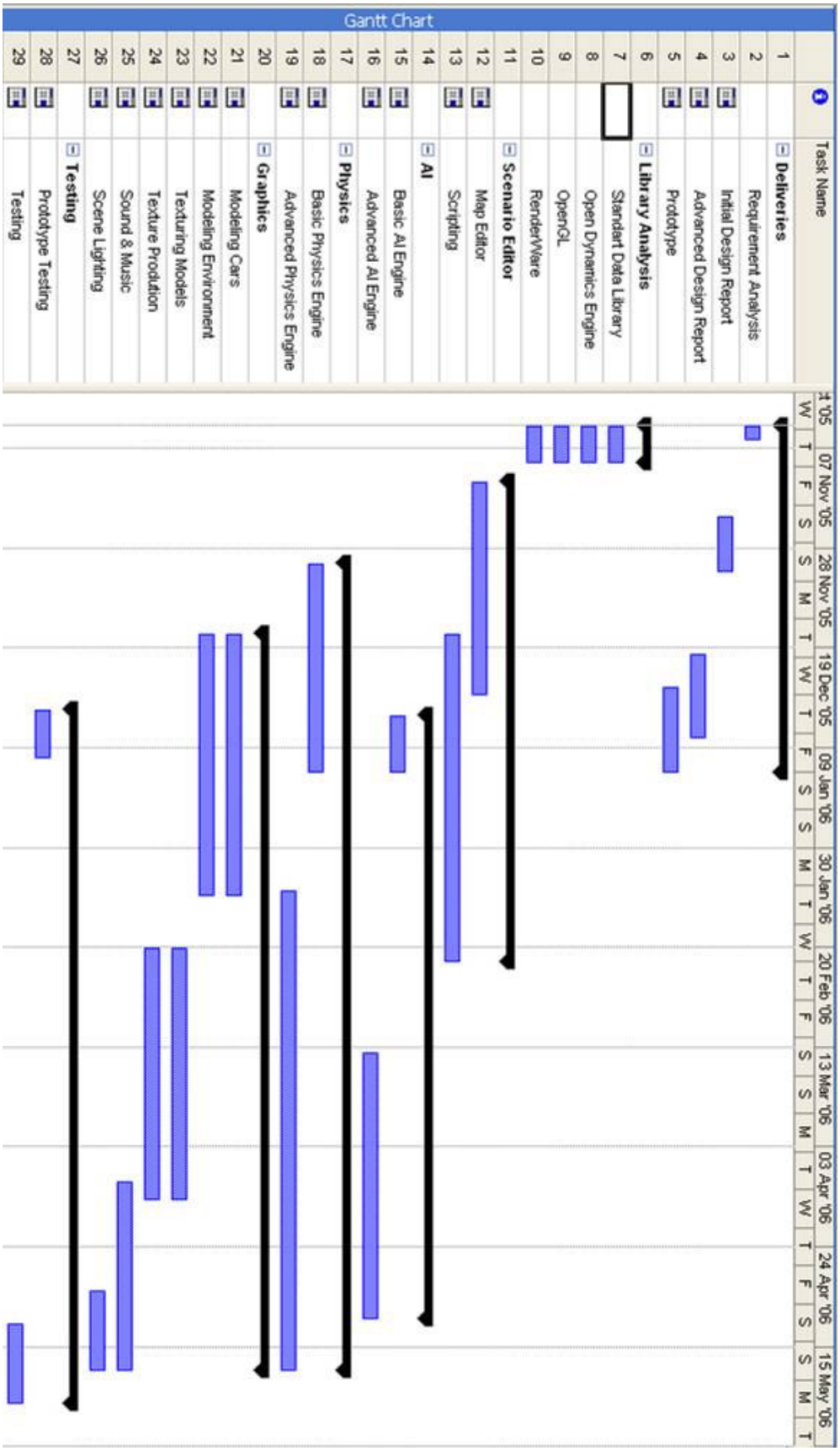
4. Catastrophic

6.3. Overview of Risk Mitigation, Monitoring, Management

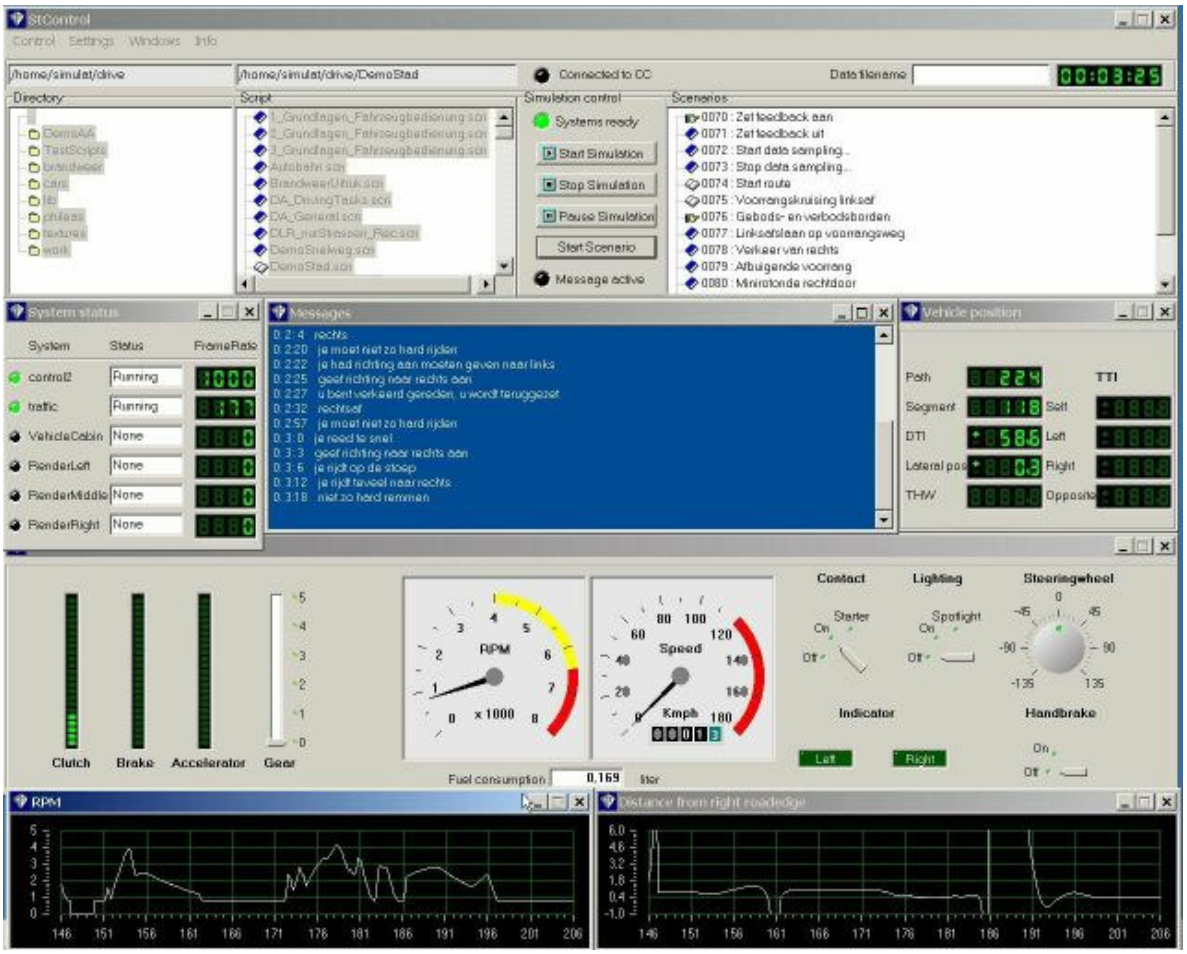
Risks	Mitigation	Monitoring	Management
Lack of Authority	Project meetings will be held in a formal way	Regular vote of confidence for the leader	New leader
Lack of Responsibilities	Team leader will inform members about their tasks regularly	Detect individual undone tasks	Warn the member seriously
Overloading team members	Team members will inform leader about their status	Incomplete tasks	Distribute the members work on others
Postponing tasks	tasks will have strict due dates	Monitor if started on time	Warn members before due dates
Hardware constraints	We will start early to hardware construction.	Ask for progress regularly.	System will work with or without hardware
AI Implementation difficulties	We will find our capabilities early.	Detect progress obstacles	Minimum required ai will be implemented
Graphical Imp. difficulties	We examined current well graphical systems	Unrealistic graphic models	basic car and environment graphics as example
Compatibility of libraries	We will take support from people experienced on these libraries.	Software prototypes don't run.	We will turn to other possible libraries.

Task Name	**Duration**	**Start**	**Finish**
Deliveries	**1 day**	**Wed 02.11.05**	**Wed 02.11.05**
Requirement Analysis	43 days	Mon 21.11.05	Fri 13.01.06
Initial Design Report	10 days	Mon 21.11.05	Fri 02.12.05
Advanced Design Report	15 days	Tue 20.12.05	Fri 06.01.06
Prototype	15 days	Tue 27.12.05	Fri 13.01.06
Library Analysis	**7 days**	**Wed 02.11.05**	**Wed 09.11.05**
Simple Direct Media Layer	7 days	Wed 02.11.05	Wed 09.11.05
Open Dynamics Engine	7 days	Wed 02.11.05	Wed 09.11.05
OpenGL	7 days	Wed 02.11.05	Wed 09.11.05

RenderWare	7 days	Wed 02.11.05	Wed 09.11.05
Scenario Editor	**76 days**	**Mon 14.11.05**	**Wed 22.02.06**
Map Editor	35 days	Mon 14.11.05	Wed 28.12.05
Scripting	50 days	Fri 16.12.05	Wed 22.02.06
AI	**91 days**	**Mon 02.01.06**	**Mon 08.05.06**
Basic AI Engine	10 days	Mon 02.01.06	Fri 13.01.06
Advanced AI Engine	40 days	Tue 14.03.06	Mon 08.05.06
Physics	**125 days**	**Thu 01.12.05**	**Fri 19.05.06**
Basic Physics Engine	35 days	Thu 01.12.05	Fri 13.01.06
Advanced Physics Engine	73 days	Wed 08.02.06	Fri 19.05.06
Graphics	**112 days**	**Fri 16.12.05**	**Fri 19.05.06**
Modeling Cars	40 days	Fri 16.12.05	Wed 08.02.06
Modeling Environment	40 days	Fri 16.12.05	Wed 08.02.06
Texturing Models	39 days	Mon 20.02.06	Thu 13.04.06
Texture Production	39 days	Mon 20.02.06	Thu 13.04.06
Sound & Music	30 days	Mon 10.04.06	Fri 19.05.06
Scene Lighting	13 days	Wed 03.05.06	Fri 19.05.06
Testing	**106 days**	**Sun 01.01.06**	**Fri 26.05.06**
Prototype Testing	8 days	Sun 01.01.06	Tue 10.01.06
Testing	**13 days**	**Wed 10.05.06**	**Fri 26.05.06**


Gantt Chart
Task Name
Deliveries
Requirement Analysis
Initial Design Report
Advanced Design Report
Prototype
Library Analysis
Standart Data Library
Open Dynamics Engine
OpenGL
RenderWare
Scenario Editor
Map Editor
Scripting
AI
Basic AI Engine
Advanced AI Engine
Physics
Basic Physics Engine
Advanced Physics Engine
Graphics
Modeling Cars
Modeling Environment
Texturing Models
Texture Prodution
Sound & Music
Scene Lighting
Testing
Prototype Testing
Testing
07 Nov '05
28 Nov '05
19 Dec '05
09 Jan '06
30 Jan '06
20 Feb '06
13 Mar '06
03 Apr '06
24 Apr '06
15 May '06

APPENDIX



E1

E2


Süre: 00:11:37
Hata: 99/99
Enerji:
km/h
051

E3


VEHICLE SHOWCASE
1968
FORD MUSTANG
FASTBACK
BRAKES
FRONT Brakes
Excellent
FRONT Hand Brake
Good
REAR Brakes
Good
REAR Hand Brake
Good
RESET SETTINGS
Engine Displacement: 302 V8
Horsepower: 230 @ 4800 RPM
Torque: 310 ft.lbs. @ 2800 RPM
Weight: 2,823 lbs.
Wheelbase: 108"
Overall Length: 183.6"
Website: http://www.fordvehicles.com/mustan
CANCEL
DONE

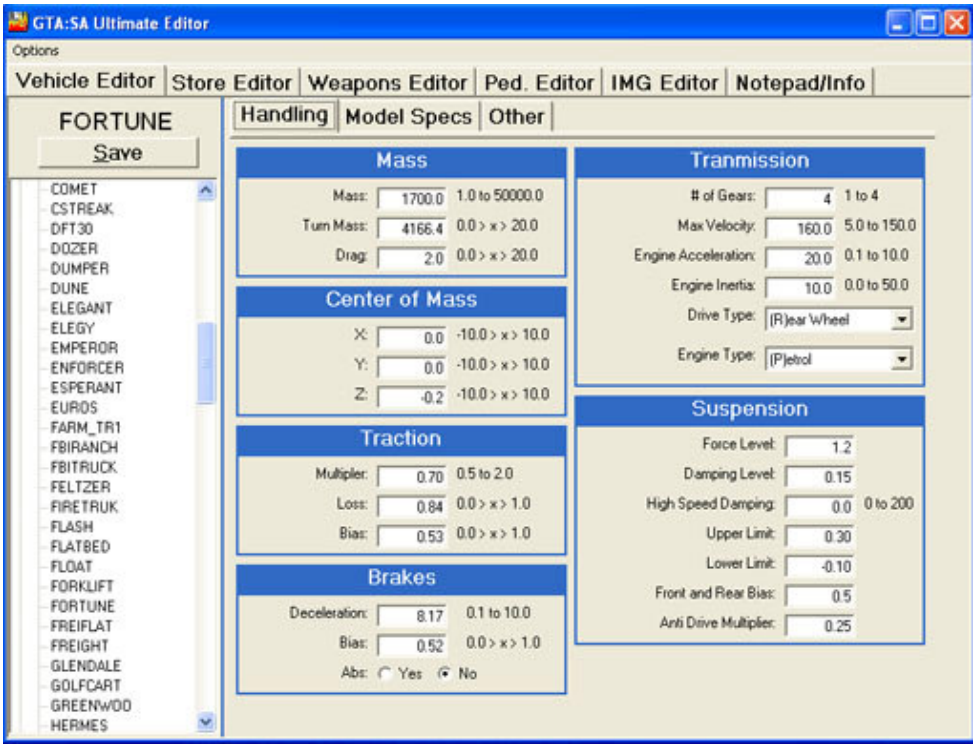
E4


1/2 Laps
Current: 1:30.87
Best: 0:00.00
EMPIRE
HOTEL
1st
1/4
1: PLAYER
2: + 0.92
3: + 0.97
4: + 1.13
N2O
Turbo
2
MPH
69

E5


UNDRGND

E6


GTA:SA Ultimate Editor
Options
Vehicle Editor
Store Editor
Weapons Editor
Ped. Editor
IMG Editor
Notepad/Info
FORTUNE
Save
COMET
CSTREAK
DFT30
DOZER
DUMPER
DUNE
ELEGANT
ELEGY
EMPEROR
ENFORCER
ESPERANT
EUROS
FARM_TR1
FBIRANCH
FBITRUCK
FELTZER
FIRETRUK
FLASH
FLATBED
FLOAT
FORKLIFT
FORTUNE
FREIFLAT
FREIGHT
GLENDALE
GOLFCART
GREENWOO
HERMES
Handling
Model Specs
Other
Mass
Mass: 1700.0 1.0 to 50000.0
Turn Mass: 4166.4 0.0 > x > 20.0
Drag: 2.0 0.0 > x > 20.0
Center of Mass
X: 0.0 -10.0 > x > 10.0
Y: 0.0 -10.0 > x > 10.0
Z: -0.2 -10.0 > x > 10.0
Traction
Multiplier: 0.70 0.5 to 2.0
Loss: 0.84 0.0 > x > 1.0
Bias: 0.53 0.0 > x > 1.0
Brakes
Deceleration: 8.17 0.1 to 10.0
Bias: 0.52 0.0 > x > 1.0
Abs: Yes No
Tranmission
# of Gears: 4 1 to 4
Max Velocity: 160.0 5.0 to 150.0
Engine Acceleration: 20.0 0.1 to 10.0
Engine Inertia: 10.0 0.0 to 50.0
Drive Type: (R)ear Wheel
Engine Type: (P)etrol
Suspension
Force Level: 1.2
Damping Level: 0.15
High Speed Damping: 0.0 0 to 200
Upper Limit: 0.30
Lower Limit: -0.10
Front and Rear Bias: 0.5
Anti Drive Multiplier: 0.25

E7


© Andre Hotzler
21:46
$01242415
AUTO
N

E8

F) Notes from the Meeting with Driving Teachers

During the meeting with three different driving teachers we noted down some advises and some important points. Their educations help the drivers to gain reflex while repeating some traffic actions. The following actions are controlled during the education:

- Departure
- Posture
- Traffic lane follow-up
- Mirror controls
- Gear transformations
- Driver Comfort
- Roadway laws appliance
- Turnings and signals
- Parking
- Overtaking
- Gear Control and engine brake
- Steering Wheel movements and control
- General behavior.

They also explained that most important factor of driving is real traffic experience that also includes some unexpected events. We gave more importance to artificial intelligence according to that topic in order to achive reality and usability.