

# **“research-theoretical text” of Team Final Year Project of Game Production**

by Yiu, the game engine designer in the team

by LI Yiu Chong (Yiu), 50262340

On 15<sup>th</sup> to 23<sup>rd</sup> April

## **Purpose of this document**

This document is being prepared for the following purpose:

- To provide a general understanding of our project and my role in the production, to any outsider.
- To state and review the motive of this project
- To record the research process throughout the whole project
- To document the achievements made through the production of this game and its process
- To express the feeling on the production process of this game
- To round-up the things I did in this year of learning and the matter that I have to deal within the time period

Hence the writing style and content of this document may look different from the requirement of the so-called “research-theoretical text” from school. I must say I would like to write in my own voice and to the future of me rather than solely for the final year project purpose. However, I would try to cover the issues required by the guideline of Final Year Project from school, while at the same time, the content of this document would be wider than the requirement in the extent.

## **Motive of Producing Game, Teaming up and in a wider extent, doing Final Year Project**

Very long time ago (if not using the term “Once upon a time...”, and so please be patient for my dilatory writing style!), probably before the year 2 of my tertiary study, when I was to select my studying stream, I was asking myself “Which stream should I choose?” At that time, the school offered Documentary, Television Commercial, Installation and Interactive stream, and I had chosen the Interactive stream. The reason was: I choose Creative Media as my tertiary study simply because I would like to shift my focus from technical aspect of computer studies to somewhere that involve more human input, such as in creativity aspect, while not leaving computation/interface design too far away. I was not strong in thinking a gist for a commercial, my hand would shake when I am taking a camera, and the syllabus of the Installation stream did not attract my attention. The only stream that I could choose was Interactive.

I would not say the stream was useless or bad; however, the learning experience of the stream was a bit different from what I was imagined. I was to deal with the design of interface, but in most situation I have to deal with interface in 2 dimensional web environment (time is also an element that I have to deal with, so you may say it was “3D”). The course covered many aspect in design, and that was the time that I knew I was weak in design.

There was one assignment in the course of that stream which, was to design a simple game system within a month (or around a month). That was the time I thought of what a game was, i.e. rules and regulation, goal design, fairness maintenance, etc. I did a bus station master role-playing (or bus station simulation, or bus allocation real time strategy, or...) game for that assignment, and the only positive comment was on gameplay design.

Another chance to design a gameplay was in the course “Advanced computing” by Professor Mark. I did a bus driving simulation using the software “Blender” (A lack of auto-save function droved me crazy some days a week, and then Professor Mike explained to us the reason for a lack of auto-saving in some software in another class, and I know I am side-tracking.) for my final project, and that was the time I had to feel the existence of 3 dimensions.

At the time, Bun was my classmate of the same stream. We often said that we would like to make game in future, and our plan of final year project was initiated. We did select and filter out possible partners, such as Ryan, Eva, Leo, Wah, and even non-BA schoolmate such as May. (And finally GeN has agreed to join the group, which was a piece of nice news to us.)

Time flies and it was the summer vacation after the study of second year. Finally we had to face the matter of the topic for final year project. I read a book of game design and team work preparation, and that started our imagination of game genres and gameplay situation. I did have alternative topics that I would like to work on as my final year project, such as studying the emotion of robot (I had never mentioned this to any professor or tutor, I simply thought that they/you would consider me as “too simple, sometimes naïve”), or just as the second topic suggested by Professor Mike: “Comparative study of Japanese and North American animations in the 90s (Comparative Study)” (I did consider that I was too young for this topic, i.e., I did not think I gained enough knowledge in early 90s anime, especially those in America.). Base on the assumption that I would not continue my study in any master of philosophy, and hence need not a thesis writing for my future, I decided to “formally” participate in this game production.

## General Statement of this Game and my role in the production

This is a traffic management strategy / traffic cop role playing / real time action game (I simply think that the genres division of game is misleading...). In other words, the game is based in the road intersection of the area outside SOGO in Causeway Bay, and the player's mission is to maintain a good traffic flow by sorting people and vehicles. My role is the gameplay designer who is responsible for (or I had acceded to deal with the following):

- building a operative platform for the game
- managing the flow of the game
- participating in the design of the game environment and objects' action in the game
- participating in the research of interface and its production
- dealing with input

## Research Process/Record of Achievement/Version Info/My Monthly "Diary"

### Before September, 2002

We were talking with different professors to decide the genres of our game, including Professor Steve Fore and Professor Mike Wong. At the same time we also have to decide the advisors of our production. Our suggestion for "Bus Driving Simulation", "Film Production", "Mini Game Narrative" and "Role-Play game in Public Estate" was rejected by Mike. We had a more than 2 hours discussion with Mike and finally we initiated our topic. We were required to send a preliminary proposal to school for approval of teaming up at late August and prepare for the idea presentation in the first week of September, so we thought of the aims, theme, target audiences, division of labour and equipment list for this game production. Also we started to look for the material for research on our production.

### September, 2002

A dark age came to me at the time. My father was in debts and my mothers would like to divorce from my father. I was also a helper for the orientation week to the whole school. Time was clamor. I ran wild in the presentation of our project. But that is life.

I had got a book solely talking about the C++ language, so I was trying to build some extremely simply program with the Microsoft Visual C++ 6.0 compiler, such as loop terminology and using different data type. The first two chapter of "Tricks of the Windows Game Programming Gurus SE" (by Andre LaMothe) helped me to gain a general understanding on how a game loop (check input -> process game logic-> collision check-> physics-> output-> sync-> process windows event and again and again...) and the way to make a window.

## October, 2002

I went deeper. I was trying to use the world simplest way to do calculation and to output text string.

Two field studies took place on 6<sup>th</sup> and 13<sup>th</sup> in Causeway Bay, one day time and one night time. We found that it was hard to photo and record sound of the object and environment, and Bun may have to produce the graphic by drawing or 3D generation. I did jot some notes for the environment, such as the duration of traffic light cycle, the passing bus line information, the amount and flow of human, although those figures did not show great use during the production. We had drawn some conclusion and did the first game specification on the mid-October.

When Bun was thinking of the different type of sprites, I was first designing the player's action and opponents' reaction and the effect to indexes. Chart of this was included in the second version of game specification.

My last week in October was devoted to proposal writing which was deadline on 1<sup>st</sup> of November (Later on I found that it was stupid to repeat saying the same thing in different chapters, it was hard for advisors to read in detail and that just did us no good...). Finally we got a proposal of about 100 pages, if not exceeding...

## November, 2002

I did not feel good in the whole November, my heart was grey. This was confirmed by my own diary.

I skipped some sessions and read the chapter 9 of Andre's book for Direct Input. Getting relative mode of mouse coordinate seemed not as difficult as absolute mode, and input from devices can be obtained just by a few steps (Create Direct Input Object->Create Device->Set Device Cooperative Level->Set Device Data Format->Acquire and Poll Device->Retrieve data from device). The cooperative flags of devices should be considered and decided with great care, as this would affect whether other programs running in operating system can access the device or not. By the time I decide to deal with the keyboard and mouse device first, leaving gamepad/joystick a little bit later on.

At the later part of November I was preparing to write a windows program, handling window messages and events, changing icons and using task menu bar. Following that I was trying to use the Graphical User Interface of Windows programming to do some graphical testing, in order to know the relationship between sprites' movement and graphical display, for example, a sprite forward movement means the x-coordinate of 4 corners of the sprite sheet is increasing,

a resize of sprite means a synchronizing change in x and y coordinate of 4 corners, a mouse drag on sprite object means a synchronizing move in x and y coordinate. Besides, I tried to deal with mouse and keyboard too.

### December, 2002

The bad mood continued for some weeks...

The 1<sup>st</sup> Windows program was compiled and shown to Mike on 5<sup>th</sup> Dec, and became the 0.1 edition of my game engine, although it was nothing related to the gameplay. Some days were spent on researching of Windows programming and multi-thread using (at the time I was thinking the reason why Mike suggested me to acquire knowledge on multi-thread programming, however, just some weeks later, Intel issued their Pentium 4 Hyper-Threading processor...), although till the day of writing this document those techniques was not applied.

Since the graphic and graphic engine for the game was not yet ready in mid-December, I was thinking of how the environment of the game would be. Should a 2 dimensional environment be used for a 2.5 dimensional game? I was trying to find the answer in some mini dojin game (同人遊戯, some games produced by individuals or small group for non-profit making purpose, and the characters in the game were based on anime or manga/comic book...) and some other 2 or 2.5 dimensional games by small scale companies, but I could not find an absolute answer.

As I was sitting there with nothing that I could think of, I tried applying Direct Input in input acquisition (and having some holidays for Christmas, the time for me to clean up the unwatched anime series...). Soon my own project folder was replaced by GeN's initial graphic engine project folder (I pretty like GeN's clean programming syntax...), and application of Direct Input on keyboard and mouse was easily transferred to the new project folder. GeN's explanation led me to think of the necessity of using a 3 dimensional environment.

### January, 2003

I was continuing the checking on GeN's initial graphic engine. I was checking the information the gameplay engine has to pass to the graphic engine for the generation of a sprite on screen, as follow:

X-coordinate (X), Y-coordinate (Y), Z-coordinate(Z)

- to locate the top left corner of the sprite...

X-velocity (Vx), Y-velocity (Vy), Z-velocity (Vz)

- to indicate the velocity, and to indicate the direction of the sprite facing

Width (W), Height (H)

- not necessary if the sprite does not change its size

Action

- to tell the graphic engine which action of the sprite should be showing

Sprite Name/Sprite Type

- to indicate which sprite itself is

This was also for me to design the object class too. And at the meantime I was also designing the game loop (Initialization -> Level Preparation & Count Down -> Gameplay Operation -> Level End Operation -> Exit Game). The game time for one level was set to 180 seconds.

Following are to be done in each stage in the game loop:

Game Initialization:

- Direct Input Initialization
- For Graphic Engine to get ready sound clip and image/visual object

Level Preparation & Count Down:

- To set every global variable required for use during the game level processing
- To initialize indexes
- To generate text buffer for output display
- To set properties of objects according to different levels
- Place menu here if any...

Gameplay Operation:

- Retrieve Input from Direct Input Objects and from Windows message
- Link sprites' reaction to input
- Sprite Objects collision check
- Reaction to Traffic Light Object
- Artificial Intelligence of Sprite Object itself
- Response of Indexes
- Output, and then loop again

Level End Operation:

- Delete objects
- Calculate result according to Indexes
- Display result
- Save / Record to file
- Process to next level / Game Over

Exit Game:

- Delete Direct Input objects
- Delete object in graphic engine
- Handle Windows message

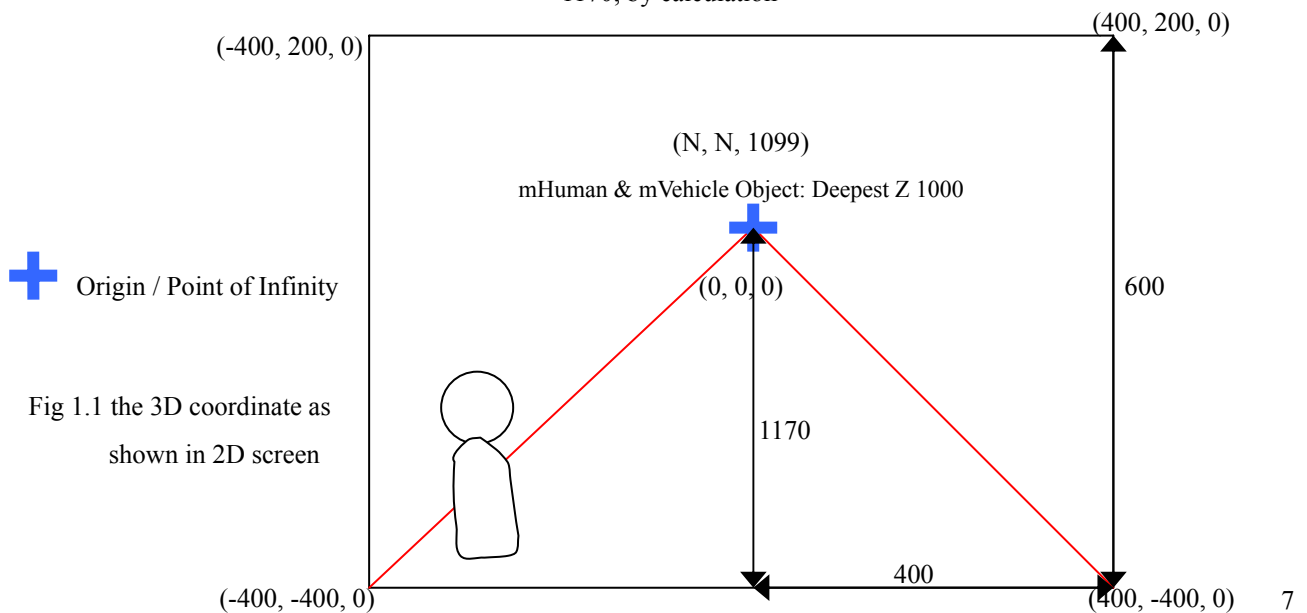
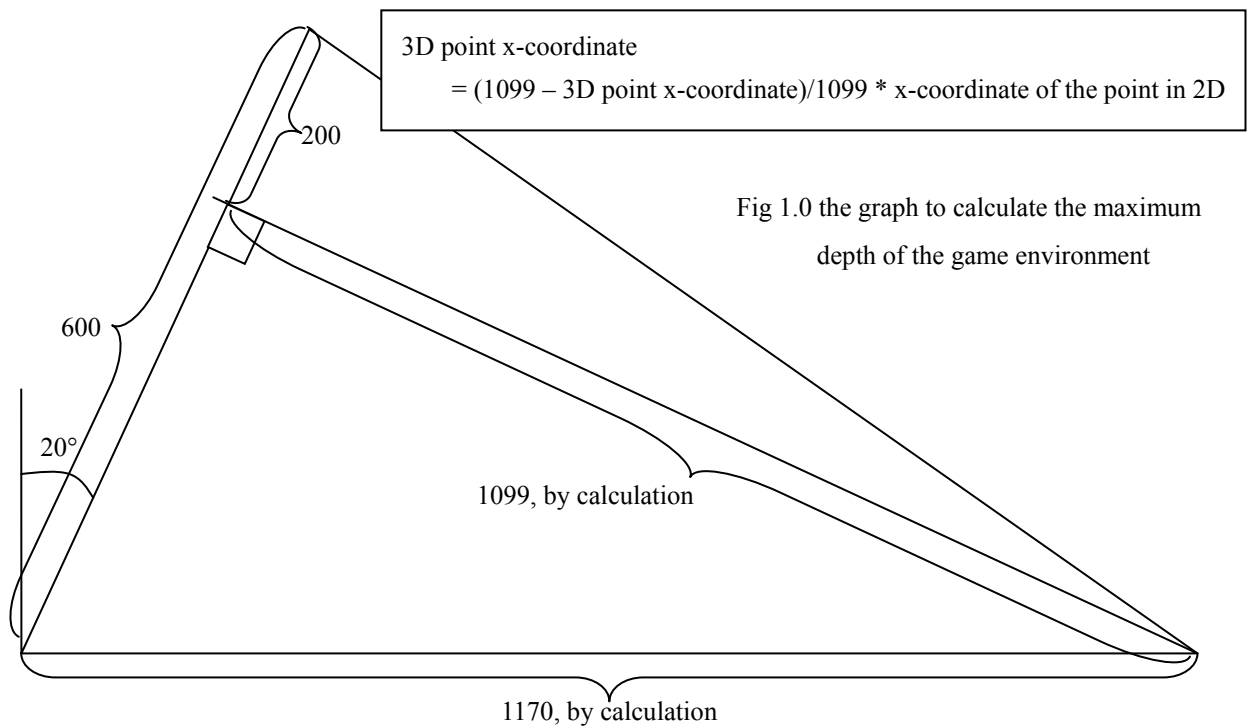
Mouse object of Direct Input keep having problem retrieving the coordinate of mouse cursor, it was showing the coordinate exceeding the screen resolution. There would not be such problem on the system which the mouse speed was set to middle and with enhancement on precision of cursor disabled in control panel of the Windows operating system (I was compiling the program under Windows XP...).

February, 2003

Chinese Lunar New Year just did not bring any effect to us; my process was just below anticipation.

I started feeling silly to have agreed to apply Direct Input into the game by myself. Since mouse input and screen output was in 2 dimensional and the environment was in 3 dimensional system, I had to deal with the conversion in 2 dimension to 3 dimension. It is because when an object is moving to the reverse side of the camera toward the background, it will get smaller and its coordinate will move toward the point of infinity until it disappears in the point, so to point to the object the coordinate of the cursor will change, even though the x and y coordinate of the object did not change actually.

To have the solution, I questioned GeN and constructed the illustrations on below. This was to get the Z-coordinate of the point of infinity in his engine, and hence I could find the formula for making conversion between 2 dimensional coordinate system to 3 dimensional system, which is also listed beside the illustrations.



By then the shape of the game environment was also defined (illustration below...). Initially I was making the environment as 800 \* 600 \* 1099, but when I was thinking the case when a human object was walking pass the red line as shown on figure 1.1, I decide to leave more space for X-coordinate and made it 1000 \* 600 \* 1000.

How about a pure change in Z-coordinate of an object moving toward the screen (or camera, using the jargon of 3D software...)? I enquired GeN in March and he answered that the object would have sudden off effect... Anyway, everything was on paper at that time and just a few manipulations was made to the project, such as input and text display, but that was good start for the work on next month.

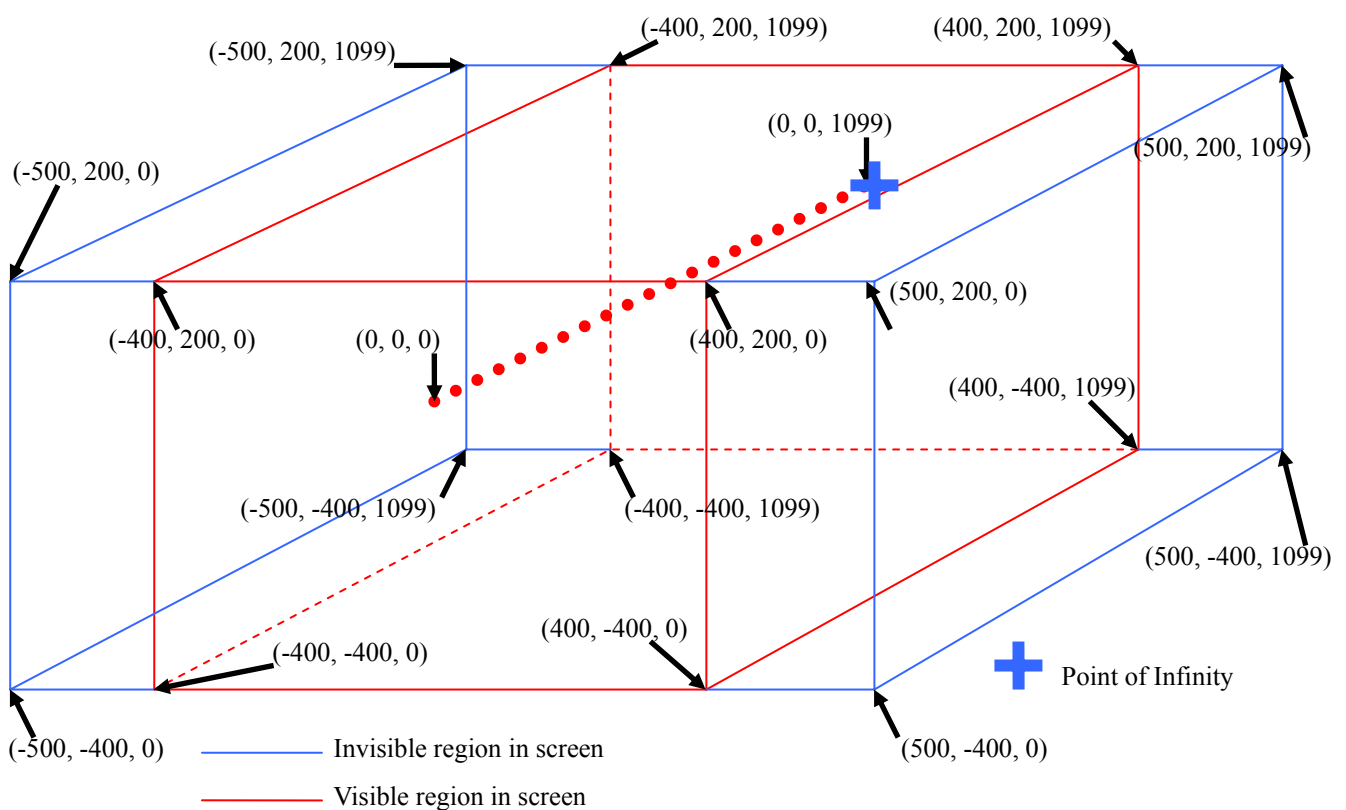


Fig 1.2 3D coordinates of the whole game environment

### March, 2003

So just after the finalization of game environment, I started finalizing (although there were still many introductions of properties to the class in later stage...) the properties in class of different objects, as listed in follow:

Object Class	mHuman * <sup>23</sup>	mVehicle * <sup>23</sup>	mTraLight * <sup>23</sup>
X-coordinate (mx in position structure) * <sup>0,1</sup>	-500 to 500 * <sup>2</sup>	-500 to 500 * <sup>2</sup>	-400 to 400 * <sup>2</sup>
Y-coordinate (my in position structure) * <sup>0,1</sup>	-400 to 200 * <sup>2</sup>	-400 to 200 * <sup>2</sup>	-400 to 200 * <sup>2</sup>
Z-coordinate (mz in position structure) * <sup>0,1</sup>	>0 to 1000 * <sup>3</sup>	>0 to 1000 * <sup>3</sup>	125 * <sup>4</sup>
Velocity in X-coordinate (mvx in vposition structure) * <sup>0,5</sup>	>0 to 4 * <sup>6</sup>	0 * <sup>6</sup>	/ * <sup>7</sup>
Velocity in Y-coordinate (mvy in vposition structure) * <sup>0,5</sup>	0 * <sup>6</sup>	0 * <sup>6</sup>	/ * <sup>7</sup>
Velocity in Z-coordinate (mvz in vposition structure) * <sup>0,5</sup>	0 * <sup>6</sup>	>0 to 5 * <sup>6</sup>	/ * <sup>7</sup>

Object Class	mHuman * <sup>23</sup>	mVehicle * <sup>23</sup>	mTraLight * <sup>23</sup>
Default Velocity X (mcvx in vposition structure) * <sup>0,5</sup>	>0 to 4 * <sup>6</sup>	0 * <sup>6</sup>	/ * <sup>7</sup>
Default Velocity Y (mcy in vposition structure) * <sup>0,5</sup>	0 * <sup>6</sup>	0 * <sup>6</sup>	/ * <sup>7</sup>
Default Velocity Z (mcvz in vposition structure) * <sup>0,5</sup>	0 * <sup>6</sup>	>0 to 5 * <sup>6</sup>	/ * <sup>7</sup>
Time Stamp on sprite creation (time) * <sup>0,8</sup>	GetTickCount()	GetTickCount()	GetTickCount()
Sprite Action Indicator (action) * <sup>0,9</sup>	0,2,3,4,5,6,7,8,9	0,2,3,4,5,6	0,1,2,3,4,5
Record whether sprite ended its life (destroy) * <sup>0,10</sup>	0,1	0,1	/ * <sup>7</sup>
Sprite Type Indicator (sprite) * <sup>0,11</sup>	2,3,4	2,3,4	2,3,12,13
Sprite Type Indicator Text String (spriteType) * <sup>0,12</sup>	(string)	(string)	(string)
Collision Check X-coordinate (hx) * <sup>0,13</sup>	0 to 60	0 to 150	0 to 40
Collision Check Y-coordinate (hy) * <sup>0,13</sup>	0 to 100	0 to 250	0 to 180
Collision Check Width (hw) * <sup>0,13</sup>	0 to 60	0 to 150	0 to 40
Collision Check Height (hh) * <sup>0,13</sup>	0 to 100	0 to 250	0 to 180
Attack Player Indicator (attacked) * <sup>0,14</sup>	0,1	/ * <sup>7</sup>	/ * <sup>7</sup>
Attack Player Probability (random) * <sup>0,15</sup>	(integer)	/ * <sup>7</sup>	/ * <sup>7</sup>
Attack Validity (valid) * <sup>0,16</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>
Attack Direction (direction) * <sup>0,17</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>
Object Class	mBg * <sup>23</sup>	mSurfaceObj * <sup>23</sup>	mAttack * <sup>23</sup>
X-coordinate (mx in position structure) * <sup>0,1</sup>	-400 to 400	-400 to 400	-n to n * <sup>19</sup>
Y-coordinate (my in position structure) * <sup>0,1</sup>	-400 to 200	-400 to 200	-m to m * <sup>20</sup>
Z-coordinate (mz in position structure) * <sup>0,1</sup>	>0 to <1099	<0 or 1099 * <sup>18</sup>	0 * <sup>21</sup>
Velocity in X-coordinate (mvx in vposition structure) * <sup>0,5</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>
Velocity in Y-coordinate (mvy in vposition structure) * <sup>0,5</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>
Velocity in Z-coordinate (mvz in vposition structure) * <sup>0,5</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>
Default Velocity X (mcvx in vposition structure) * <sup>0,5</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>
Default Velocity Y (mcy in vposition structure) * <sup>0,5</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>
Default Velocity Z (mcvz in vposition structure) * <sup>0,5</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>
Time Stamp on sprite creation (time) * <sup>0,8</sup>	GetTickCount()	GetTickCount()	GetTickCount()
Sprite Action Indicator (action) * <sup>0,9</sup>	/ * <sup>7</sup>	0,3,4,5	0,2,3,4
Record whether sprite ended its life (destroy) * <sup>0,10</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>
Sprite Type Indicator (sprite) * <sup>0,11</sup>	(unlimited) * <sup>22</sup>	2,5,6,1	2,3
Sprite Type Indicator Text String (spriteType) * <sup>0,12</sup>	(string) * <sup>22</sup>	(string)	(string)
Collision Check X-coordinate (hx) * <sup>0,13</sup>	(unlimited) * <sup>22</sup>	0 to 40	0 to 40
Collision Check Y-coordinate (hy) * <sup>0,13</sup>	(unlimited) * <sup>22</sup>	0 to 40	0 to 40
Collision Check Width (hw) * <sup>0,13</sup>	(unlimited) * <sup>22</sup>	0 to 40	0 to 40
Collision Check Height (hh) * <sup>0,13</sup>	(unlimited) * <sup>22</sup>	0 to 40	0 to 40
Attack Player Indicator (attacked) * <sup>0,14</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>
Attack Player Probability (random) * <sup>0,15</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>
Attack Validity (valid) * <sup>0,16</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>	0,1
Attack Direction (direction) * <sup>0,17</sup>	/ * <sup>7</sup>	/ * <sup>7</sup>	0,1

## Remarks

- \*0 class object properties are in different type:  
mx, my, mz, mvx, mvy, mvz, mcvx, mcvy, mcvz, hx, hy, hw, hh – floating point figure  
time, action, destroy, sprite, attacked, random, valid, direction – integer  
spriteType - string
- \*1 X, Y, Z are for storing coordinate in 3D coordinate system, except mSurfaceObj class, which is for sorting layers of different class objects in 2D coordinate system
- \*2 For those object class with X-coordinate -500 to 500, when the objects are wholly or partly entered the zone -400 to 400 will be shown on the screen. Y-coordinate of all classes are just fitting the the visible region of the screen.
- \*3 Z-coordinate of any objects which are under 3D coordinate system would be limited to >0 to 1000. Z-coordinate of 0 and below is reserved for mSurfaceObj class and mAttack class.
- \*4 The Z-coordinate of traffic light object was limited to 125 at level 1, however, the objects of mTraffic class can be ranged from >0 to 1000.
- \*5 mvx, mvy & mvz are used for storing the current velocity of the objects, while mcvx, mcvy & mcvz are reserved for storage of their default velocity. 3 coordinates can be recorded, although finally only velocity of 1 coordinate was used.
- \*6 The figure means the velocity of the object 1/50 second since 1/50 second is the update rate of all object.
- \*7 Not applicable / no such property in the class.
- \*8 Time Stamp records the time the object is constructed (except mAttack class, which changed when the attack is created). This property is only useful in mAttack class, but I still considered to keep the Time Stamp on each object, just if it may be useful in future development.
- \*9 action is used to record the action the sprite is processing, different values have different meanings in different classes:  
mHuman – 0 = clear, 2 = still, 3 = walk, 4 = run, 5 = push (being pushed),  
6 = fight (fight against player), 7 = injury (due to player's attack), 8 = crash (by vehicle),  
9 = stepback (step backward when seeing red or flash green traffic light)  
mVehicle – 0 = clear, 2 = still, 3 = accelerate, 4 = run, 5 = decelerate, 6 = crash (with human object)  
mTraLight – 0 = clear, 1 = off, 2 = red, 3 = red+yellow, 4 = green,  
5 = yellow (for vehicle traffic light) / flash green (for pedestrian traffic light)  
mSurfaceObj – 0 = non-game cursor (for mouse) / default (for background), 3 = push cursor (for mouse),  
4 = fight cursor (for mouse), 5 = dodge cursor (for mouse)  
mAttack – 0 = clear, 2 = attack (when the attack is on the air), 3 = explode (when it hits player / human),  
4 = dodge (the attack is dodged by player in dodge mode, only to the attack object to player)
- \*10 Record whether the object was out of the environment or destroyed. (0 = not destroyed, 1 = destroyed)
- \*11 This is for record the object's sprite sheet type. The figure listed is only applicable to level 1. For the whole game, it has no limitation in this value (just limited by our effort maybe...)  
mHuman – 2 = old female woman 01,  
3 = office type 01,

4 = strong man 01  
mVehicle – 2 = double decker bus 01,

3 = private car 01,

4 = motorbike 01

mTraLight – 2 = vehicle light front,

3 = vehicle light back,

12 = pedestrian light front,

13 = pedestrian light back

mSurfaceObj – 2 = mouse cursor,

5 = covering interface,

6 = deepest background,

1 = mark

mAttack – 2 = attack object toward player,

3 = attack object toward human object

- \*12 A text string for recording sprite name, initially reserved for file name of sprite sheet for graphic engine.
- \*13 It is really hard to explain these 4 properties without illustration, and fig 1.3 in the next page will show...
- \*14 This is specially for mHuman class to indicate whether they will attack player or not. For those human objects who do not actively attack player, the property will be initialized as 0 and changed to 1 if they are attacked. For those who will actively attack player no matter being attacked or not, the property will be initialized as 1 at their constructor function.
- \*15 This record the probability the human object will attack player. Different types of human objects will be given different values.
- \*16 This recorded whether the attack is still valid, i.e. within the valid time of the attack object and not dodged by player (if it is attacking to player). 0 means invalid and 1 means valid.
- \*17 This record the direction of attack object, either 0 = attack to player or 1 = attack to any human object.
- \*18 The biggest in value, the deepest the object is. Mouse is given -1, Index panel is 0 and deepest background is 1099.
- \*19  $-n$  to  $n$  is  $-400$  plus object width / 2 to  $400$  plus object width / 2.
- \*20  $-m$  to  $m$  is  $-400$  minus object height / 2 to  $200$  minus object height / 2.
- \*21 The order to display the object of the class was decided by the Time Stamp.
- \*22 There is no limitation in game engine, simply because the class designed was not used in current stage.
- \*23 Classes are designed to record the following object:
  - mHuman – human object, i.e. pedestrians and walkers on the street
  - mVehicle – any vehicle operating on the road
  - mTraLight – traffic lights, both pedestrian traffic light and vehicle traffic light
  - mBg – specially designed to record the background which will overlap some objects, and that lead the possibility of collision of those objects with mBg objects, but till now no object was created using this class, and so no function was designed during the programming stage later on.
  - mSurfaceObj – mouse cursor, and any object except attacks and not necessary to record with 3D, such as

menu items and deepest background in the game.

mAttack – attack object, both the attack done by human toward player and done by player to human objects.

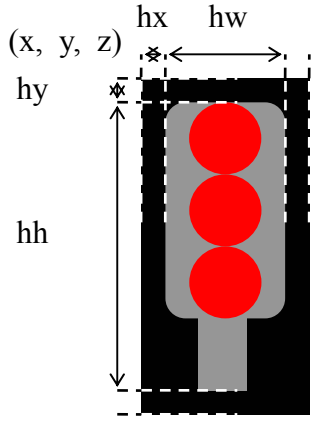


Fig 1.3  
the relationship of x, y, z, hx, hy, hw and hh properties. (demonstrated using a traffic light object)

So every object class is documented above. In the beginning stage I was only planning to have mHuman, mVehicle, mTraLight and mSurfaceObj, so only 4 classes appeared in the early version. Also, no such amount of properties was created to those classes too, indeed only mx, my, mz, mvx, mvy, mvz, type (later on named to sprite), time and action was there. Later on when I am

thinking on the action of collision checking, I remember that the empty area of a sprite sheet was filled black, and those area should not be included in collision check according to the meeting with mates in earlier stage, so I added hx, hy, hw and hh as shown in figure 1.3.

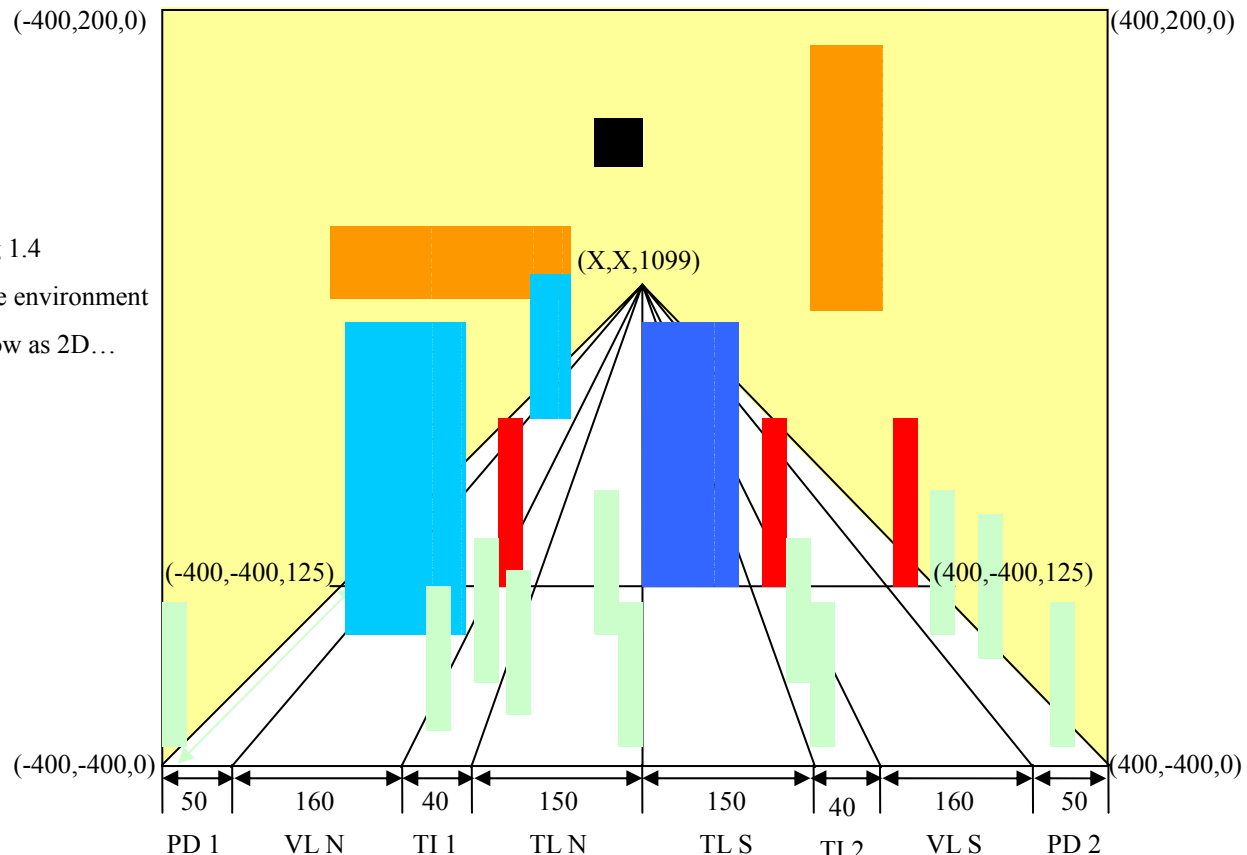


Fig 1.4  
The environment show as 2D...

Human/Pedestrian	Background Object	TL N – Tram Lane Northward	TL 1 – Traffic Island 1
Traffic Light	Deepest Background	TL S – Tram Lane Southward	TL 2 – Traffic Island 2
Vehicle	Tram	Mouse Cursor	VL N – Vehicle Lane Northward
			PD 1 – Pedestrian Road 1
			VL S – Vehicle Lane Southward
			PD 2 – Pedestrian Road 2

Also because of collision check, I was thinking of the necessity to check the background object which lies in front of some objects, so I prepared to have mSurfaceObj, although I was not informed that there will be such kind of object actually in the game. The mSurfaceObj was base on mTraLight class, and just the action property was removed at the meantime. The destroy, attacked, random, valid and direction property and mAttack class was added further later when the attack action was available on early April.

After design of classes it was the design of the first level. Before the end of March I was unable to contact Bun, the graphic designer, to discuss the design of level, so I had dreamed and realized the level by myself (maybe it was a wrong decision, anyway, anyway, I did learn how to think in 3 dimension by myself without help of 3D software...). Figure 1.4 on previous page and figure 1.5 below will show you how the environment of level 1 looks:

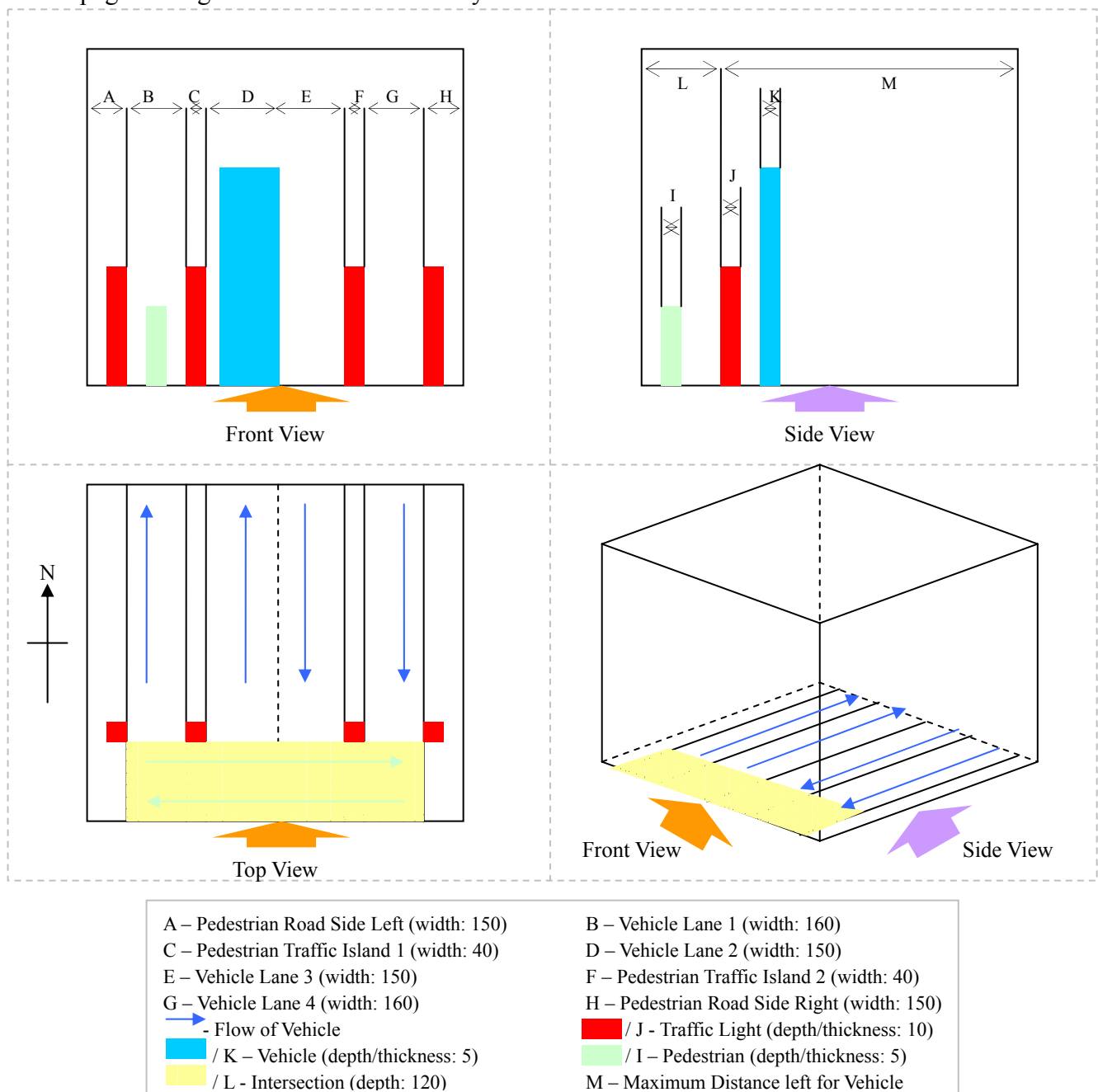


Fig 1.5 The environment shown as 3D...

Since I did not have any graphic for reference, the environment illustrates by the above figures may have further change in later stage of our production.

At the beginning of production I had thought of loading data from files during the game execution, and the data required to load from file was also listed in my own notes, but finally I decided not to load file on the initialization stage or during the game, because both GeN and me did not realize any case which such loading was necessary, and file loading was slow. The variable and attributes used in the gameplay engine were all defined in header files according to the objects' class, and no data would be needed from new source during the execution of the game.

Just at the last week of March, I realized that it would be too simple to simplify the traffic light (both pedestrian traffic light and vehicle traffic light) to only two signals: red and green. So when designing the first level, I also listed the time duration of different light signal during the gameplay:

Sprite Type and State:

2 – Vehicle Light (façade) | 3 – Vehicle Light (back) |

4 – Pedestrian Light (façade) | 5 – Pedestrian Light (back)

1 Cycle (30 secs) Type	Duration 1 (0 <sup>th</sup> – 6 <sup>th</sup> sec)	Duration 2 (6 <sup>th</sup> – 10 <sup>th</sup> sec)	Duration 3 (10 <sup>th</sup> – 11.5 <sup>th</sup> sec)	Duration 4 (11.5 <sup>th</sup> – 28 <sup>th</sup> sec)	Duration 5 (28 <sup>th</sup> – 30 <sup>th</sup> sec)
2	R	R	R+Y	G	Y
3	R	R	R+Y	G	Y
4	G	G Flash	R	R	R
5	G	G Flash	R	R	R

So a month had passed.

### April, 2003

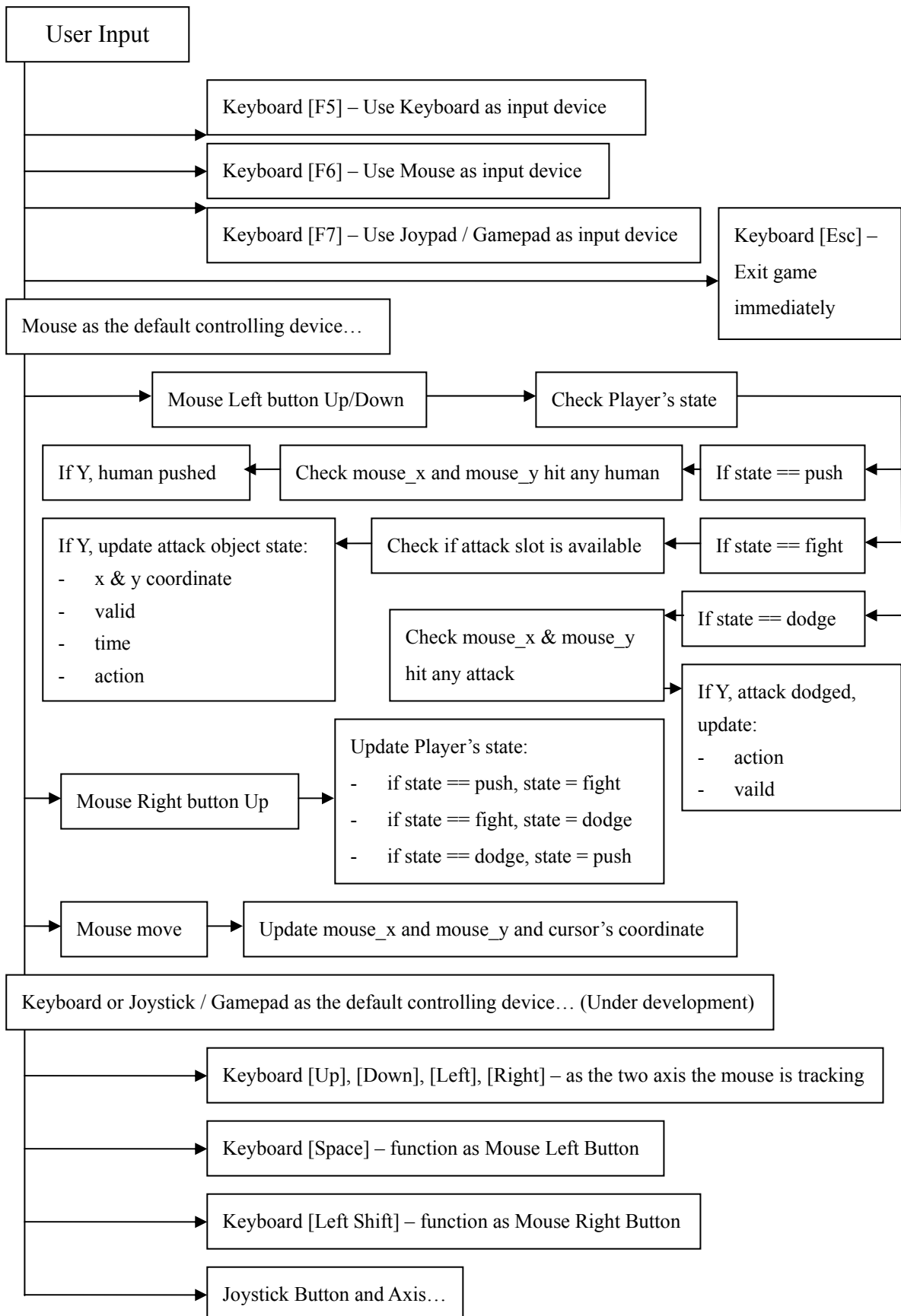
It was a shame to compare my production schedule inside our proposal with my current progress. I knew I had been late for many tasks. Anyway, this was the month that I devoted the most time on the compiler and did the core programming. Fortunately, my own progress was quite smooth, though maybe I was too late to inform GeN the very initial gameplay engine was produced.

It was the time to think of the action of every single object in details. There was human, vehicle, traffic light, background, surface object and attack class, and the game engine was to deal with their own artificial intelligence and collisions of them. I decided to have the sequence of analysis in the following order:

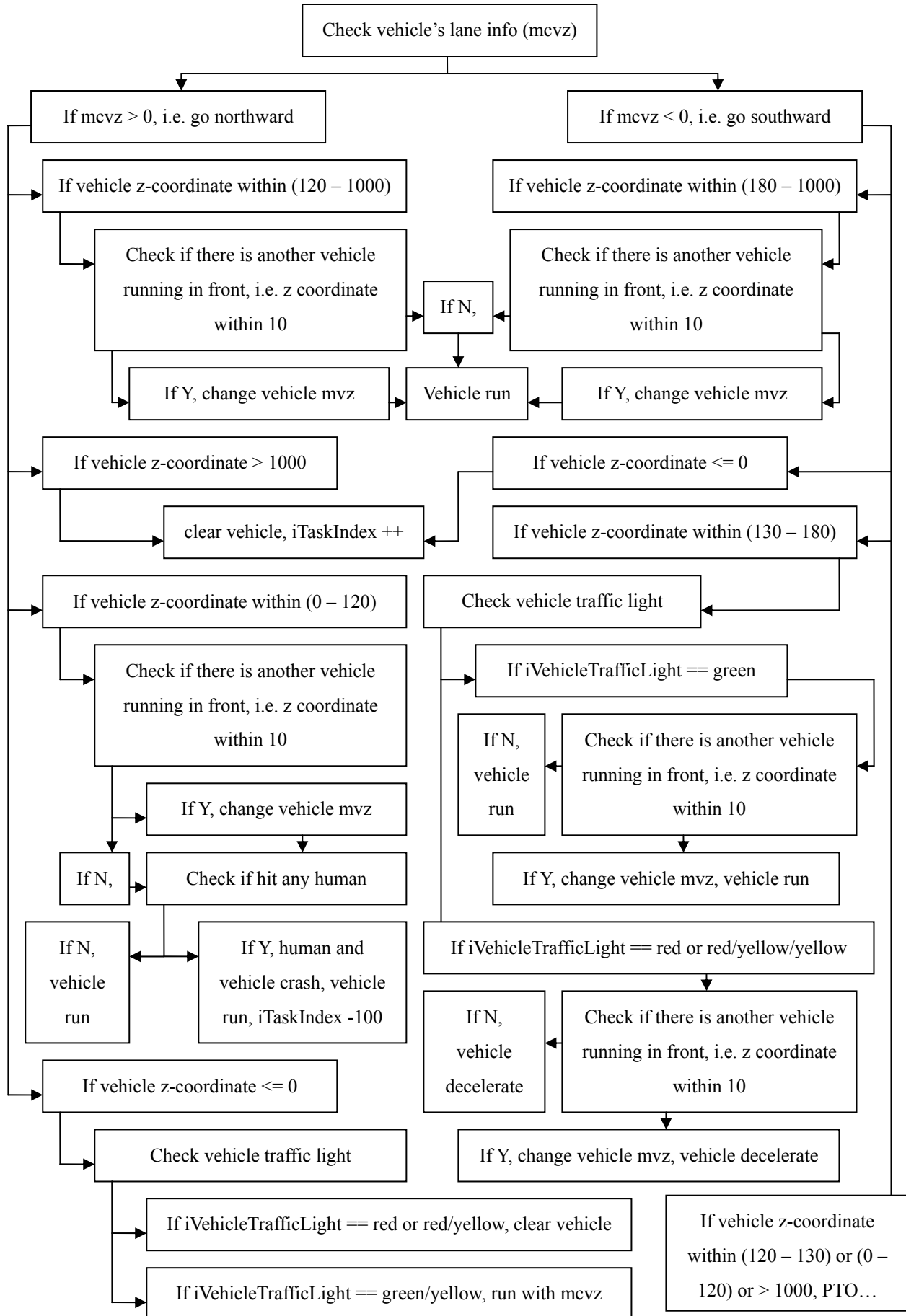
1. Creation of human objects and vehicle objects (according to time).
2. The collision between attack class object and human class object.
  - To check if attack object is valid and toward human objects
  - To check if a valid attack hits a human object
  - To modify indexes
  - To continue the artificial intelligence of attack object or to end its life
3. The collision between attack class object and surface object (mouse cursor).
  - To check if attack object is valid and toward player
  - To check if player dodge the attack in dodge mode
  - To modify indexes
  - To continue the artificial intelligence of attack object or to end its life
4. The artificial intelligence of attack class object
  - End its life if it was created for certain time duration
5. Retrieve Input
  - Retrieve keyboard, mouse and gamepad/joystick input, modify variables
6. Linkage of Input and human class object.
  - If in push mode and button pressed and cursor hits human object, make the human object pushed
7. Linkage of Input and surface object (mouse cursor) and attack class object.
  - If in fight mode and button pressed, create attack object in cursor location
  - If in dodge mode and button pressed, check if cursor hits attack object, and if yes, end the life of attack object
8. Reaction to traffic light of human object
  - Change the action of human object according to the traffic light and its own location
9. Reaction to traffic light of vehicle object
  - Change the action of vehicle object according to the traffic light and its own location
10. The collision between human objects and vehicle objects
11. The collision between vehicle objects and vehicle objects
12. The collision between human objects and human objects (actually there was no action designed in such circumstance)
13. Modify indexes

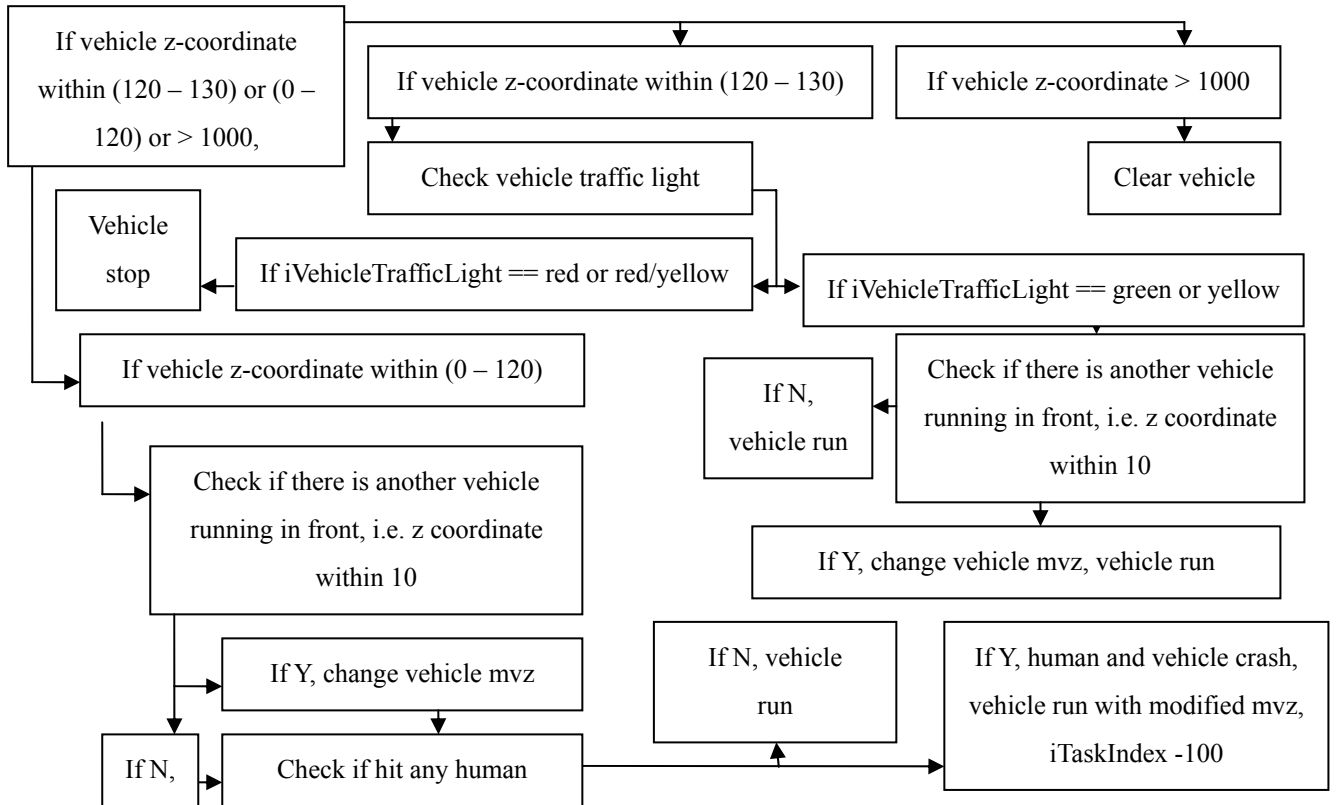
I hate to draw flow chart in document (it is such inconvenience to use Microsoft Word to draw a flow chart...), but it is necessary to draw the flow chart for some of the circumstances listed above, such as the treatment of player's input, objects' reaction to different traffic light signals. Charts are in the next page. In the beginning of programming stage, I omitted the attack (to player and to human class) function, and when I was integrating the attack function, I found

the analyzing sequence of the gameplay engine was problematic, so I changed the order of some steps and made the sequence as shown on last page.

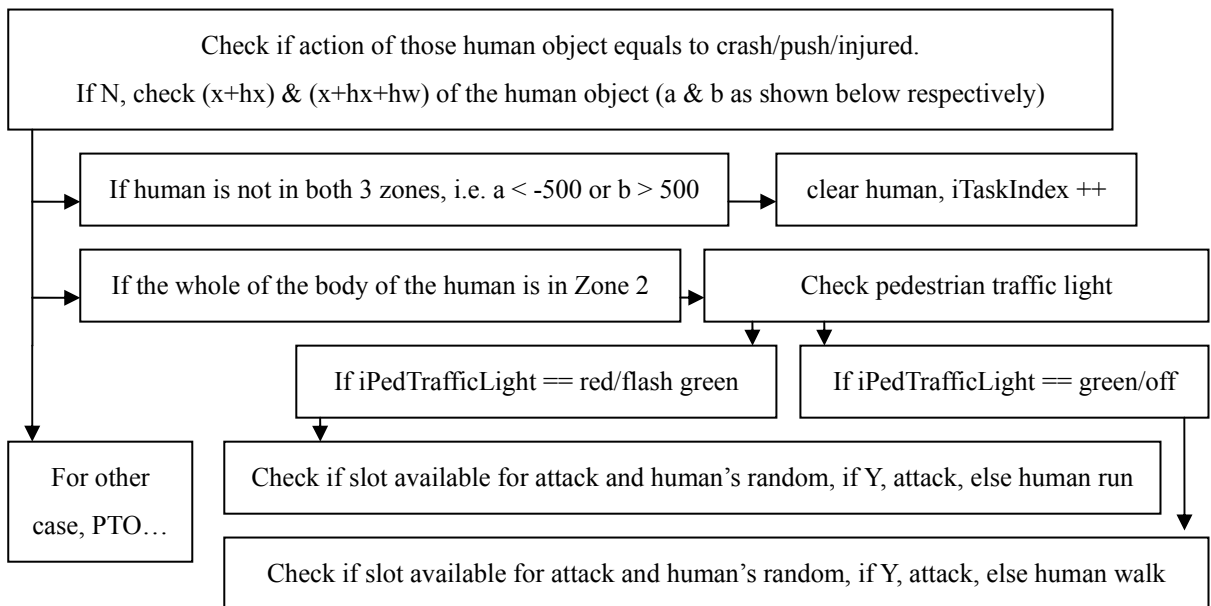
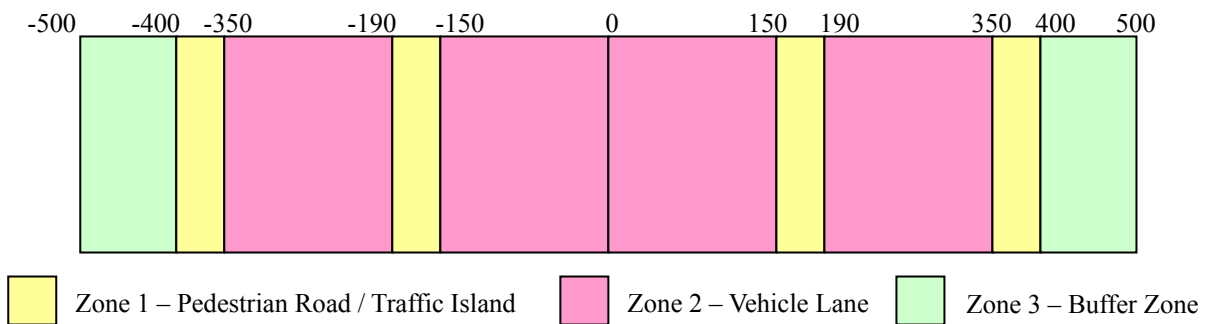


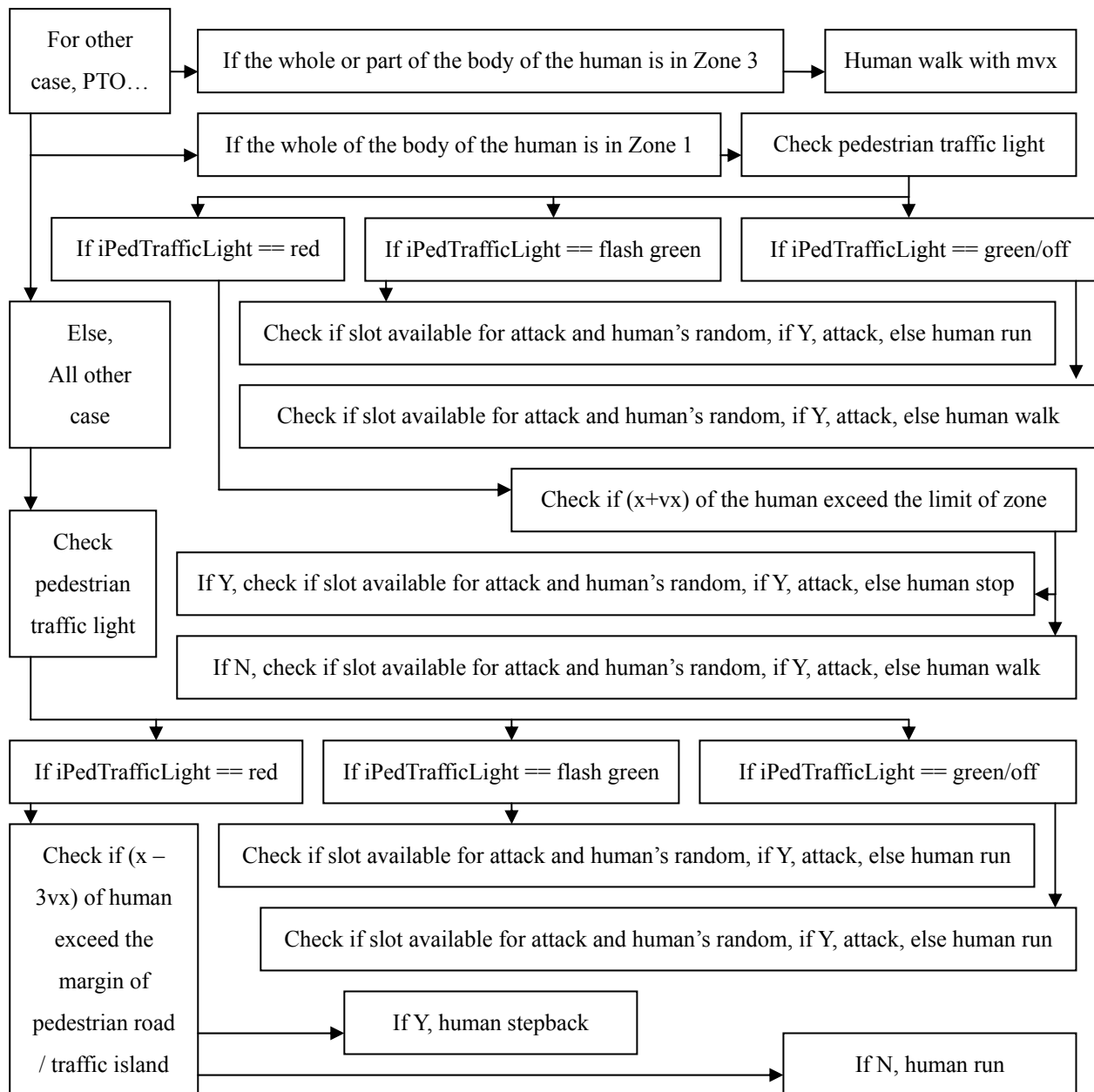
Reaction of vehicles to traffic light signals:





Reaction of human objects to traffic light signals:





Below is the list of actions the player can do during the game:

- Ignore the opponent
- Push the opponent
  - ➔ Use right click to select “Push” mode, then click left button on opponent
  - ⇒ Opponent being pushed with 2x forward speed
- Fight the opponent
  - ➔ Use right click to select “Fight” mode, then click left button on opponent
  - ⇒ Opponent being killed
- Dodge the attack of opponent
  - ➔ Use right click to select “Dodge” mode, then click left button on opponent’s attack
  - ⇒ Opponent’s attack being avoided

Here are the remarks for Indexes:

Time Index: 180s (count downward)

- Level ends when Time Index equals 0, Human and Vehicle objects clear, counts Mark Index.
- Reset on next level.

Life Index: 10000 (count downward to 0)

- When player is hit, Life Index minus 1000.
- When Life Index equals 0, counts Mark Index, Game Over.

Task Index: 0 (count upward to 10000)

- When one human cross the road, Task Index adds 100.
- When one vehicle/tram passes the road, Task Index adds 100.
- When one human being crashed by vehicle or killed incorrectly, Task Index minus 1000.
- When Task Index equals 10000, level ends, counts Mark Index.

Mark index:

- Counts on level end. Accumulates throughout levels.
- 1 Time Index lefts = 3 Mark Index
- 1 Life Index lefts = 5 Mark Index
- 

Another thing deserves to talk about is the definition of collision of objects in the game:

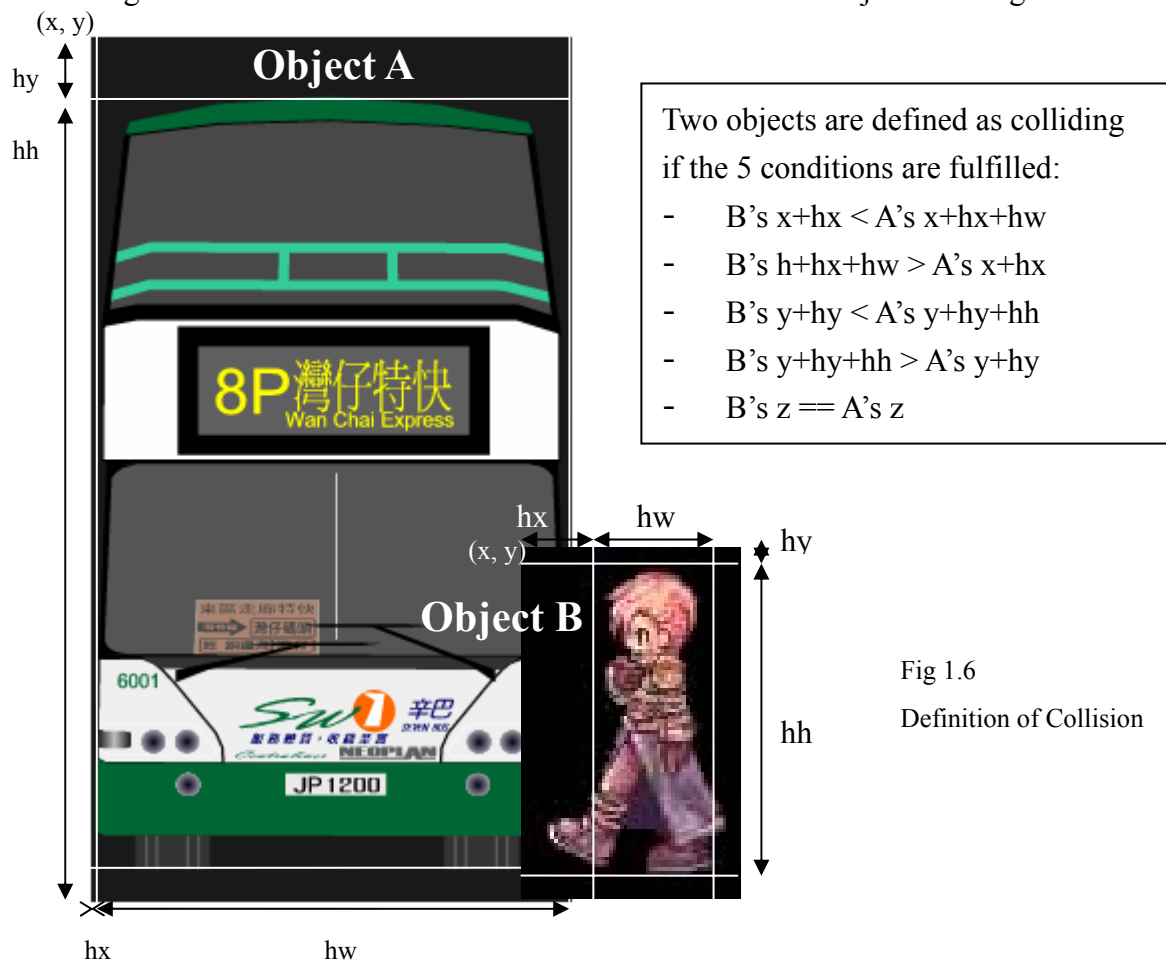


Fig 1.6  
Definition of Collision

There are some special remarks on the Attack object (both toward human and toward player):

- During the game only 1 attack toward human object and 3 attacks toward player simultaneously.
- An attack lasts for 1.2 seconds, only 1 to 1.2 seconds is the valid time for injuring the human objects / player...
- One attack hurt only 1 human once / hurt only player once.
- Player can only dodge attack in dodge mode.

So that's more or less how the game engine was. To certain extent it was not a complex engine and it could be made more complex. However I did not want to make it more complex in current stage, because the graphic and sound for the level were not yet ready that GeN and me was unable to see how was the effect of the both engine. Since it was not easy to debug without looking at the real output, I decided to stop here for this moment (April 16<sup>th</sup>), both to look deeply whether I made mistakes during my programming, and to wait for at least some graphic.

The last thing I did in the current stage was the arrangement of text output display:

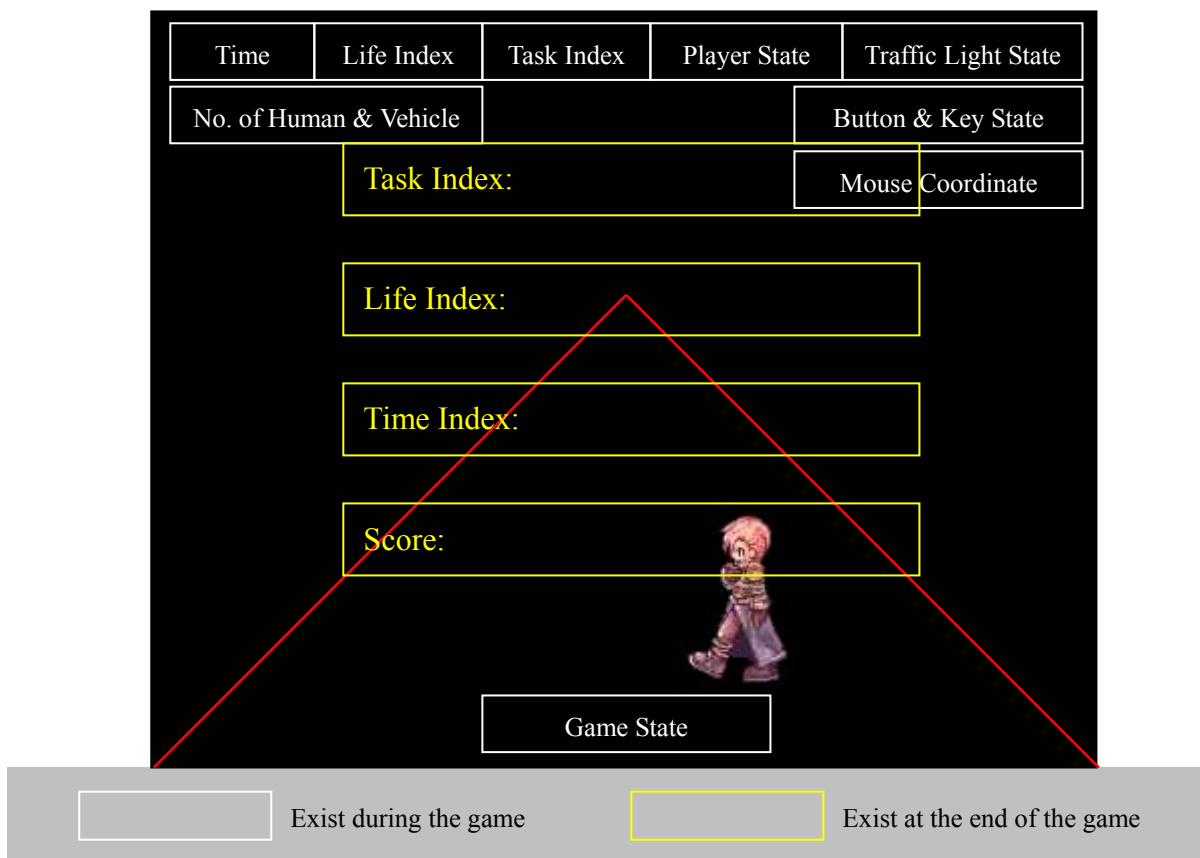


Fig 1.7 Text Output Display

To certain extent our working efficiency was low, anyway, my work stopped here before I wrote this document.

### In future (if any)

Before we are going to display this piece of final year project to audiences of our presentation, I may make some following refinement in the following aspect:

- The way of score counting at the end of level
- Joystick / Gamepad Input, may apply force feedback (and may not)
- Input selection (Keyboard verse Mouse verse Joystick / Gamepad)
- Multithread processing (this would be the least possible thing to be done as I did not see any possible application of multithread processing in most part of the program source code...)

(update on 23/04/2003)

Just received Bun's graphic on 22<sup>nd</sup> April, and discovered that the set of graphic has a different aspect, point of infinity and environment. I would like to wait till the drawing of decision of whether we would have to change the aspect/environment and do the programming task again tomorrow, when will be the presentation of final draft, although it seems that it would be a hard decision to make.

### Our future – May

(update on 03/05/2003)

According to our decision made on 24<sup>th</sup> April, after seeing our advisor professor Mike, we would keep using the aspect for the rest of our production.

During the period between the days 25<sup>th</sup> to 30<sup>th</sup> April Joystick/GamePad input was implemented as one input method in our game. I did also tried using lines of code for force feedback support in the game engine but even the lines were from Microsoft MSDN, no force feedback effect was generated during the game execution. The reason was still under estimation. Score output at the end of the level was reorganized to produce a "counting" effect, although it was just for fancy looking. Player should now be able to select the input during the game because of the reorganization of action-input mapping. I also tried to see if multithread can be used, but since gameplay and graphic engine was in two object class, I just did not want to make things complex in my stage and handed in the possibility of multithread processing to GeN.

Graphic was still under production on the writing day, and I was preparing the sound effects in our game, so that Bun could spend all his effort on making sprite's graphic and writing his own piece of research-theoretical text. (Phoebe's mail on 30<sup>th</sup> April mentioned that we have to submit our final piece to school general office on 6<sup>th</sup> of May, although we would still have a lot of modification on the game after that and before our final presentation, and although I still didn't think that we can hand in the true final piece to school on time just like most of our

schoolmates...) Selected sound would have a 44.1/22.05 kHz sampling rate and 16/8 bit sampling resolution with stereo output (according to guideline of GeN) and with Nintendo 16-bits gaming machine output characteristic (according to Bun's suggestion).

Yesterday GeN mentioned to me that he discovered some potential hazard in the graphic engine and the production of the engine based on the current one had to be stopped, and he did started the production of a new engine. He estimated the production of new engine to be finished by today or tomorrow. That seems a great deal of work.

(update on 05/05/2003)

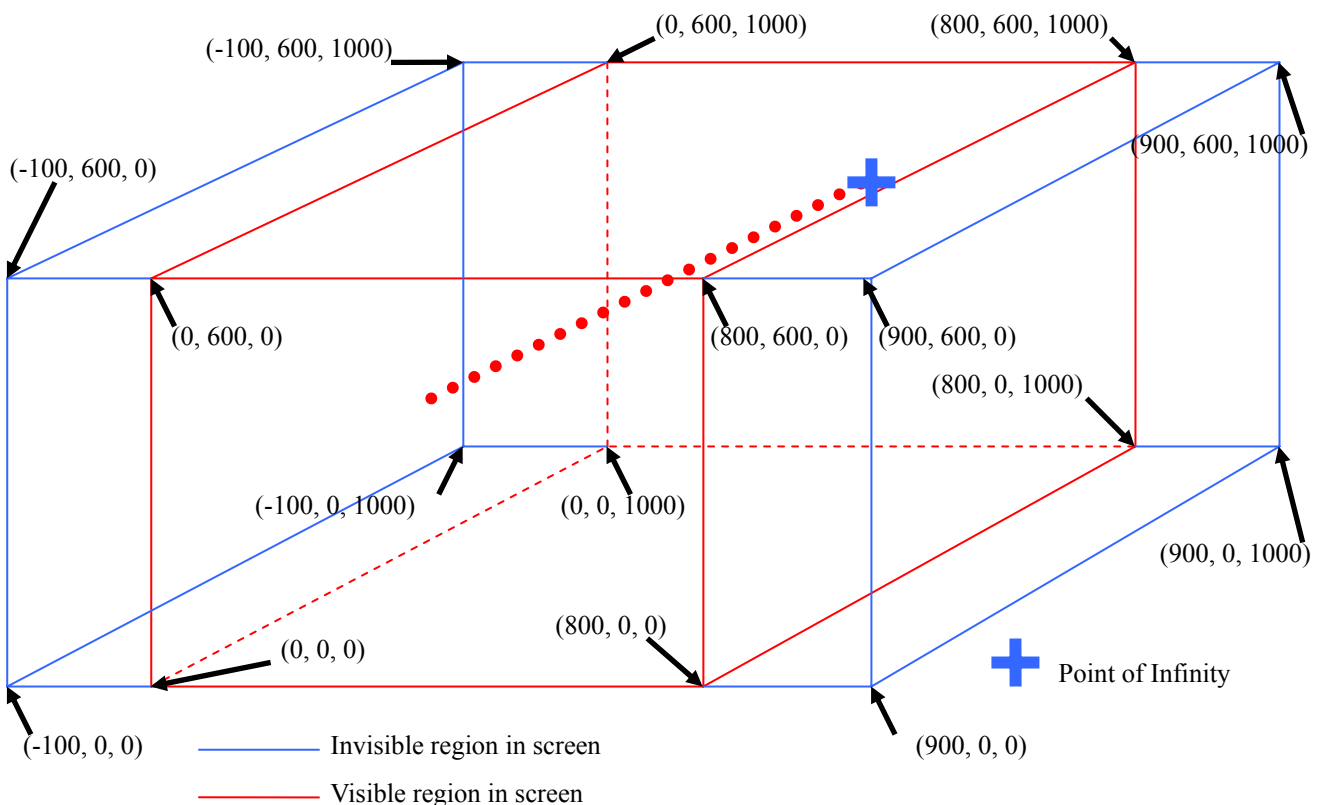


Fig 1.8 New 3D coordinates of the whole game environment

As the graphic engine has to be written again due to the hazard in the graphic engine, and several adjustments have to be made with the gameplay engine, so to make the process a little bit more convenient, the coordinate system was modified to something just like the figure above.

The new graphic engine provide a lot of fancy function like coordinate checking, speeding setting, setting visibility and 2D/3D sprite hittest checking (please refer to GeN's text if you want to know the detail of those...), so in the new gameplay engine it is not necessary to keep the coordinate of sprite. I don't need to build classes for each kind of sprite, and collision illustration stated in previous paragraphs was now invalid (I just don't know and don't care the hierarchy of hittest function provided in GeN's engine).

As the graphic for the game comes very very late (The sprite sheet of first vehicle object was still not ready up till the moment of writing this paragraph...), I decided to remove the attack object and that is related to attack of human object in the game. So there is no attack action for those pedestrian in the game, and hence the dodge player mode was also removed as it is now unnecessary. The fight action of player was to shoot or create fire toward pedestrian, now replaced by direct hitting toward human object. Just to hope that a reduction in graphic would make the task of sprite preparation more convenient and so we could receive graphic earlier, although the effect was not satisfactory.

Beside those mentioned above, the backbone of the gameplay engine was just the same to the one using with the old graphic engine, so the chart and diagrams are still applicable and not to repeat here.

## Achievement and Review

To certain extent writing a program (a “meaningful” program...) was far more complex than what I was imaging. The study and production of this game in this year was an invaluable experience to me. It seems that there will not be any final product at last, even when the time we have to do our presentation on May – we just do not have the tendency to finish at least a run-able prototype on that time. But being unable to finish the project does not mean that we learn nothing from the project.

We did not have a “concrete” producer for this project. Every task was finished just by the self-awareness of the team members. Our project advisors did show his influence on the project and fulfill their role (at least I think the actions of our advisors did show enough attention to this project...), however, we lack a strong leader (probably a producer or a director...) which shows enough mania on game production.

I did gain experiences on many aspects during the production. It was the first time for me to design and imagine a 3 dimensional environment without help of any 3D software, solely using pen and paper for design. I had thought of making a model of the environment, but it would be a waste of time as pen and paper just did the job nicely (at least thought-out the time I was programming the engine). I had to predict the action and movement of objects and see if there will be problems in the game engine, and if there is anything to remark I would record it down and be caution of those when doing programming. The experience would not be like this if we were doing a 3 dimensional game, as the objects would have different “thicknesses” and hence the environment could not be design only by hand drawing.

Although it was a 2.5 dimensional game, the environment of the game was not in 2.5 dimensional (there is not 2.5 dimensional environment in reality) and actually it was in 3 dimensional. When I was thinking of the interaction between input and the objects in the environment, I realized that I have to deal with the conversion between 2 dimensional (the input and output) and the 3 dimensional environment. That was a funny experience as I had to do calculation and make a formula form what we decided in previous game description. Here I do not want to repeat the graph. The thing was, I still think that the formula was odd, but I believed it and used up till now...(I probably need to construct a new one once a new infinity point is set.)

Just few days before writing this document I bought a force feedback gamepad. It was a product of Logitech, but when I was playing with force feedback function enabled, the effect was odd. I still think of whether applying force feedback control in the game or not, since the effect of force feedback was not really natural to and necessary for this game.

Before this production I had neither touch the language C++ nor the complier Microsoft Visual C++ 6.0 and .net (two versions make no much difference to our production...), so although it was a bit silly to say so, being able to use C++ and the complier to program was also one of my achievement. To certain extent it was a lot more neat and tidy to build class and function in Microsoft Visual Studio than Macromedia Flash, and the complier was much more intelligent.

Macromedia Flash is a great product; it provides a great platform for animation and application. Its drawback was unable to handle works which are too complicate, or it runs slowly. By making the gameplay engine and graphic engine ourselves did provide far more efficiency to us, it would utilize the processor to 100% (well, games in most genre also do), and by doing programming ourselves, we can use multithread processing to enhance the performance of the program. The only thing was whether we provide enough material or content to utilize such advantage.

This was the first time I work as a team for such long, and each of us would have much more opportunities in the future. I understand the reason why school discourages us to work in a team project, but I must say working as a team would have great differences compare to work individually, the stress confronted was different.

(update 05/05/03)

A more efficient way of communication would be suggested for game production. After presenting the first draft to Professor Mike, I do feel that the communication improved, and the effect shows directly in the last few days of production. Even on May we found that there is potential error in graphic engine and a great deal of modification need to be made in the

project, the programming part was in good progress (although I still don't know whether the game will be fine finally...) and took only around 72 hours to finish the whole thing...

Each of us was communicating with different language during production, and that caused a lot of misunderstandings, especially for the graphic production, in which a lot of mistake was made even during that last few days of production. I admit that we lack communication in early stage, and so this should be a remarkable lesson for this production.

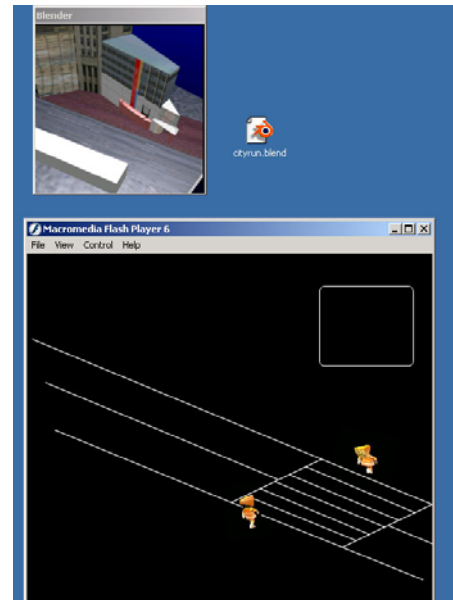
## Personal Feeling on the Project

(Written on 23<sup>rd</sup> April, 2003)

To certain extent, I am really angry and annoyed about the production of this project.

Yesterday (22<sup>nd</sup>) afternoon I received a phone from Bun, he said that he has uploaded two human sprites to his web server and let us to have a look. I went for the images and asked him if he want to change the screen aspect, or in other word, the vanish/infinity point, he said yes. When I asked him for the background image of the environment, he said it was not yet prepared. (A piece of image was prepared in the next day, 23<sup>rd</sup> April, to show us the aspect of camera and the environment. Image is as shown on the right.)

I really wanted to kill somebody at that time when I was listening to his phone; it was because that means the "effort" I did to design the environment in the past weeks became futility.



It seems that we would not be able to show any "draft" or something like that on 24<sup>th</sup> of April, it will even be hard to do so on the day we make our final presentation. I must say I should have to bear some responsibility for that.

To certain extent I must say that I have been spent too much time on waiting for graphic or discussion of the design of the environment. That had delayed my gameplay engine programming and hence I was already handing the initialized gameplay engine to GeN quite lately (on midnight of 16<sup>th</sup> April it was ready for GeN to download). I was in great pressure when I was designing the first edition of the environment, and I could imagine how GeN was feeling being cheated.

During the time when I was waiting for the graphic, I was not sitting there doing nothing. I was designing the environment according to the point of infinity in GeN's graphic engine prototype (issued on around 27<sup>th</sup> December, 2002).

During the design of the environment I was not in a freeway. Since the graphic was not ready, I had to consider whether the designed graphic later on can be applied to both engines, and that's the reason why the size of a human sprite in the environment stated above was in 60 (width) and 100 (height) (it was based on the initial graphic engine on 27<sup>th</sup> December, 2002), and all other objects' size in the environment were scaled base on the size of human object. Along the design period I tend not to design too many actions for each object and try to reduce the number of type of objects, in order not to place too much burden on my partners, so you may now discover that the number of actions for each object tends to be a bit low.

(This paragraph contains remarks to me in future that it should be something to be caution during game design. I am here not for framing up somebody counting for compassion from anyone or for a higher GPA, but I really want to record the stress I faced throughout the production on March and April...) I did not think that there was anything too hard for me to program or an overloading workload; rather, I felt the stress of doing this project was far greater than the workload. It was really a hard time that I cannot discuss with my team-mate to share the burden on environment design, and I was suffer from insomnia for around two weeks time on April, just because when I was sleeping, the figures in the environment would come out and occupy my brain...In the second week of April, when my progress was just a little bit more than in half way, since I was still unable to receive any graphic demo, I had thought of giving up and terminate the production at that time. On 12<sup>th</sup> April I decided to do in such way (It was a childish method, and I am childish, so it worked on me...): I went to Yahoo Auction and to place a bid on one thing I really long for, and made the appointment for the transaction on 17<sup>th</sup>, and promised not to let myself to attend the appointment if I was unable to at least handing in something to GeN to continue his part. Finally, the initial version gameplay engine was ready for GeN to download on midnight of 16<sup>th</sup>, and I am now the owner of that "gift to myself" (「櫻野みねね畫集-支天輪」, if you want to know the name, and I am sure 'I'(the me in future) want to know...). So, so....

On 22<sup>nd</sup> I was informed that the environment in the graphic illustrated would be different from the initial decision, so...I don't know what to say anymore, and just wait for the decision tomorrow, when we will see our advisors and discuss. Maybe I have to do the procedure I did in past two months again in the next few weeks, and, maybe I can have a good sleep tonight...

(update on 3<sup>rd</sup> May, 2003)

Production spirit was still not in good condition; however, we were trying to finish something mark-able on 6<sup>th</sup> May. GeN discovered serious problem in graphic engine, but he was working very hard on solving that and stated the day for finish of making the refinement. It seems there is still a great deal of workload in front...

I had never thought of giving up the production of this project during the past production. It might be a good suggestion, but it just didn't appear in my head during the last quarters of year.

When I was young, I often had to travel from Tuen Mun to Kowloon (I moved to Tuen Mun on 1986, at 4 years old). At that time, traffic was not as convenient and did not have much selection, only bus service was available. The bus at that time was in two axis, with front engine (Leyland Victory Mark II and Dennis Jublient was popular at that time) and had far weaker horse power (195bhp). During the Chinese New Year holiday, when people went home after visiting relatives, crowd appeared in bus terminus and every bus was overloaded. Bus was therefore could only climb over the endless slope in Tuen Mun Road (maybe called Tuen Mun Highway nowadays) as slow as human walking. Bus compartment was standing full of passengers with the howling engine of over 60°C temperature running at about 5 km/hr, that bus driver actually could park the bus at the road side and call for help from KMB, because such overload could cause bus overturn in such a bad designed road. However every bus drivers were keeping their patient and drove with the fan blowing hot air and revilement of passengers beside, finished the journey with more than 1 hour suffer, because they knew that at the time every bus were just as full, KMB was utilizing all its buses, no bus would have space to accommodate the overloaded passenger and every passengers were rushing to arrive home.

I kept continuing the production simply because of those scenes.

I don't know whether our production can finish on time or not, I am trying to see if I can help my teammate to solve their trouble, I just don't very care about the result of this project and my GPA of FYP course. I don't know why, but that's my life. Thank you for the cheering of our advisor, I may give up the production next time, but this time not before the deadline 6<sup>th</sup> of May.

(update on 6<sup>th</sup> May, 2003)

The final version of graphic engine was received around 22:30 of 5<sup>th</sup>, and the last piece of sprite (the first piece of vehicle object...) was around 03:15 today, and now is 11:00, fixing the last potential problem of hittest function.

The eye sockets look horrible black. I want to sleep. I have not sleep for 23 hours after the 5

hours sleep yesterday...(Just don't want to border you even more, paragraphs cut.)

Is this project a well constructed project/program? I won't justify, it is not my responsibility. In this moment, the only words I would say is: we did it...

## Project/Program Specification, Control Keys, Bugs Report

This is a not-too-shabby product, so please don't use a computer with shabby configuration to view this CD-Rom...Here is our recommendation for system requirement of the game:

### System Requirement –

- CPU: Intel Pentium III 800 Mhz or Compatible
- Memory: 128MB (actually the program consume over 90 MB during execution...)
- Storage: 170MB is required for execution in hard disk space, a CD-Rom
- Display: NVidia GeForce 2 or above/equivalent
- Sound: Support DirectX 8.1b Audio and Show
- Input: Keyboard and 3-buttons mouse  
Joystick required for the “Joypad” edition
- OS: Microsoft Windows 98 Second Edition / ME / 2000 / XP (Home/Pro)
- Other: DirectX 8.1b pre-installed

Just in case if you would like to open the source file and compile yourself, the following software is required:

- Microsoft Visual Studio.NET / Microsoft Visual C++ .NET
- DirectX 8.1b SDK for C++

I don't know how the graphic was prepared (just leave the explanation to Bun), the sound were being selected from sound library, and edited with Sonic Foundry Sound Forge 5.0. Sound clip are 16-bit depth, with stereo channel and sampling frequency 44100Hz (Sound source are included in my personal CD-Rom).

Control Key of the game is as follow:

Player's Reaction	Keyboard	Mouse	Joystick
Move Cursor	[Arrow Keys]	Axis	Axis set 1
Change Player State	[Left Shift]	Right Button	Button 2
Push / Attack	[Space]	Left Button	Button 1
Change Input Method (Keyboard / Mouse / Joystick)	[F10] – Keyboard	Middle Button	Button 3
	[F11] – Mouse		
	[F12] – Joystick		

So that is not complicating. The only thing which is complicating is debugging...

Bugs not yet solved / to be solved / may be able to solve:

- Cursor hittest checking to other types of 3D sprites is still now showing a very correct outcome. Just don't even know whether it is the problem of GeN's engine or gameplay engine.
- Objects scales are not in correct proportion. I cannot place the traffic light objects to the desire position, but that does not cause any error for the gameplay engine.
- For the "Joystick" version, it cannot be played without a joystick...
- Some unexpected error may show during the execution. By restarting the computer may solve some of the serious error.
- Attack action of human sprite was removed, so life index now becomes invalid and functions only at the end of the game, for score calculation.

I cannot think of any other anymore, my brain just stopped "rotating"...

Just not to border my advisor too much (the length of this text is already the half required length of a thesis writing final year project...), following is the last chapter.

### **File Indication of my own CD-Rom attached**

The following folders are inside my own CD-Rom:

- Document
  - Contains the document which I think is necessary during the production, so something which I think you should have a look (If you did not...)
- Project Folder
  - Contains the project folder of the previous version of gameplay engine, just excluding the most current and final version.
- Sound
  - Includes the sound for different events, including:
    - ◆ Background – Count Down
    - ◆ Background – During the game, background music
    - ◆ Background – Sound Effect at Game Ending
    - ◆ Human/Pedestrian – Crash
    - ◆ Human/Pedestrian – Injury
    - ◆ Human/Pedestrian – Push
    - ◆ Traffic Light – Red
    - ◆ Traffic Light – Green
    - ◆ Traffic Light – Off

So I think this is more or less the end of my text...