

Carleton University

COMP 4905 - Honours Project

An Intelligent Targeting System

By

Bryan Mann-Lewis

Supervised by

Dr. Michel Barbeau

School of Computer Science

August 14th, 2009

Abstract

This project is about designing a better targeting system in video games as the ones currently in place are extremely limited. Current targeting systems look purely at proximity and not at other factors that may affect the desired target for the player. First, a brief introduction to this project will be given. Next, a better look into targeting systems and their role in video games will be done. After this, the overall goal of the project will be explicitly stated, and the motivation behind the project will be given in full. Following this, the overall development cycle of the project will be laid out, from conception to independent testing. The results of this testing will be summarized and analyzed, and this will lead into the possible benefits this project could possess. Finally, the difficulties encountered will be covered followed by a conclusion on targeting systems in videogames as a whole.

Acknowledgements

Firstly, I would like to thank Dr. Michel Barbeau for taking the time to supervise this project. Secondly, I wish to thank all independent testers for their efforts and criticisms of the project. Thirdly, I wish to thank Nick Gravelyn for providing the XNA tutorial “Alien Aggressors”, which served as a code base for the project. I also wish to thank Alex Pass for the spaceship sprite, ‘missstrublingstorm’ for the art used for the menus and ‘xipx’ for the background art during game play. Finally, I wish to thank ‘The Prodigy’ for the song “Invaders Must Die”, which is used as the soundtrack to this game.

Table of Contents

List of Figures.....	5
Introduction.....	6
Targeting Systems.....	7
i. What is a Targeting System?.....	7
ii. A Very Brief History of Targeting Systems in Video Games.....	7
Project Goal.....	9
Project Development Cycle.....	11
i. Creating a Game.....	11
ii. Creating a Basic Targeting System.....	12
iii. Creating the Enemy A.I.....	13
iv. A More Advanced Targeting System.....	14
v. Player Interaction with the Targeting System.....	16
vi. Cosmetic Changes.....	17
vii. Independent Testing.....	18
End Results/Test Feedback.....	19
Difficulties Encountered.....	23
Possible Benefits.....	25
Conclusion.....	26
References.....	27
Appendix.....	28

List of Figures

Figure 1: A target indicator.....	12
Figure 2: The three enemy types.....	13
Figure 3: A screenshot of the final product.....	18

Introduction

This project was created with C# using the openly available XNA framework. It is a top down, third person perspective video game. For the player, the goal of this game is to eliminate all enemies on screen through progressively difficult levels. There are three different enemy types within the game: 'white', 'yellow' and 'red'. The white enemies move and fire in random directions and are the most common. The yellow enemies follow the player quickly and stop to fire in batches at the player. The red enemies follow the player slowly, do not stop to attack the player and each take 3 hits to kill. The player controls a spaceship and may use an auto-targeting system to help aid in the elimination of enemies and thus the completion of levels.

Both the enemy A.I. (Artificial Intelligence) and the targeting system will be discussed at length in this report, as these were the two main concepts behind the game design. First, however, it is important to understand what a targeting system actually is.

Targeting Systems

What is a Targeting System?

In a general sense, a targeting system is something that is designed to focus on a specific object. Targeting systems exist in many facets of the real world. For example, the military uses guidance - or targeting - systems for weapons in order to hit specific targets [Lockheed Martin, 2009], the medical world uses medicines or treatments that target and cure specific ailments in the human body [National Cancer Institute, 2004], etc. In the video game world, targeting systems exist in order to aid the player during game play by putting the focus onto a specific object or enemy within the game. From there, it is much easier to interact with that object or enemy as the players focus stays on it despite what is happening around them. This interaction could range anywhere from pressing a button to shooting at an enemy. It is these video game targeting systems that this project is based upon.

A Very Brief History of Targeting Systems in Video Games

Although a targeting system was not a new concept by 1996, no video game so heavily both relied on, or promoted one as much as “The Legend of Zelda: Ocarina of Time” for the Nintendo 64 [Gertsman, 1998]. It promoted a form of targeting known as “Z-

Targeting” that allowed the player to lock onto an enemy at the press of a button. At the time it was stated that “hitting Z while looking at an enemy will cause you to lock on to that opponent. From there you can circle-strafe around them, hop from side to side, and always block in their direction. This is key to fighting all but the most basic of enemies and is extremely well conceived.” [Gertsman, 1998]. Essentially, the closest enemy to the player, based on the player’s orientation, was selected by this targeting system and the player then had focus on that enemy. The player also had the ability to change targets if deemed necessary. This was felt as a great concept when first introduced and garnered much praise from reviewers [Game Rankings, 2009].

Over time, games have become more and more complex and as a result enemy behaviour has become more and more diverse; yet, this form of targeting did not. It has been copied by games time and time again but has never been improved upon [S. Ghazazani, personal communication, February 22, 2009]. This is where the problem lies. Why hasn’t the targeting system evolved beyond simply targeting the closest enemy to the player? It has always bothered some gamers [S. Ghazazani, personal communication, February 22, 2009] and thus the concept and goal behind this project came into fruition.

Project Goal

The problem with targeting systems within video games as they currently exist today is that they fall into one of the two following categories:

1. Target the closest enemy or object to the player in any direction.
2. Target the closest enemy or object to the player based on the direction the player is facing.

The goal of this project is to go beyond these two categories. There are many more considerations that must be made when targeting an enemy rather than simply where the enemy is in relation to the player. These considerations include:

1. What types of enemies are currently on screen?
2. Which ones are attacking the player?
3. Are there any special conditions on any of the enemies?

Taking all of these conditions into account, hopefully a better targeting system can be devised.

For example, say that there is a more powerful enemy in back of the current playing field that is repeatedly attacking the player from a distance, while the enemies closest to the player are doing nothing. Would it not then be more beneficial to have the targeting system automatically lock onto the enemy in the back? This would alleviate much frustration as the player would otherwise have to work their way to that enemy just to

ensure they are not killed by the lack of flexibility in the targeting system. It is such flexibility that this project hopes to achieve.

On top of the considerations previously mentioned, it is also important to realize that no system is infallible and thus it will make errors from time to time. Allowing the user to have direct control over the system to correct these errors, then, would also be beneficial. So if the user engages the targeting system and is not happy with their current target, they always have the option to switch to another one in the direction of their choice, systematically. Again, consider the above example of the attacking enemy in the background. If the targeting system targets a different enemy, then the player could simply change targets to the attacking enemy with a couple quick button presses, allowing them to cycle through all the enemies on screen. Ideally, the targeting system would take this change into account in the future and make further decisions geared more towards the player's desired target. In this way the system will begin to learn and adapt to an individual's playing style over the course of the game.

Taking all of this into consideration, this project will try to prove that such a targeting system can be made, effectively and efficiently.

Project Development Cycle

Creating a Game

Prior to the start of this project, research was done within the XNA community to see if a suitable game - one that would benefit from such a targeting system - existed and was freely available to use. No ideal solutions were found, so instead a popular XNA tutorial was used as a code base that a game would be built on top of [Gravelyn, 2008]. This tutorial handled the basic tasks of a video game: enemy movement, player control, multiple levels and player/enemy interaction. Beyond these concepts the game needed to allow the player and the enemies to have full three hundred and sixty degree movement within the playing field and a top down viewing perspective. This would allow for the enemies to theoretically move and attack the player from anywhere around them, while also allowing the player to see and move towards whatever they wish on the playing field. One can see then, that a targeting system - a system designed to allow the player to focus on one enemy at a time - would be very beneficial given the circumstances.

Ultimately it was decided that the game would be 2-D (two dimensional) as the top down perspective, functionally, would work the same for a 2-D or 3-D (three dimensional) game and a 2-D game consumes less resources than its 3-D counterpart, making it more modular. Player/enemy interaction was extended to both bullet collision and player to

enemy collision and the basic game structure that would utilize a targeting system was created.

Creating a Basic Targeting System

First, it is important for the player to know *which* enemy they are focusing on when they initiate the targeting system. For this reason, the first thing created for the targeting system was a visual indicator that would be placed over top of an enemy on screen.



Figure 1: A target indicator

The next step was getting this indicator to display itself at the press of a button. The 'Q' key was decided to toggle this image on the screen. When the user pressed this key, the indicator would be drawn to the screen over a specific enemy's position at every frame update, so it would follow and stay on the enemy. Player focus on this target indicator was then added in to allow the player's rotation to stay in line with the position of the target indicator so that the player was always looking directly at it.

Finally, a simple mathematical equation was used to determine what enemy the target indicator should be drawn over, based on their distance (d) from the player:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Where (x_1, y_1) is the players position and (x_2, y_2) is the enemies position. This equation was applied across all enemies currently on screen and the enemy with the shortest distance to the player was the one the target indicator was drawn on top of. Thus, a ‘target the closest enemy’ targeting system was created as the starting point.

Creating the Enemy A.I.

It was decided that in order to create a targeting system that would factor enemy behaviour into its selection, at least three different types of enemies should be made. Each enemy type would appear slightly different in both physical appearance and colour in order to give the player a visual indicator as to the types of enemies they are currently facing. The three types of enemies were ‘white’, ‘yellow’ and ‘red’. The white enemies moved around the playing field randomly and fire at random (although frequent) intervals in whatever direction they are currently facing. They take one hit (either from the player’s ship or the player’s bullets) to kill. The yellow enemies represent a bigger threat as they will follow the player; however, they will stop to fire at the player in random batches. They also take one hit to kill. Finally, the red enemies represent the largest threat for the player as they will follow the player (slowly), do not stop to attack the player and each take three hits to kill.



Figure 2: The three enemy types

These enemies spawn at different rates based on how difficult they are for the player to dispatch across the multiple game levels. For instance, the white enemies greatly outnumber the yellow enemies whom outnumber the red enemies. With three different enemy types now created, a more advanced targeting system can now be created to take enemy behaviour and threat into account.

A More Advanced Targeting System

Every enemy has 3 different threat values, along with an 'is attacking' variable. The threat values for each enemy are as follows:

1. Base Threat: This represents the minimum basic threat value of an enemy. White enemies have a base threat of 1, yellow enemies have a base threat of 2 and red enemies have a base threat of 4.
2. Change Threat: This value will be added to the base threat at any given time in order to assess the enemies overall threat. If the player manually changes targets, this value is altered to reflect that change.
3. Overall Threat: This value is the combination of the base threat and change threat. Things like how far the enemy is to a player and if they are attacking will also affect this value. It is this threat that the targeting system looks at when making comparisons between possible enemy selections.

Based on these three threat values, then, it is possible to devise a different threat for each different type of enemy, taking into account not only their behaviour, but their state (attacking, not attacking) and also their distance from the player.

The three final steps when creating this targeting system were to:

1. Assure that the enemies distance from the player affected the threat of that enemy.
2. If an enemy is attacking the player, then they should take precedence with the system.
3. Assure that the enemy is within the players 'field of vision'; in other words, assure that the enemy was in the general direction the player was facing when the targeting system is used.

For the first problem, a ratio is used based on the mathematical distance equation previously mentioned in order to calculate the threat an enemy would have based on this distance:

```
enemy.overallThreat += 500f /  
(float) (Math.Sqrt(Math.Pow((enemy.position.X - playerShip.position.X),  
2) + Math.Pow((enemy.position.Y - playerShip.position.Y), 2)));
```

Thus, the closer an enemy is to the player, the greater their overall threat will increase. A large number like 500 was chosen because of the fact that the x and y co-ordinates on the playing field could be in the high hundreds.

For the second problem, an 'is attacking' variable is used to signal when to add a high number of points to an enemy's overall threat. The final problem is addressed with a very simple solution: based on the direction the player is facing, create a very liberal 'box' that stretches indefinitely in the direction the player is facing. If an enemy is in that box, then consider it for targeting. Now a targeting system that takes into account enemy placement, type and state has been created.

Player Interaction with the Targeting System

The final step in creating this targeting system was allowing for the player to change targets at their will. It is not very flexible to assume that something will work correctly 100% of the time and thus there needs to be room for error built into the system. When the player has a lock on a target, the right or left arrow keys will allow the player to shift the target indicator to the next enemy to the left or right of the currently targeted enemy respectively. These changes raise a 'change target' flag in the game. This flag tells the game to pay attention to the next enemy the player kills. Based on the type of that enemy, increment a specific value related to that enemy by 1. At the beginning of each level, add these values to the appropriate enemy 'change threat' values so that the next time the player initiates the targeting system, it will be swayed more towards targeting a specific type of enemy.

For example, say the player initiates the targeting system and a red enemy is targeted. The player decides that they are unhappy with this selection and manually changes the target indicator over to a yellow enemy, and then kills the yellow enemy. This is

remembered and the yellow enemies will have their 'change threat' values increased to reflect this so that they stand a better chance of being chosen over a red enemy in the future.

In this sense, the targeting system not only allows for error, but *learns* from it. Over time, the more the player corrects the system to their liking, the more the system will adapt to their playing style, thus becoming not only a logically sound targeting system, but a personally sound one as well.

Cosmetic Changes

The last step before allowing for independent testing is a much more superficial one that is meant to purely benefit the player. This step relates to graphical and audio alterations to allow for a better looking, nicer sounding game. Nice, freely available background images were used for game menus and game play and a professional song was selected for the soundtrack. Beyond that, a simple, free to use spaceship sprite was used to represent the player and little explosions were drawn whenever an enemy is killed or the player ship is hit. Enemy sprites were made by hand, although heavy influence was drawn from the arcade game "Space Invaders" by Tomohiro Nishikado, and the font that the game used was changed to make it more pronounced. Finally, sound effects were altered or otherwise removed so the player would not be overwhelmed with various sounds.



Figure 3: A screenshot of the final product

Independent Testing

The last stage of the projects production is the independent testing stage. In order to gain a wide spread of perspectives on the game and its various facets, four diverse types of testers were selected:

1. A non-gamer: This person rarely, if ever, touches a video game. This will allow for a completely fresh take on the game play and allow them to play with no pre-conceived notions of how auto-targeting systems have worked in the past.

2. A casual gamer: This person plays video games from time to time, but nothing too complex or too involved. This will allow for a person with a familiarity to video games provide feedback on this system on a more theoretical level.
3. A new gamer: This person used to play games casually, but has recently started developing a habit for them. They have a fair amount of experience and thus their feedback on the targeting system will be more critical.
4. A serious gamer: This person has been playing video games for the majority of their life and thus will have played many games with targeting systems and have a strong base for comparison. Also they will likely be highly critical of the targeting system presented here.

With these four types, a wide array of viewpoints, expectations and experience will be gathered when the critiques of this targeting system are made.

End Results/Test Feedback

The game and targeting system has been completed as proposed. It takes into account enemy behaviour, position and player preference when making a choice. Independent testing went well across all testers. Each tester was asked a series of nine questions at the completion of their play time (see: Appendix - Survey Questions, pp.28-29). A summary of each tester's response to the game follows:

- Non-gamer: The non-gamer relied heavily on the targeting system as the game progressed as they had difficulty keeping up with all of the action taking place in the game. They never switched targets manually and didn't notice if it was targeting the biggest threat as they were "too busy trying not to die". They found the game to be stressful but were also happy with the way the targeting system worked for them.
- Casual gamer: The casual gamer felt that overall the targeting system "made the ability to hit targets a lot easier and without the need to fly all around to catch them". They began using it more and more frequently as the game progressed. They also never manually switched targets; however, they did actively note that the targeting system was targeting the largest threat for them as opposed to simply the closest enemy and felt that "...was a great idea. The closest one was not

always the most difficult to defeat”. They were also happy with the way the targeting system worked for them.

- New gamer: The new gamer played through the game many more times than the others. They noted that the targeting system “...made targeting enemies less frustrating, and I love that there was the freedom to change targets”. In this sense, the new gamer did feel the need to manually change targets occasionally as the game progressed. They also noted that after doing so, the targeting system seemed to adapt to their playing style and began targeting what they wanted to target more often. One problem they had with the game was that they did not want to have their ship looking in the general direction of the enemy to target them; they felt it should take every enemy on screen into account when making its decision. They noted that the system seemed to be targeting the largest threat for them quite frequently, but “for the times when it didn't work, I could effectively use the left and right arrow keys and switch the target to the other enemy I wanted to aim at”. They were also happy with the way the targeting system worked for them.
- Serious gamer: The serious gamer took less time to play the game, but reached a further point in it than all other testers. Regarding the targeting system, they “felt that overall it was done well. It targeted the units that conveyed the most threat to my ship”. They came to rely more heavily on the targeting system as the game progressed. At first, they would manually switch targets quite often; however, “...afterwards I found that I ended up [switching] it less and less”. They took note

that the system seemed to adapt itself to their target switching and as the game went on, began targeting what they wanted to target right off the bat. Like the new gamer, they also thought that that the targeting system should take every enemy on screen into account when making its choice rather than the enemies in the direction the player was facing. They took note of the fact that the targeting system was selecting the largest threat on the screen and felt that “it was great!” Finally, comparing it to other games of this nature, they stated that they “wish other games used this type of system”.

It is evident, then, that the targeting system worked well for a wide array of players. It is not perfect, but those that felt the need to manually switch targets did notice the system learning from them as the game progressed. Every tester appreciated the fact that it would target on a threat level vs. a proximity level and the fact that they could manually switch targets at any time in case the system was wrong in its initial assumption.

Difficulties Encountered

By far the largest difficulty encountered with this project was the fact that there was no free to use, open source XNA game done in a style that would benefit from such a targeting system. A tutorial was found that allowed a player controlled sprite to move left and right and fire up, and had a collection of enemy sprites on the screen that also moved from left to right and fired down. It had the basic code for menu control and level transition and that was it. Everything else in the game had to be hand coded. On top of that the sounds and images used in the tutorial were of poor quality and had to be replaced.

The second largest difficulty encountered was that of enemy behaviour. Three different enemy A.I.'s not only had to be made, but done in a way so that they don't interfere with each other. Just having the yellow enemy stop when it fires, for example, took a while to code in. A health system had to be added across all enemy types to allow for red enemies to take three hits before dying.

The third largest difficulty encountered was collision detection for player to enemy ships. If the player occupied the same space as an enemy, that enemy would be killed and the player would also have to take a hit. This was difficult to get working as the base code had a couple 'hit' variables for the ships and figuring out which one meant what, on top of altering it without affecting legacy code, took a long time to get working.

Finally, an effective threat based system for the targeting system to use was difficult as it relied heavily on complex mathematical equations and ratios. Much research was done into the field of vector mathematics before a suitable method was found and utilized.

Possible Benefits

Based on the test feedback, it is clear to see that this system not only goes beyond simply targeting the nearest enemy on screen, but also does so in a satisfactory way for the player. This system could be extrapolated for use in 3-D games as well, and a threat based targeting system is superior to a purely proximity based one. If such a system were to be implemented into future games, a lot of frustration would be removed for many potential players. Many video game players that were interviewed during the course of this project all felt that there were many games that would cause you to die not because of player skill limitations, but because of targeting limitations. This project was done to prove that the limitations around a targeting system can be greatly reduced if someone is simply willing to take the *time* to do so. In the future, hopefully such a targeting system will come into fruition and become popular, as the current ones in place now, according to the players themselves, are not good enough.

Conclusion

As game complexity grows, so too must the various facets around these games. Targeting systems have been a part of video games for a very long time, and have yet to really evolve into anything beyond targeting the closest enemy to the player. Yes, this would keep complexity down, but at what cost? Many players have become frustrated with this type of targeting and feel it is time for something better. This project, this ‘intelligent targeting system’ proves that there is a better solution out there for game players. A wide array of game players were used to test this theory out and they all agreed that it works well. It is not perfect, but the players used to test this system feel that it is a better solution than what is currently available. This and its ability to learn from its own mistakes, to adapt to the players playing style, makes this system incredibly versatile. Every tester stated that the targeting system “worked for them”. This is a very positive result as it means that not only has this system succeeded on a technical level, but on a personal level as well. This project, while on a very small scale compared to most commercial video games, has proved to be a success.

References

Game Rankings. *The Legend of Zelda: Ocarina of Time*. Retrieved July 31, 2009 from <http://www.gamerankings.com/n64/197771-the-legend-of-zelda-ocarina-of-time/articles.html>.

Gertsman, Jeff (November 23, 1998). *The Legend of Zelda: Ocarina of Time Review*. from http://www.gamespot.com/n64/adventure/legendofzeldaoot/review.html?om_act=content&om_clk=gssummary&tag=summary;read-review.

Gravelyn, Nick, (April 22, 2008). *Alien Aggressors: The Making of a Complete Game with XNA Game Studio 2.0*. from http://www.ziggyware.com/readarticle.php?article_id=170.

Lockheed Martin Corporation. *F-35 Lightning II Electro-optical Targeting System*. Retrieved July 31, 2009, from <http://www.lockheedmartin.com/products/F-35LightningIIEOTS/index.html>.

National Cancer Institute (August 25 2004). *Radiation Therapy for Cancer: Questions and Answers*. from <http://www.cancer.gov/CANCERTOPICS/FACTSHEET/THERAPY/RADIATION>.

XNA Creators Club. *XNA Creators Club Online*. Retrieved July 31, 2009 from <http://creators.xna.com/en-US/>.

Appendix

Survey Questions

1. Overall, how did you feel about the auto-targeting system?
2. Did you come to rely on it as the game progressed?
3. How often did you need to manually switch targets?
4. If you did manually switch targets multiple times, did you notice the targeting system leaning more towards the enemy you would change targets to?
5. Are there any improvements you would suggest for the targeting system?
6. Did you notice any bugs or oddities with the targeting system?
7. Do you feel that this targeting system has worked better for you than those of other third person perspective games?
8. Did you notice the targeting system was trying to target the biggest threat as opposed to the closest enemy? What did you think of that?

9. Any other comments about the game?