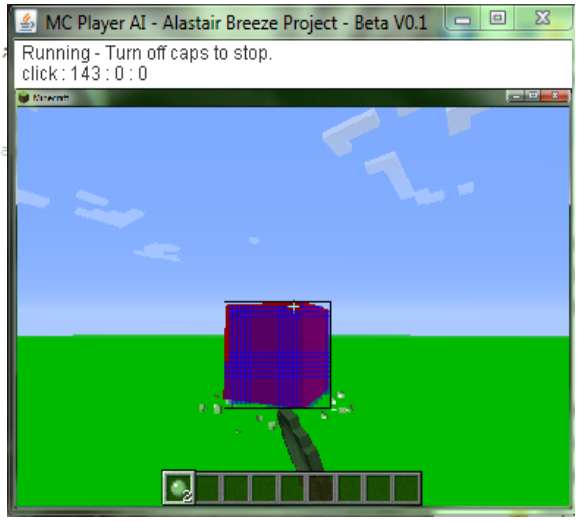
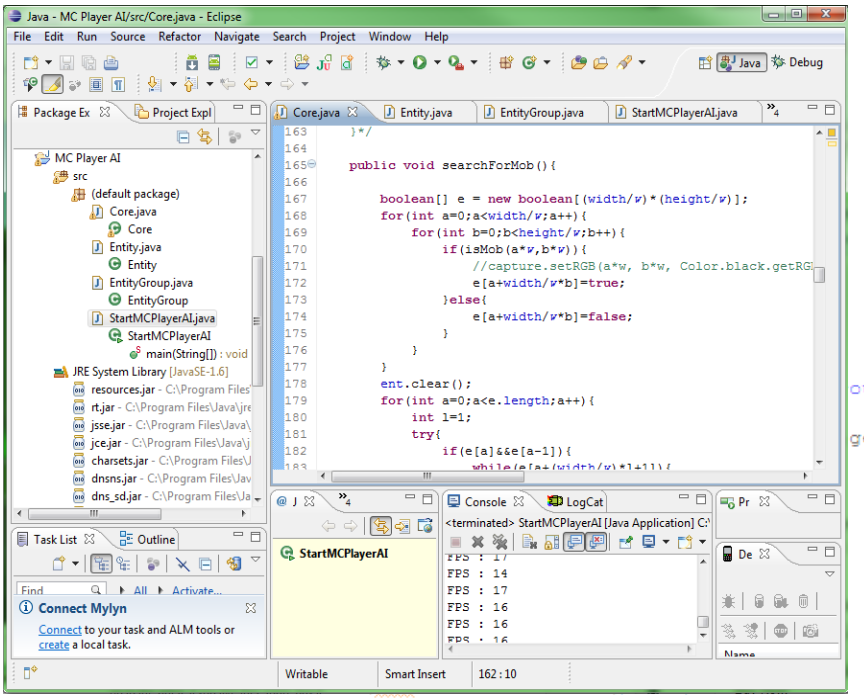


The Advancement of Artificial Intelligence (AI) In Games And Creating My Own Example

By Alastair Breeze



Contents

Origins Of AI

The fascination around Artificial Intelligence (AI) has grown within computer scientists since some of the earliest computer systems. The term originated from a famous computer scientist called John McCarthy, where he defined it as, “the science and engineering of making intelligent machines.”[1] A simplistic explanation is using computer systems to solve problems that would usually need human intelligence to solve. Many people assume AI is simply robots simulating a human completing their day to day tasks, a stereotype projected from many films, however in reality AI is a broad field of computer science that integrates seamlessly all around us, such as monitoring a room's atmospheric information and controlling the heating and air intake accordingly, much like the thought process of a human controlling the heating, but implemented in a computer system. I wish to look into extending my knowledge of AI due to it's boundless opportunities which it unlocks in computer science. Furthermore I wish to broaden my skills in complex problem solving, with the aid of technology and mathematics.

AI In Games

Since some of the earliest computer games, AI has been used as a means to challenge the player and/or to immerse the player in the realistic experience. AI has also led to the opportunity for games to be played by a person on their own, more commonly known as 'single player', because the human can now challenge the AI.

A famous example of an early computer game that utilises AI is PAC Man. A game consisting of the player being chased by enemies, whilst trying to collect points. Without the AI, the enemies would simply navigate randomly, furthermore making them less likely to find the human player. Lacking a challenge for the human. However the AI was developed in a way to make the enemy travel in the direction of the human player, leading to a more challenging experience.

Another good early example of a game where AI is used is Pong. In this game the AI allows a player to play on their own and still be challenged. When playing, the AI moves the paddle in the direction of the ball, hoping to then clash into the ball and not loose. Despite the method being extremely simple, it still makes for a good challenge for the human player. Being tough to beat can subconsciously lead the player into game theory thought processes, whereby the player thinks of a tactical system to try beat this AI. An example could be leading the human player to aim at the walls to spring the ball off and confuse and make the ball pass the AI's bat to score. Complex thought processes like this challenge the player, helping to make the game more interesting, and without doubt aiding it to become the success that it was.

In newer games, AI has advanced enormously from the early examples. A fantastic example of this is Sims 3. In the Sims, a person is given the control of a family that are set against the challenge of life. The AI simulates life and the random events that may occur, ranging from being fired from a job, to feeling emotions and getting angry. All of these artificial elements help to immerse the player into a lifelike experience.

Inspiration struck at home when I was working on modifying a pre-existing game 'Minecraft' and I happened to be looking through some videos about the process. I stumbled across a short video of the creator of the game 'Minecraft', known as Notch, showing himself adding some more complex AI to his game 'Minecraft'[2]. In the video he decides to add AI to the skeletons which use bows and arrows, and he made their targeting more accurate. This would be great as it would make skeletons more of a challenge to compete against in the game. The AI he added worked by predicting the humans game play movements to estimate where the player will be when the arrow is in their proximity. Eg if the player is moving to the left, the skeleton will now aim more to the left which will hopefully collide with the player by the time the arrow reaches them. In this example, Notch made a very tough piece of AI code that made it very difficult to attack the enemy in the game. In all, this AI is a very good at adding a challenge to the game.

Inspiration from this made me want to create my own AI for a game, but a bigger and more complex piece of AI that will, rather than challenge the player, be the player; a challenge that will require a large amount of analysis into realising how a human actually plays a computer game.

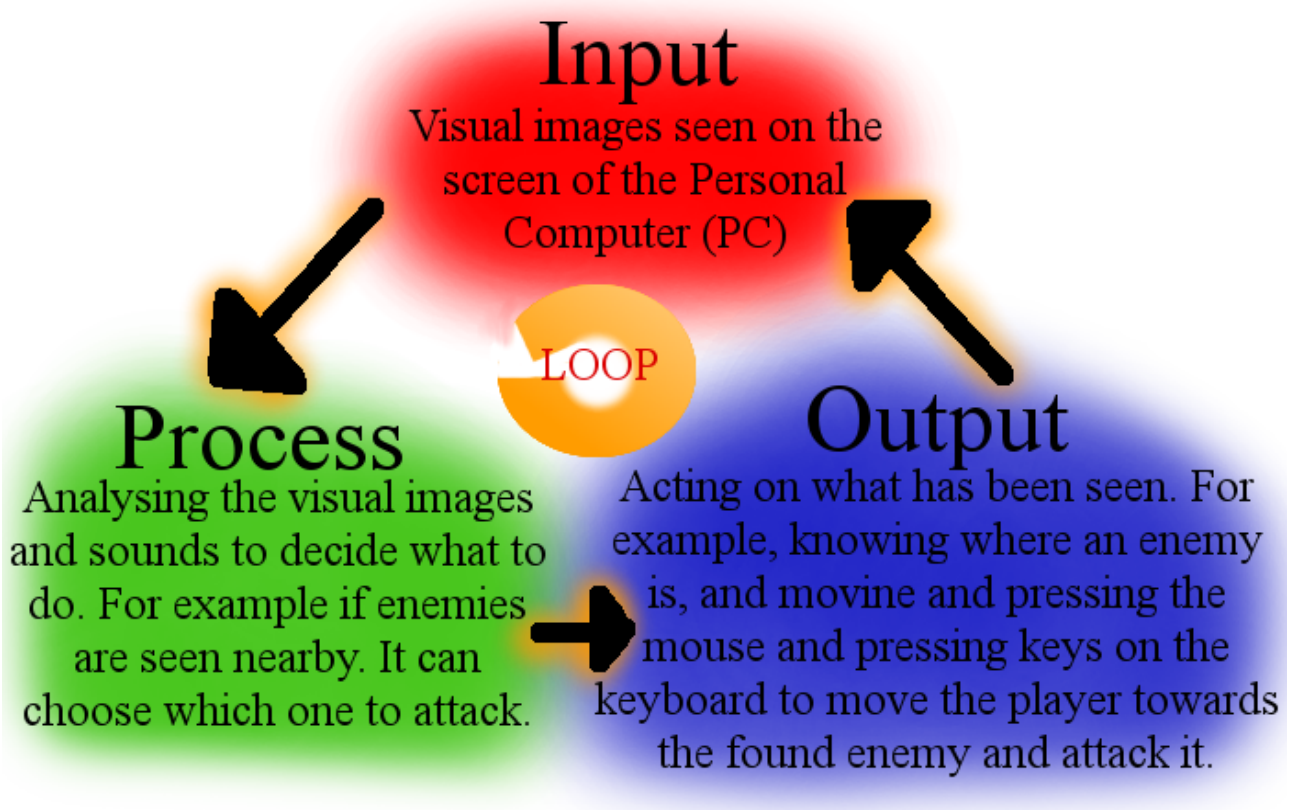
Designing My Own Project

Following my fresh knowledge in the field of AI in games, I set out to try and build my own AI. The challenge was to create an AI software that plays a game like a human would. The AI would then would take over the human and be competing against the game on the behalf of the human. I chose the game 'Minecraft' as the game is a sandbox, meaning it can be played in almost any way you wish, ranging from building something to, in this instance, simply attacking any nearby enemies to survive, a challenging objective for the AI to accomplish.

First I decided that it would be essential that my AI does not hack/change any code of the game that I would be competing against. For example, I cannot change any pre-existing game code to power my AI, such as finding out the Cartesian X,Y,Z coordinates of an enemy. This information could, for example, be processed to calculate the exact trajectory for a bow and arrow. Pinpoint accuracy could be achieved from this, but this would be something that I perceive as cheating because this would give a big advantage to the AI compared to a human player. The main reason I decided not to hack was because I wanted it to be like a human, and therefore the AI would only have the same input and output that as a human player. The AI will therefore be completely separate from the pre-existing game, and run as a separate piece of software.

I decided to program the AI in Java, as it is the language in which I have good experience. Java also has capabilities for all the tasks I will require it to use. A good example is the Robot class built into Java which allows me to simulate tasks, like moving the mouse to allow the AI to take control of the game. This class also allows me to be able to screen capture the images of the screen which will come in handy for the AI to view the game. Another benefit of Java is that the language is cross platform and therefore my AI would work on nearly any modern Personal Computer (PC).

Following my research into AI, I have realised that I need to begin by turning playing the game as a human into a step by step system. I will then be able to program my AI software to accomplish the system, in order to play the game like a human would. Here is a simplified system that a human follows when playing the game:



Feedback :

The human may give intelligence derived from previous actions or processes that will influence future decisions.

I hoped to create an AI software that could accomplish all this. It will be following the same principal steps that the above human system follows when playing the game.

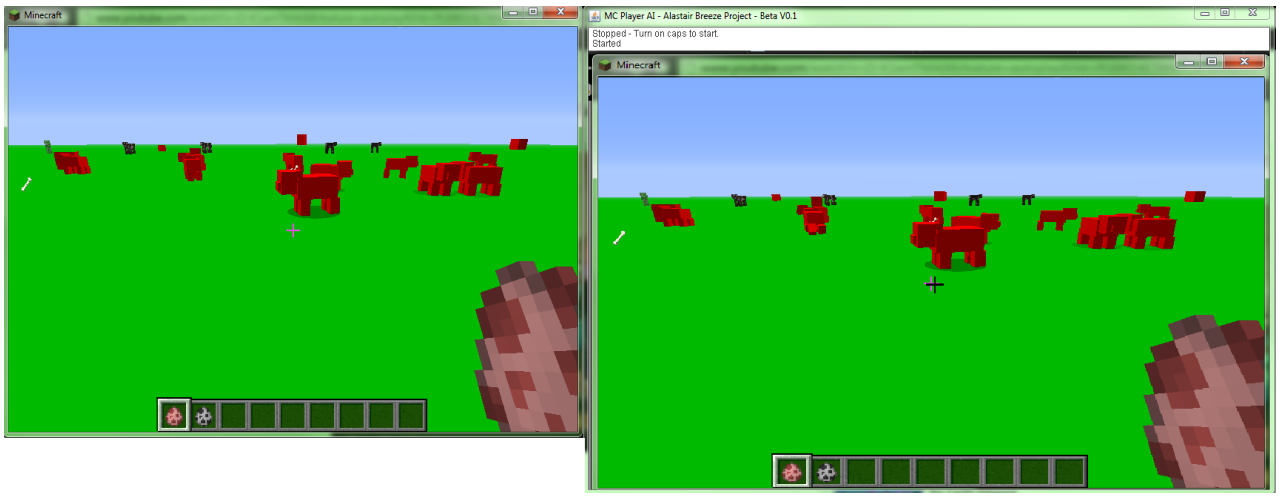
Ways that the AI will differ however, is how it will not be using any sound input in order to play the game. Sound effects could assist to alert the AI to where the monsters are, for example hearing a Skeleton monster. However the sounds will give negligible benefits, as there would be other sound effects which may influence the AI falsely. For example, just walking in game will create sounds that may confuse the AI, therefore it would be more accurate to just use visual input.

Another variation is the way that the AI will not use feedback. The reason for this is that it will be more accurate to simply scan the scene each time with no influence from the past. A way to understand this is, if an enemy is scanned it will not memorize this for future, instead it will have to re-detect the enemy in future. Altogether this will cause the AI to be able to react immediately to any sudden changes in the game, and not be stuck on previous thoughts. For example, if an enemy suddenly appears, the AI could choose to act on the new enemy immediately without being influenced by the previous target.

Converting the Human System Into An AI Program

Input :
Capturing the images of the game is the way the input for the AI works. It does this by the programmer allocating a rectangle on screen it wishes to screen capture which is where the game window is. The AI then caches this data to be stored for later processing as a 2D image. It is a 2D image because the output of a conventional screen is 2D images. This is where the data is temporarily stored to be analysed and used. Here is an example of the AI capturing the data :

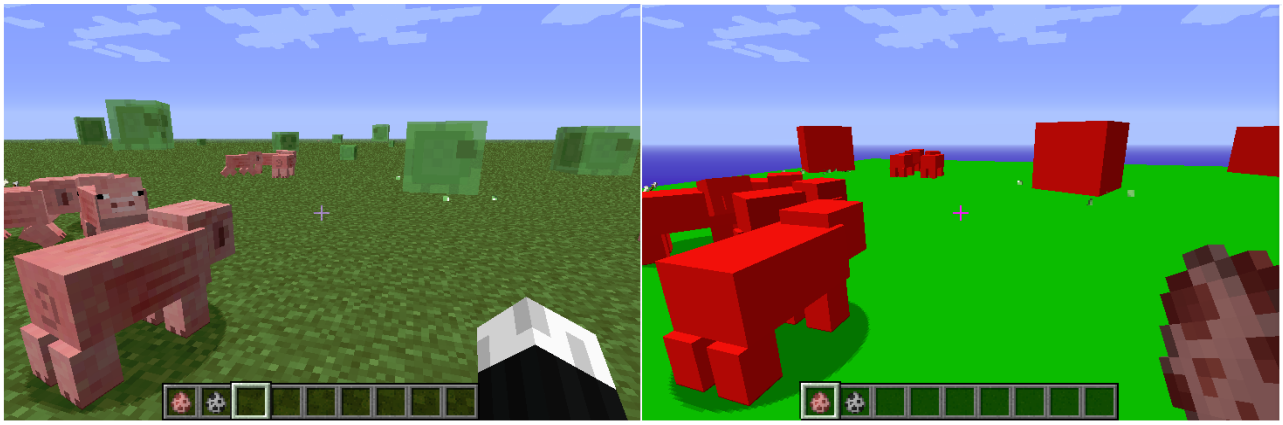
The Game Running | **The AI running beside it and screen capturing the game**



As you can see the pre-existing game is running as a window on the left, and on the right you can see the AI program running its own window simultaneously. The AI program is screen capturing the game image that is being shown on the game window on the left, and then displaying it inside its own window.

It is quite apparent the graphics of the game have been simplified. This has been done by adding a texture pack of my own that converts the textures of the game into simpler more primary colours. For example the ground is green, sky is blue and enemies are shown as red. This will be helpful later on when processing the pictures. Here is an example of what the texture pack changes :

The Default Texture Pack **My Simplified Texture Pack**



Above you can see 2 pictures of the game with and without a texture pack. On the left the game is using the default texture pack. However on the right the game is using my own simplified texture pack, using clearer primary colours, yet still looking realistic and just as playable as with the normal texture pack. Clear primary colours help the AI to detect more accurately.

Now that the AI has a cached image of what is being shown on the game, the AI can then process this data. But before this, the program condenses the amount of image data that is being used. This must be done because even if only a small 2D area of the screen is being screen captured, it can still be a lot of data to process. Like a small screen area of dimension 600 by 800 pixels is 480000 pixels of data to be processed, and even with some of the most powerful PC's around, this would significantly decrease the speed of the AI which would directly affect the speed of the AI's decisions, making it less effective at its task. The way I got around this problem was that I cut down the amount of pixels by only using specific pixels on the 2D image by condensing it. I do this by only using every 6^{th} x pixel and every 6^{th} y pixel resulting in less pixels to process:

As you can see there are many black squares showing which pixels are being used. This will then in fact cut down the amount of pixels down 36 times going from 480000 to about 13333 pixels. This will greatly improve performance.

Processing:

Now that the AI program has captured the screen and cut down the amount it needs to process, it can now process the pixel data to decide how to act.

It does this by individually going through each of the remaining pixels splitting the colour down. To do this I can get the colour value of a pixel, which is usually 32 bits of data. A bit is a value that is either 1 or 0, and so this means that there are 32 1's or 0's defining the colour of a pixel. This can in fact create up to 4294967296 different colours. If we leave the colour of a pixel in binary form we get what's called a long group of 1's or 0's like this random example:

10110010 00101011 00101001 01000011

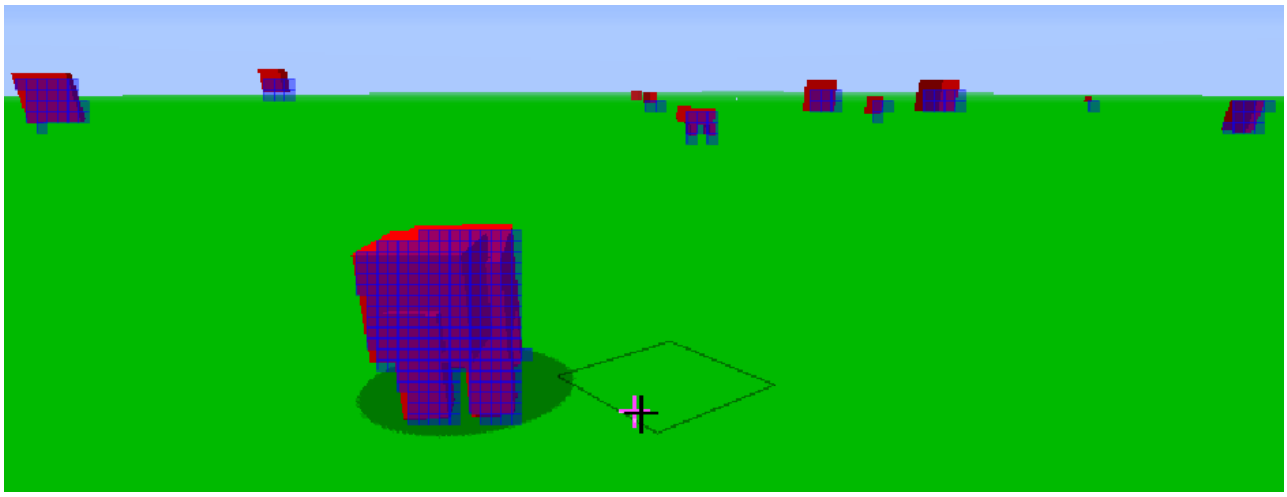
If we now split this example into groups of 8 we get :

The Binary Group	10110010	OO101011	OO101001	O1000011
This value corresponds to the information	(alpha)	(red)	(green)	(blue)
The binary turned into decimal	178	43	41	67

From splitting the binary up and converting it to decimal, I now have alpha, red, green and blue values, each of the values ranging from 0 to 255. To convert a binary number into a decimal number I must turn each of the 1's or 0's into a value which can then be added together to make the decimal number :

Digit Number	1st	2nd	3rd	4th	5th	6th	7th	8th	
Digit represents	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Value	1	0	1	1	0	0	1	0	TOTAL
Multiplied together	128	0	32	16	0	0	2	0	178

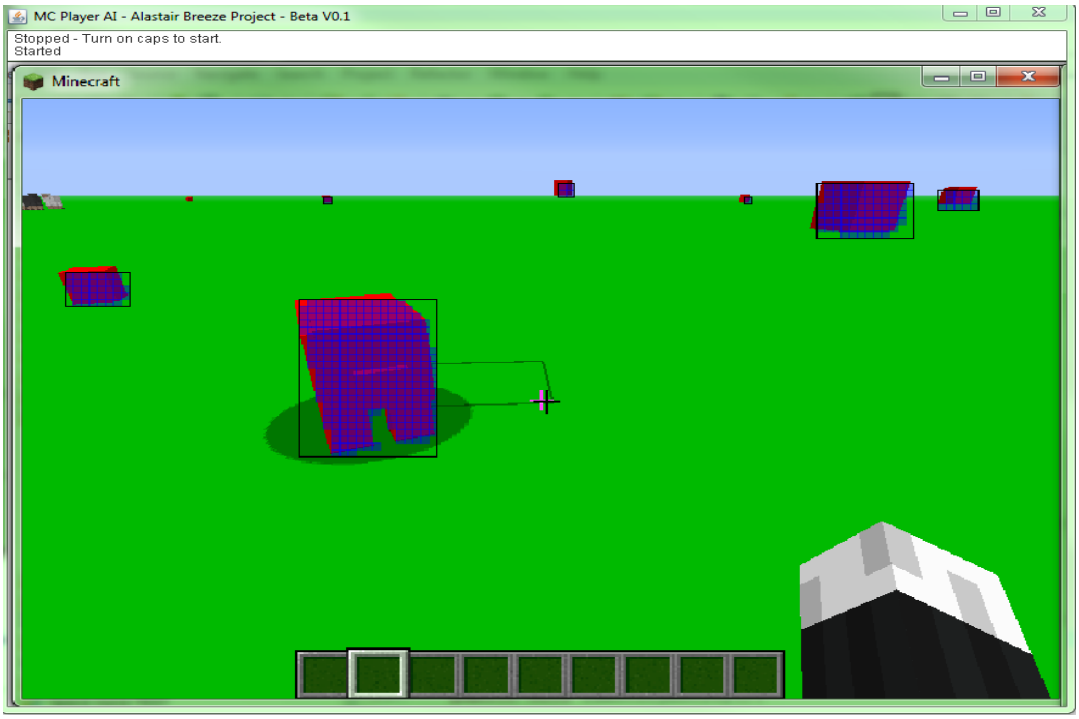
With these colour values, I can use simple inequalities to work out a discrete colour of each pixel such as deciding that a pixel has a high red value and low green and blue value, which deducts that the pixel can be classed as red. Classifications assist deciding if the pixel is enemy, ground or sky. In this instance, it is red so an enemy. Ground and sky values would be green and blue and so can now be discarded; however, the enemies are kept for further analysis. The following image shows defined enemy pixels on screen as squares and layered with a translucent blue square to inform the programmer of the detected enemy pixels:



This screen shot is of the AI program now recognising the enemies that it may need to act on. You can see the blue squares are the pixels where enemies are detected. As you can see the enemies are being layered over with blue squares. Here the AI is being very accurate at spotting and defining the enemies on screen.

Thousands of enemy pixels are being recognised, and so it is hard for the AI to yet decide what to do. Therefore it needs to group any of the pixels that are part of the same enemy into 1 corresponding enemy. So now, if any of the enemy pixels are close together, they are then grouped into full enemies. This is repeated until all pixels are grouped. This is very processor intense, however, the AI is able to cope at a good speed thanks to the condensing of pixels. Accuracy here however is not compromised as the enemies are nearly always grouped into correct single enemies.

Here is an image of what the software is seeing now that it has grouped the enemies :



Now the software is drawing black rectangles around the grouped enemies. This shows that the detected enemy pixels have been joined into simple single enemies that can now be analysed further.

A list of enemies has now been created, the next task is deciding if any are posing a danger and need to be attacked. To do so it first calculates an area value for each enemy by treating it as a rectangle and multiplying width by the height. The area is an important value as it helps estimate how far away the enemies are by assuming that all enemies are similar in size. From this assumption closer enemies would appear larger and far away enemies smaller. Slight error can arise with varying sizes of enemy, however, it is accurate enough to give a reliable approximation. Distinguishing the largest enemy to travel towards and attack is the next objective. This method of sorting works well as it does usually find the closest enemy, however it is flawed because the AI will assume that very small enemies are far away and therefore not a threat. However, in most cases, the smaller enemies that will be ignored are not a very dangerous enemy to the user, so not too detrimental to the effectiveness of the AI. In fact in some cases, a larger more dangerous enemy that is far away from the enemy becomes the attention of the AI instead of the closer, smaller, and less dangerous enemy. Thus the AI is prioritizing the more dangerous enemy. I could try to improve this in future is by trying to calculating the distance of an enemy by using vector movements as a way to get a more accurate portrayal of how close and large an enemy is. However this would be a huge amount of work for something that would likely barely improve the effectiveness of the AI, and would also make assumptions about the velocities of different enemies.

Predicting what the enemy is was a feature that I added to the AI. For example working out whether it was a harmless animal like a pig, or a dangerous enemy that could kill the player. This would be a very valuable piece of information in deciding which enemy to attack, as it could prioritize more dangerous enemies. I attempted to do this by working out the shape of the enemy. This is because usually the dangerous enemies like zombies are shaped tall and therefore have taller height than width. Almost square objects could be classed as slime enemies, and wide and not very tall enemies could be classed as animals. Aspect ratios of the detected rectangle of the enemy made it possible by dividing the width by the height. This ratio would then tell me whether the rectangle was tall or wide or square. This however was not accurate enough to make reliable decisions because enemies being viewed at different angles often distorted the classification. Therefore, in my opinion, the decisions weren't worth taking into account.

Output :

Once the AI has decided the enemy it wishes to attack, the AI then can act by moving the mouse, clicking or pressing keys on the keyboard. If however it can't find anything to attack, it will simply walk forward by pressing the W key.

Humans playing the game would react when it sees an enemy to attack by moving the mouse to aim the cross in the

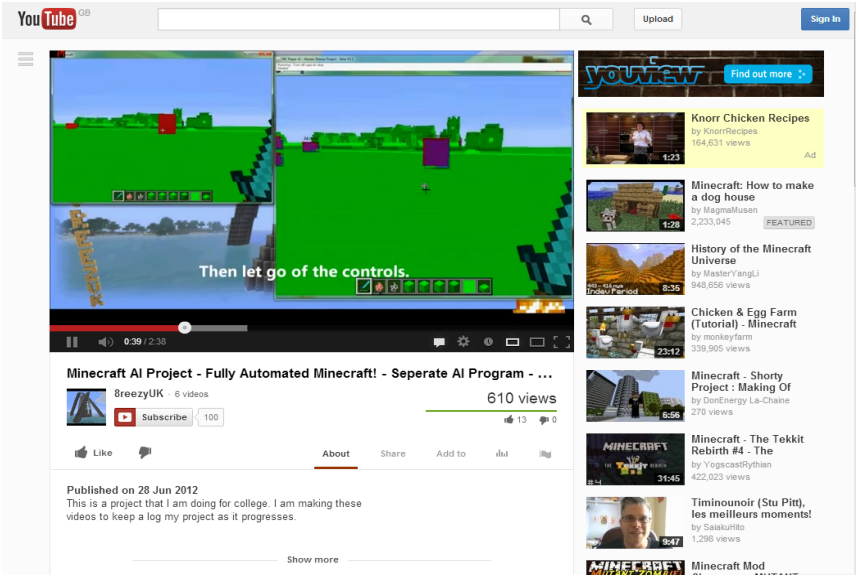
middle of the screen. Once the cross is over the enemy and the enemy is close enough, it can click to attack it. So first of all, the AI moves the cursor a fraction towards the centre coordinates of the enemy, slowly aiming the player. If the pixel over the centre of the screen is identified to be an enemy (i.e. red) and the enemy is large enough to be close, the left mouse button is then clicked to attack the enemy. Whilst this is happening the W key on the keyboard is also being pressed by the AI in order to move the player towards the enemy. All of the commands are made possible by the Robot class built into Java, which allows the AI program to simulate making inputs with the mouse or keyboard as if a human were doing it.

Altogether, all of this then results in an AI that can simulate a human playing the game. This input process, output and feedback is then repeated until the AI program is closed, repeating at about 30 times a second. The result is a very powerful piece of AI with extremely quick reactions. Almost certainly quicker than that of a human as it takes the AI on average about 33 milliseconds because 1 second divided into 30 repeats is about 33 milliseconds, a very fast reaction time. This also results in a fast twitch like movement of the aimer in the game due to the AI, as it rapidly moves the mouse.

Analysis Of My AI

Overall the AI works very well, as it does manage to kill many enemies quickly and efficiently, and is a lot better than some humans would be at playing the game thanks to its 'lightning' reactions. So in all, the AI is successful as it does simulate a human playing the game very well.

During the process of keeping a log of my progress, I published a video of the AI software.[3] Response from the internet has been very positive, one of which said, “fake.” [3] At first this may seem like a negative piece of feedback, however this is the best compliment achievable to the AI. This is because I can deduce that the statement is implying that the person thinks that it is not AI controlling the game, they think its a human faking the process leading me to infer the AI is so realistic in emulating a human, it appears to the commenter as one. This slightly relates to the Turing test from Alan Turing's 1950 article Computing Machinery and Intelligence where he discussed, “can machines think.” [4] In this article, Turing explains how for a machine to be intelligent, it can be tested using tests such as, “The Imitation Game,” where basically the machine appears human to an observer. [4] He then goes on to explain tests in which this can occur. However as it can be inferred that it appears human to an observer, it therefore comes across as intelligent. I had many other comments about the AI all of which were very encouraging, with people looking forward to its future development of the AI. Some people wanting to download the AI software asking, “could you put up a download so we can try it out.”[3]



(For Comments See: Source 1)

One of the most encouraging comments is one which says “This is very cool. And so the day begins when server owners have to start watching for bots. I both fear for what will come of this project and am very interested to see what will happen.” This comment implies that the person sees the potential of the AI, however they fear a day where the game is populated by AI software along with human players.[5] (See : Source 2)

Future development could include continuing the AI to make it more intelligent. For example, allowing it to decide routes towards enemies if there are dangerous terrain hazards around. I would also look towards further optimizing the code in order to improve the reaction time.

A method that I would improve the detection system is a field of computer science known as machine learning. This

field is a common area of AI in that it may allow an AI software to be able to learn and become more effective in its job from experience. An example of this could be in my AI program where the AI would store data about the enemies it sees, for example if an enemy kills the AI's player in the game. The AI will be aware of this (this is a human condition), and when this type of enemy is detected again, the AI will be more aware of it and avoid it. The difficult part however is trying to detect the type of enemies in an accurate way. If however the AI detects wrongly, a human like the programmer could correct the AI to learn for future reference. This detection system would be a great asset to the AI and its thought processes, however the accuracy of it would be weak and the implementation would be a colossal task to undertake for it to be accurate. Also this system would likely require huge processing requirements, which in turn could result in a slower less effective AI.

From what I have learned about AI, it is evident to me that, as we progress with technology, AI will become a more important integrated part of our life. Furthermore with the constant advances in the power and size of computational systems, AI will become a more feasible and effective solution for an increasing number of tasks. AI in games particularly shows the vast leaps we are making and, similar to my project, how AI can complete a task with the capabilities of being faster than a human, as in my example where the reaction speed of the AI is arguably faster than an average human player. Hopefully what we can learn from AI in games, as well as improving games, can help to push us forward in the development, or even inspire a generation to want to build AI for the future. Finally, I feel that the project has helped to really prove the possibilities of what can be done by almost anybody with freely available tools and some time, and look forward to the future and what can become of AI.

References :

[1] McCarthy, J (1959?) “The science and engineering of making intelligent machines” [Internet], Stanford University.
Available at : <http://www-formal.stanford.edu/jmc/whatisai/node1.html> [Accessed 13 December 2012]

[2] 2PlayerProductions (2011) “Coding with Notch” [Internet Video], Youtube.
Available at : <http://www.youtube.com/watch?v=BES9EKK4Aw4> [Accessed 13 December 2012]

[3] Breeze, A (2012) See source 1 for complete references [Internet Video], Youtube.
Available at : <http://www.youtube.com/watch?v=QWXQH2DqY44> [Accessed 14 December 2012]

[4] Turing, A (1950) “Can machines think,” “The imitation game” [Internet], UMBC University.
Available at : <http://www.csee.umbc.edu/courses/471/papers/turing.pdf> [Accessed 16 December 2012]

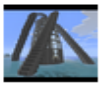
[5] bioemerl (2012) See source 2 for reference [Internet], Reddit.
Available at : <http://www.reddit.com/r/Minecraft/comments/xzqd1/afk_minecraft_my_minecraft_ai_project/>
[Accessed 16 December 2012]

Extra Resources :

Source 1 :

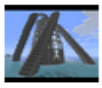
Comments from 8reezyUK are me : (Alastair Breeze)

abrachoo 5 months ago
This is awesome!!!!
but then again... so is everything you post.
Reply · 2

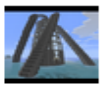


8reezyUK 5 months ago
haha, thankyou :')
Reply · in reply to abrachoo

davdua1991 5 months ago
fake
Reply ·



8reezyUK 5 months ago
how could it be fake? look on the right hand side of the screen, you can see entities being scanned. That is about 90% of my code working right there, and your looking right at it, adding the AI movement is the easy part once you know where they are on screen. Please tell me what you find fake about it.
Reply · in reply to davdua1991



8reezyUK 5 months ago in playlist Uploaded videos
Tbh, now that i think about it, the fact you don't think its real AI and that its a human controlling it is actually the best compliment you could have game me :) it proves that my project is believably human, so thankyou :D
Reply · 4 · in reply to davdua1991

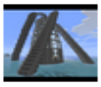
bluetiger811 4 months ago in playlist Uploaded videos
Best. Comeback. Ever.
Reply · 3 · in reply to 8reezyUK

TheMonkeyWhacker 5 months ago
This is amazing. Also, could this AI (or a variation of it) be given to mobs, like the rave people in the rave mod you made?
Reply ·

8reezyUK 5 months ago
sadly not,this AI is not built into MC,its a seperate program that works just like the way we humans play the game,by looking at the screen and then moving the cursor and pressing keys accordingly,so i couldnt add it to the code of the ravers. I could however seperately add complex AI to my mod if people wanted

Also,I could one day do an experiment where I get a bunch of people on a server and they could all run my AI program and we could have the rave mod on and people would go mad and fight :)
Reply · in reply to TheMonkeyWhacker

biomerl 1 month ago in playlist More videos from 8reezyUK
Has this been worked on recently, is there any hope of a minecraft AI release?
Reply ·



8reezyUK 1 month ago in playlist Uploaded videos
Heyy, theres been a little work, but nothing revolutionary yet :/ and i really wanna release it, but at the moment its a project thats part of a pretty big college qualification im doing, so am not allowed to release it until the qualification done i think :/ sorry, wish i could tho :(
Reply · in reply to biomerl

biomerl 1 month ago
Hey, its all good.

I would just hate to see this die. I would love to see more videos of it though, even if its nothing evolutionary.
Reply · in reply to 8reezyUK

nedless shep 1 week ago
could you put up a download so we can try this out
Reply ·