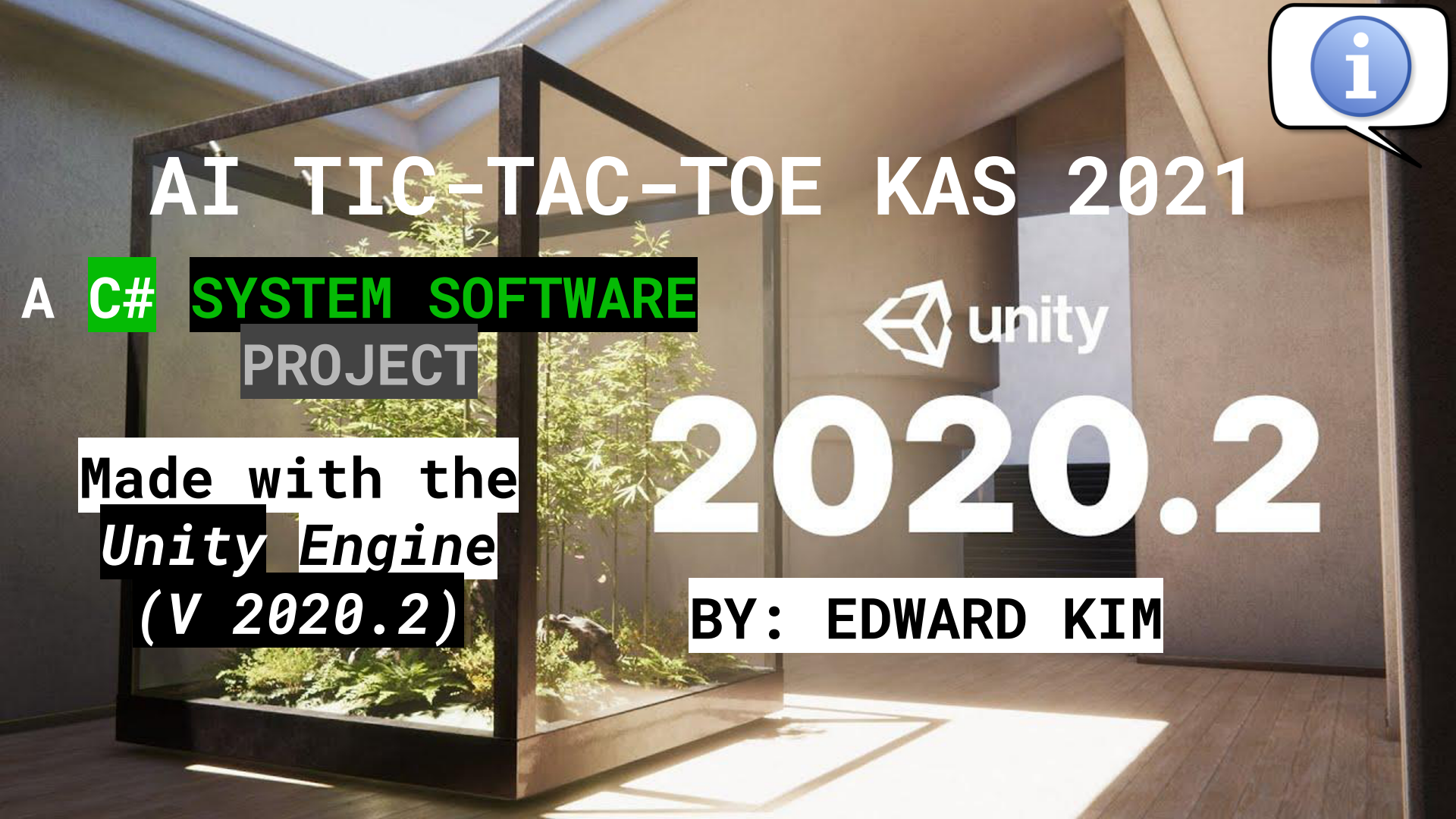

AI TIC-TAC-TOE KAS 2021
A C# SYSTEM SOFTWARE PROJECT
Made with the Unity Engine (V 2020.2)
unity
2020.2
BY: EDWARD KIM

Abstract

My research question was which type of general competitive AI, which I assigned as ASAI (attack strategy AI), DSAI (defense strategy AI), and AaDSAI (attack and defense strategy AI) was the most efficient. Though most of AI you think “learns”, my level of AI is very simple and requires no training, with just simple C# algorithms. Although this might not be as progressive, this is the first fundamental step in order to think about beyond logic and think about machine learning. This research can have various societal impact such as increasing the success chance of a human through business and society in careers (like chess where it increases a human's critical thinking and learning). Or it can even be a virtual learning friend or coach that can train students to learn how to do an adversarial task, where it goes against the learner. And, in future generations, people can then extensively develop the virtual learning buddy/ teacher to adapt to the learner’s capabilities and weak/strong areas of knowledge and challenge the learners on their weak spots, then teach them how to do things on those weak spots based on their "attack"/"defense" strategy knowledge.

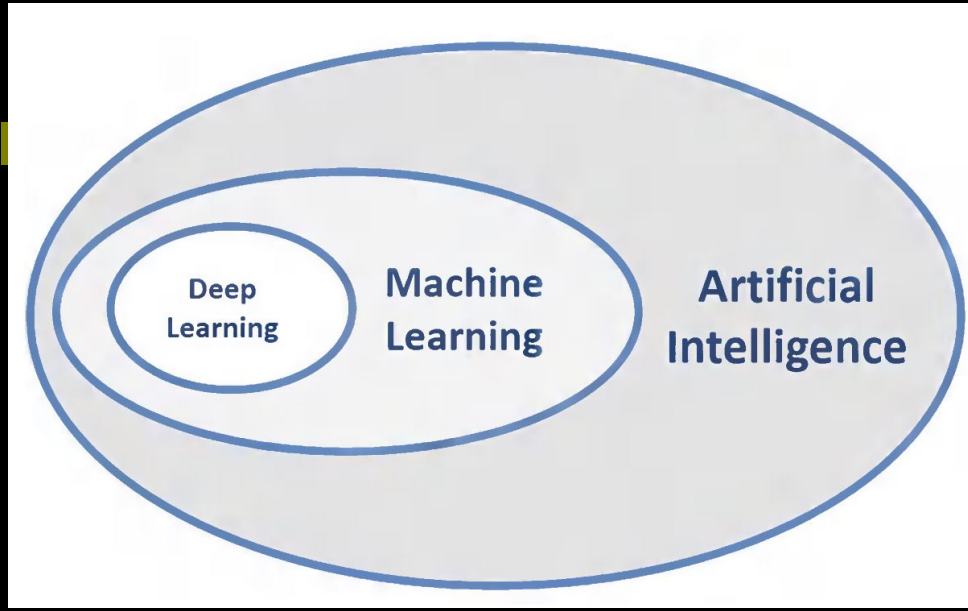
Introduction to AI-The Basics

The term AI is rooted from basic computer science and automation. Most advanced AI include these characteristics:

-Takes in an input
-Outputs information as an action
-Converts these inputs into numeral values
-Stores values in a data library and ”learns” from it

Such examples include Siri, Tesla Autopilot, Google Home, Waymo Autopilot, Google Ads, and many more. Siri in particular uses suggestions from [frequent] human input, which they set as the database. For instance, if a person asks Siri to find online images of cars, you will find images of cars. AI use ML (Machine Learning)/DL (Deep Learning) algorithms to “learn”.

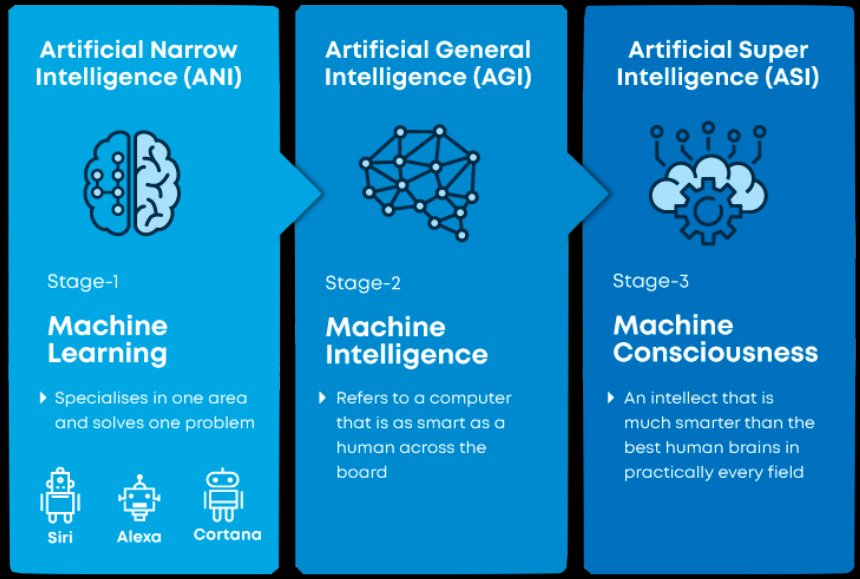
Figure 0 [1.0]



Introduction to AI-Levels of AI

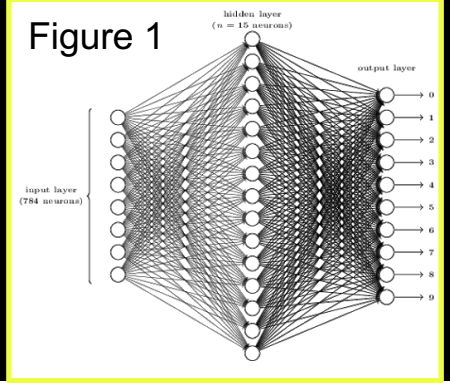
-ANI (Artificial Narrow Intelligence) specializes in few types of inputs/outputs
-AGI (Artificial General Intelligence): much smarter and can take distinct types of inputs/outputs, which we are studying today.
-**Hypothetical** ASI (Artificial Super Intelligence), has highest intellect than one of the most acute humans and will be known to have “machine consciousness,” and do not require human input.

Some believe when we reach a point of AI where it has a mind of its own, it might be the greatest achievement, but also the last achievement, where AI could override human supervision and cause calamity. Others believe it will be extremely impactful on our daily lives, and they could do hard tasks just for us in a matter of seconds, while it takes days for us humans.



Introduction to AI-Learning

Figure 1



-Machine learning (ML)- AI uses algorithms to interpret data and predict patterns.

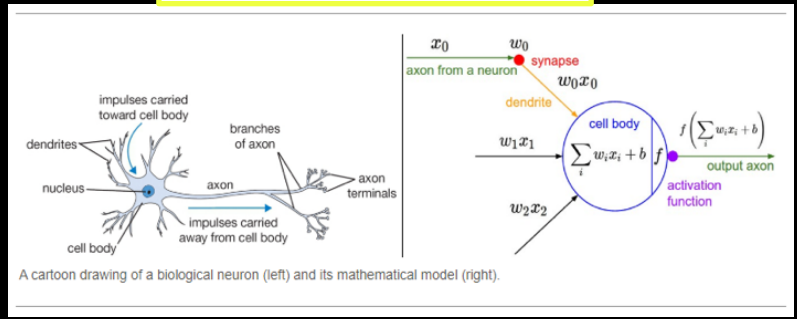
-Deep learning (DL)- AI has much more advanced intelligence, can mimic a human neural network [2], and can do more things by themselves.

-Figure 1[1.1] represents a neural network [2] : each of the balls are called neurons, each associating with a weight (w)[2]. They act very similar to human neurons and gave us a better idea of how they work. In the input layer [2], they put in [human] inputs;

-Back prorogation

-hidden layer(s) [2], the inputs can go through extraction processes, organizational segregation from the extracted data, and other methods of data orientation/interpretations.

Figure 2



-Figure 2[1.2] represents of the relationship of a human

They act like a mathematical function, and most inputs can be converted to numbers. For each neuron in the hidden layer [2], valued -1 t
1, they look for specific components, which are in their data library, from the input. The neurons of the next hidden layer [2] may look for more complicated components. However, this may take more time to compute, and can take up much energy. So, there are….

-deep neural networks [2], with DL: there are more hidden layers [2] which use multiple/more ways to find a specific component(s), such as including the backward propagation.

-output layer[2]: they give out the output and does an action out of those outputs

-The sum of the output neurons must be equal 1

Introduction to AI-Activation Functions

Usually, the activation functions [3] are added to support a neural network [2] to adapt to patterns from the data, with non-linearity. These functions are for converting inputs into numbers in a specific range, or associates with the numbers. There are multiple features an activation function [3] has:

-vanishing gradient problem: the activation shifting towards 0 because of the decrease of the output through a back prorogation
-zero-centered: the output of the activation function [3] does not go towards 0 and gradients do not shift
-computationally expensive: activation functions [3] are required in every neuron in every layer

There are multiple activation functions [3] used:

-Sigmoid: sig(t)=1/(1+e^(-t); receives real-valued inputs and interprets between 0 and 1; it is computationally expensive and causes the vanishing gradient problem/not zero-centered
-Hyperbolic Tangent (Tanh): an extent to the Sigmoid but is zero-centered
-Softmax: checks if that the sum of all of the neurons in the output layer[2] is equal to 1
-Rectified Linear Unit (ReLU): f(x)=max(0,x); widely used; does not have the vanishing gradient problem; not zero-centered, and some of the neurons “die out” (“dying ReLU” problem), and do not react to the input/output
-Perimetric (PReLU)/Leaky ReLU: f(x)=max(αx,x); Leaky ReLU solves the “dying ReLU” partly; become a linear function if α = 1, thus α is never close to 1; if α is a hyperparameter (parameter that controls the learning), it becomes the PReLU
-ReLU6: f(x)=min(max(0,x),6);

Introduction to AI-Not All AI are Advanced

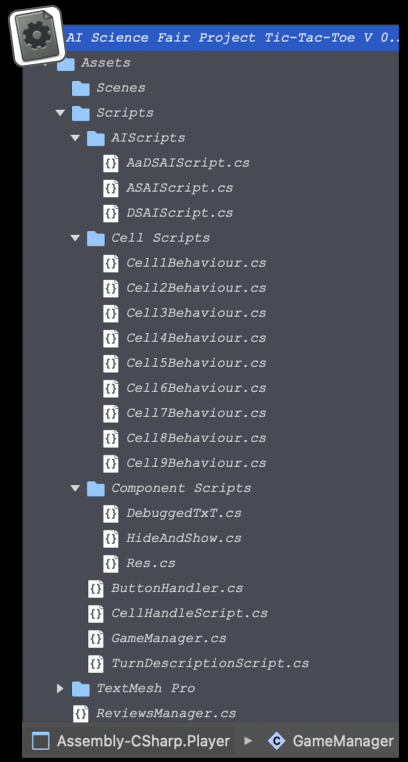
However, AI can much simpler than these implications, such as a simple algorithm(s) for playing a role in Tic-Tac-Toe for attacking/defending, which is my experiment for my research. Or a television device that turns on when their audio sensors detect the word “TV,” which can be refactored into a simple software algorithm with no learning. Some AI do not ML/DL.

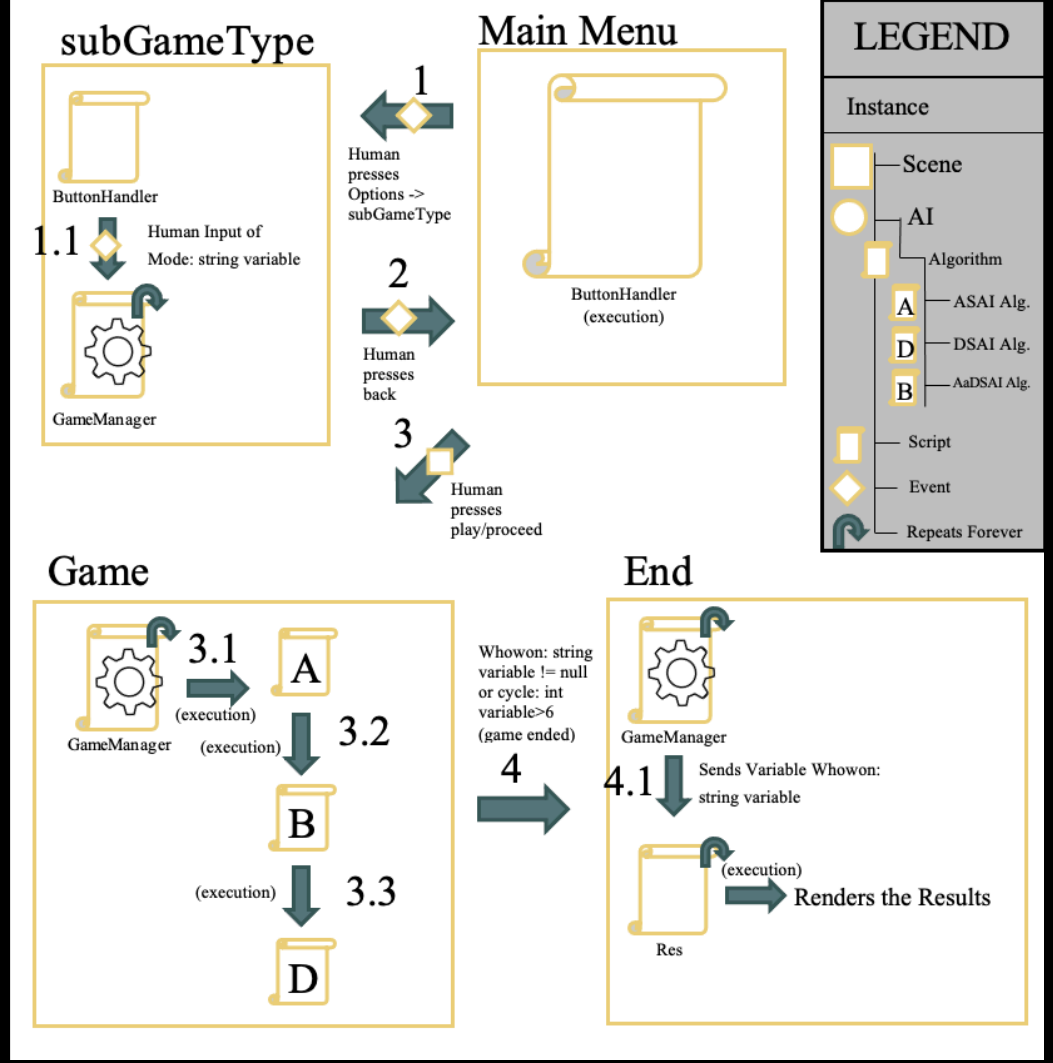
General Software Structure

The software is built through the Unity software and is controlled through scenes and asset objects (with .meta files including data of their components and settings), mainly the GameManager.cs C# script. It controls and calls the algorithms of my software and transitions their turns and configures the values for the fields/functions required for the algorithms, such as the Cycle: int, specificCell[1-9]Empty: string, boolCell[1-9]Empty: bool, mode: string, Fill[1~9](whichAI), and others included in the Solution Outline[1,2] slides. There are 6 scenes: “Main Menu”, “Options Menu”, “Game”, “subGameType”, and “END”. The GameManager.cs script is executed in the “Game” scene, the “subGameType” scene is where you pick the gamemode, and the “Main Menu” scene is where you proceed to the “Game” or “Options Menu”/“subGameType” scene. Finally, the “END” scene is where they show the results, as well as the renderings and shows the variable value of strand: string, which are a combination of all the variables in the GameManager.cs script.

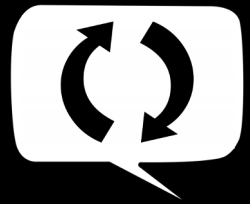
For the main cell moderations, there are 9 cells, all which are labeled by their numeric value. The specificCell[1-9]Empty fields can be 5 possible string values: “filledbyASAI”, “filledbyDSAI”, “filledbyAaDSAI”, “yes”, and null. The boolCell[1-9]Empty fields can be, of course, 3 possible values: true, false, and null. By these values, the “Cell[1-9]Render” game objects in the “Game” scene can render the assigned sprite constant “E” if it’s representing variable equals “yes”, “X” if it is equal “filledbyASAI”, “O” if it is equal “filledbyDSAI”, and “B” if it is equal “filledbyAaDSAI”. The strand is debugged in the software execution routine and appears in the “END” scene.

Solution/Project Hierarchy




subGameType
ButtonHandler
1.1
Human Input of
Mode: string variable
GameManager
1
Human
presses
Options ->
subGameType
2
Human
presses
back
3
Human
presses
play/proceed
Main Menu
ButtonHandler
(execution)
LEGEND
Instance
Scene
AI
Algorithm
A
ASAI Alg.
D
DSAI Alg.
B
AaDSAI Alg.
Script
Event
Repeats Forever
Game
GameManager
3.1
(execution)
A
(execution)
3.2
B
(execution)
3.3
D
Whowon: string
variable != null
or cycle: int
variable>6
(game ended)
4
End
GameManager
4.1
Sends Variable Whowon:
string variable
(execution)
Renders the Results
Res

Methodology



The data/results will be recorded through points, as where if an AI wins, it will gain 1 point; loses 1 point if it is defeated; neither gain or lose points if a tie occurs. There will be graphs in forms of win/lose average, first/second point difference, and simple point count bar graphs, recorded in a Google Sheets [total, f/s, avg] and Docs documents. Furthermore, negative points are tolerated.

Win/Lose Average Method: $f(AI)Avg^{W/L} = \frac{[AI]^{Pt\ total}}{80}$

First/Second Pt Differ Method: $Differ^{1st/2nd} = f(AI)^{Pt\ total\ 1st} = ([AI]andDSAI)^{[AI]\ Total} + ([AI]andAaDSAI)^{[AI]\ Total} + ([AI]andASAI)^{[AI]\ Total} - ([AI]and[AI])^{[AI]\ Total}$

$f(AI)^{Pt\ total\ 2nd} = (DSAIand[AI])^{[AI]\ Total} + (AaDSAIand[AI])^{[AI]\ Total} + (ASAIand[AI])^{[AI]\ Total} - ([AI]and[AI])^{[AI]\ Total}$

Total Method $f(AI)^{Total} = f(AI)^{Pt\ total\ 1st} + f(AI)^{Pt\ total\ 2nd}$

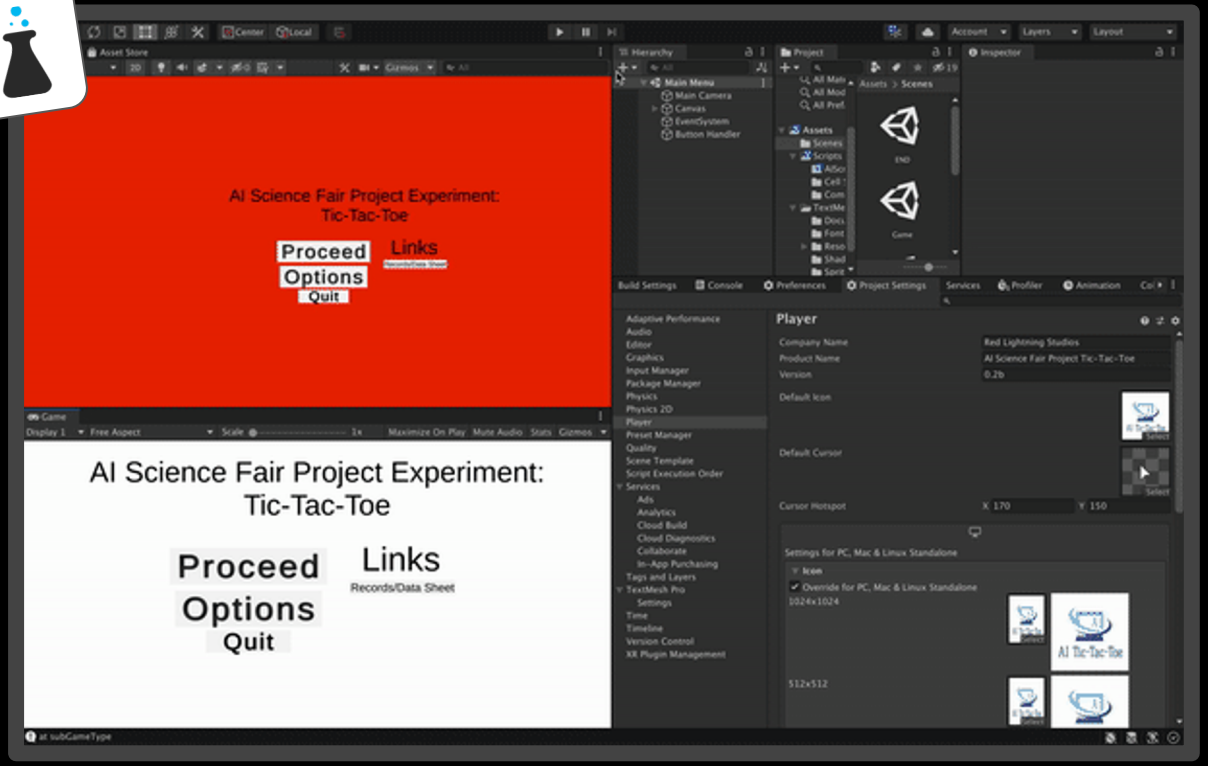
+ (ASAIand[AI])[AI] Total - ([AI]and[AI])[AI] Total

otal Method f(AI)Total = f(AI) Pt total 1st + f(AI) Pt total 2nd

Procedure

[1]: Run my Unity-built software
[2]: Pick the desired game mode, 20 times each;
[3]: Record them onto the Google Sheets [total, f/s, avg] and Google Docs.

Procedure Walkthrough



Solution Outline (1)

```
public void Update()
{
    checkif();
    Debug.Log("CYCLE" + Cycle);
    checkif();
    strand = "specificCells: " + specificCell1Empty +
    ", " + specificCell2Empty +
    ", " + specificCell3Empty +
    ", " + specificCell4Empty +
    ", " + specificCell5Empty +
    ", " + specificCell6Empty +
    ", " + specificCell7Empty +
    ", " + specificCell8Empty +
    ", " + specificCell9Empty +
    "; boolCells: " + boolCell1Empty +
    ", " + boolCell2Empty +
    ", " + boolCell3Empty +
    ", " + boolCell4Empty +
    ", " + boolCell5Empty +
    ", " + boolCell6Empty +
    ", " + boolCell7Empty +
    ", " + boolCell8Empty +
    ", " + boolCell9Empty +
    "; Other: " + Turn +
    ", " + TurnNum +
    ", " + whoWon +
    ", " + GameType;
    //", " + subGameType;
    checkif();
    Debug.Log(strand);
    checkif();
    if (whoWon == null && ALLCELLSEMPTY == false)
    {
        tik(mode);
    }
    else
    {
        Debug.Log("GAME ENDED");
        SceneManager.LoadScene("END");
        if (Cycle > 10)
        {
            Debug.Log("ERROR");
        }
    }
    if (Cycle > 10)
    {
        Debug.Log("GAME ENDED");
        SceneManager.LoadScene("END");
    }
    Debug.Log(mode);
    Debug.Log(GameType);
}
```

Execution Routine (through the Unity Event Function Update())

Fields (variables/constants)

```
public static string specificCell1Empty = "yes";
public static string specificCell2Empty = "yes";
public static string specificCell3Empty = "yes";
public static string specificCell4Empty = "yes";
public static string specificCell5Empty = "yes";
public static string specificCell6Empty = "yes";
public static string specificCell7Empty = "yes";
public static string specificCell8Empty = "yes";
public static string specificCell9Empty = "yes";
public static bool boolCell1Empty = true;
public static bool boolCell2Empty = true;
public static bool boolCell3Empty = true;
public static bool boolCell4Empty = true;
public static bool boolCell5Empty = true;
public static bool boolCell6Empty = true;
public static bool boolCell7Empty = true;
public static bool boolCell8Empty = true;
public static bool boolCell9Empty = true;
public static string Turn;
public static bool ALLCELLSEMPTY;
public static int Cycle;
public static int TurnNum;
public static string whoWon;
public static string GameType = "onlytwoeach";
public static string strand;
public static string mode;
//-----------------------------------------------\\
```

checkif(no args) function for checking if all cells are filled, another condition that the execution routine can use for ending the game and proceeding to the END scene

```
public void checkif()
{
    if (boolCell1Empty == false && boolCell2Empty == false && boolCell3Empty == false && boolCell4Empty == false && boolCell5Empty == false && boolCell6Empty == false && boolCell7Empty == false && boolCell8Empty == false && boolCell9Empty == false)
        ALLCELLSEMPTY = true;
    else
        ALLCELLSEMPTY = false;
}
```

Solution Outline (2)

FillCell(string whichAI) Functions

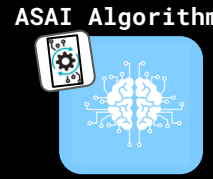
```
//------------------------------------------------\\
public static void Fill1(string whichAI)
{
    if (specificCell1Empty == "yes")
        specificCell1Empty = "filledby" + whichAI;
    else
        Debug.Log("gErr_001\n" + "This may have been caused because none the conditions are == true");
}
public static void Fill2(string whichAI)
{
    if (specificCell2Empty == "yes")
        specificCell2Empty = "filledby" + whichAI;
    else
        Debug.Log("gErr_001\n" + "This may have been caused because none the conditions are == true");
}
public static void Fill3(string whichAI)
{
    if (specificCell3Empty == "yes")
        specificCell3Empty = "filledby" + whichAI;
    else
        Debug.Log("gErr_001\n" + "This may have been caused because none the conditions are == true");
}
public static void Fill4(string whichAI)
{
    if (specificCell4Empty == "yes")
        specificCell4Empty = "filledby" + whichAI;
    else
        Debug.Log("gErr_001\n" + "This may have been caused because none the conditions are == true");
}
public static void Fill5(string whichAI)
{
    if (specificCell5Empty == "yes")
        specificCell5Empty = "filledby" + whichAI;
    else
        Debug.Log("gErr_001\n" + "This may have been caused because none the conditions are == true");
}
public static void Fill6(string whichAI)
{
    if (specificCell6Empty == "yes")
        specificCell6Empty = "filledby" + whichAI;
    else
        Debug.Log("gErr_001\n" + "This may have been caused because none the conditions are == true");
}
public static void Fill8(string whichAI)
{
    if (specificCell8Empty == "yes")
        specificCell8Empty = "filledby" + whichAI;
    else
        Debug.Log("gErr_001\n" + "This may have been caused because none the conditions are == true");
}
public static void Fill9(string whichAI)
{
    if (specificCell9Empty == "yes")
        specificCell9Empty = "filledby" + whichAI;
    else
        Debug.Log("gErr_001\n" + "This may have been caused because none the conditions are == true");
}
//------------------------------------------------\\
```

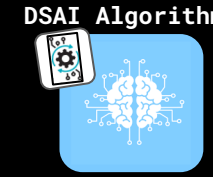
CheckWhoWon(no args) Function (excerpt)

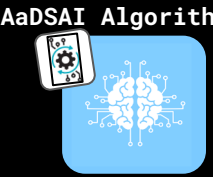
```
public static void CheckWhoWon()
{
    //horizontal:
    //ASAI:
    if (GameManager.specificCell1Empty == "filledbyASAI" && GameManager.specificCell2Empty == "filledbyASAI" && GameManager.specificCell3Empty == "filledbyASAI")
    {
        whoWon = "ASAI";
    }
    if (GameManager.specificCell4Empty == "filledbyASAI" && GameManager.specificCell5Empty == "filledbyASAI" && GameManager.specificCell6Empty == "filledbyASAI")
    {
        whoWon = "ASAI";
    }
    if (GameManager.specificCell7Empty == "filledbyASAI" && GameManager.specificCell8Empty == "filledbyASAI" && GameManager.specificCell9Empty == "filledbyASAI")
    {
        whoWon = "ASAI";
    }
    //DSAI:
    if (GameManager.specificCell1Empty == "filledbyDSAI" && GameManager.specificCell2Empty == "filledbyDSAI" && GameManager.specificCell3Empty == "filledbyDSAI")
    {
        whoWon = "DSAI";
    }
    if (GameManager.specificCell4Empty == "filledbyDSAI" && GameManager.specificCell5Empty == "filledbyDSAI" && GameManager.specificCell6Empty == "filledbyDSAI")
    {
        whoWon = "DSAI";
    }
    if (GameManager.specificCell7Empty == "filledbyDSAI" && GameManager.specificCell8Empty == "filledbyDSAI" && GameManager.specificCell9Empty == "filledbyDSAI")
    {
        whoWon = "DSAI";
    }
    //AaDSAI:
    if (GameManager.specificCell1Empty == "filledbyAaDSAI" && GameManager.specificCell2Empty == "filledbyAaDSAI" && GameManager.specificCell3Empty == "filledbyAaDSAI")
    {
        whoWon = "AaDSAI";
    }
    if (GameManager.specificCell4Empty == "filledbyAaDSAI" && GameManager.specificCell5Empty == "filledbyAaDSAI" && GameManager.specificCell6Empty == "filledbyAaDSAI")
    {
        whoWon = "AaDSAI";
    }
    if (GameManager.specificCell7Empty == "filledbyAaDSAI" && GameManager.specificCell8Empty == "filledbyAaDSAI" && GameManager.specificCell9Empty == "filledbyAaDSAI")
    {
        whoWon = "AaDSAI";
    }
```

Algorithm

The AI Algorithms depend on the gamemode, the current cycle (ex: if the ASAI Algorithm and DSAI Algorithm is executed, the cycle is increased by 1)m what cells are filled and not filled by a specific AI(s). There are 3 Algorithms for 3 AIs: ASAI (attack AI), DSAI (defense AI), and AaDSAI (Attack/Defense AI). Their algorithms are sometimes applied in a slightly different order depending on the gamemode.







ASAI algorithm:

-[1,2]: fills the center cell (cell 5) if it is empty, but if it is already filled, it fills a random cell

-[3,4,5]: it uses multiple conditions to generate its next move, which it's main goal is winning.

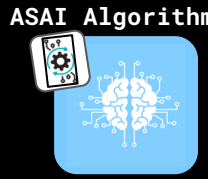
One example of a condition listed in my ASAIScript.cs is:

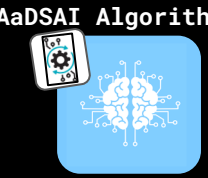
```
//[...truncated…]
else if (II && III && si == "yes"){ GameManager.Fill1("ASAI"); }
//[...truncated…]
```

Algorithm

DSAI algorithm:

- similar to the ASAI algorithm, except that for the conditions, it doesn't use plain bool variables, such as the used II and III fields, instead it moderates conditions for the cell where if it is NOT filled by the DSAI and it is NOT empty. I have implemented this because it will then defend against itself, and this can interfere with my algorithm and cause various bugs.
- instead of assigning the only argument “ASAI” of the void function GameManager.Fill(1), I would assign the argument “DSAI” instead, since it would act like the ASAI if not careful.






```
//[...truncated…]
 //horizontal
        //1,2,3
if (I && II && siii == "yes"){ GameManager.Fill3("AaDSAI"); Debug.Log("FILLED")}
else if (I && III && sii == "yes"){ GameManager.Fill2("AaDSAI"); Debug.Log("FILLED");}
//[...truncated…]
```

Algorithm

AaDSAI algorithm:

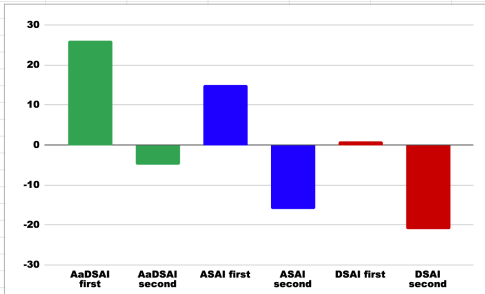
- The combination of the ASAI and DSAI algorithm, except for each cycle, they do the DSAI strategy part first and then the ASAI strategy part

- Example conditional of algorithm:

```
//[...truncated…]
 //horizontal
          //1,2,3
if (I && II && siii == "yes"){ GameManager.Fill3("AaDSAI"); Debug.Log("FILLED")}
else if (I && III && sii == "yes"){ GameManager.Fill2("AaDSAI"); Debug.Log("FILLED");}
//[...truncated…]
else { else if (I && IV && svii == "yes") {GameManager.Fill7("AaDSAI");} //[...truncated…]}
```

Results

	Pt
AaDSAI first	26
AaDSAI second	-5
ASAI first	15
ASAI second	-16
DSAI first	1
DSAI second	-21



	ASAI	DSAI	AaDSAI
Avg Pt.	-0.0125	-0.25	0.25

Date: March 2
MODE: "DSAIandAaDSAI"

1. AS/DS/AaDS AI won
2. AS/DS/AaDS AI won
3. AS/DS/AaDS AI won
4. AS/DS/AaDS AI won
5. AS/DS/AaDS AI won
6. AS/DS/AaDS AI won
7. AS/DS/AaDS AI won
8. AS/DS/AaDS AI won
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won

Date: March 2

11. AS/DS/AaDS AI won
12. AS/DS/AaDS AI won
13. AS/DS/AaDS AI won
14. AS/DS/AaDS AI won
15. AS/DS/AaDS AI won
16. AS/DS/AaDS AI won
17. AS/DS/AaDS AI won
18. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

Date: March 2
MODE: "AaDSAIandDSAI"

1. AS/DS/AaDS AI won
2. AS/DS/AaDS AI won
3. AS/DS/AaDS AI won
4. AS/DS/AaDS AI won
5. AS/DS/AaDS AI won
6. AS/DS/AaDS AI won
7. AS/DS/AaDS AI won
8. AS/DS/AaDS AI won
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won

Date: March 2

11. AS/DS/AaDS AI won
12. AS/DS/AaDS AI won
13. AS/DS/AaDS AI won
14. AS/DS/AaDS AI won
15. AS/DS/AaDS AI won
16. AS/DS/AaDS AI won
17. AS/DS/AaDS AI won
18. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

Date: March 2
MODE: "ASAIandAaDSAI"

1. AS/DS/AaDS AI won
2. AS/DS/AaDS AI won
3. AS/DS/AaDS AI won
4. AS/DS/AaDS AI won
5. AS/DS/AaDS AI won
6. AS/DS/AaDS AI won
7. AS/DS/AaDS AI won
8. AS/DS/AaDS AI won
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won

Date: March 2

11. AS/DS/AaDS AI won
12. AS/DS/AaDS AI won
13. AS/DS/AaDS AI won
14. AS/DS/AaDS AI won
15. AS/DS/AaDS AI won
16. AS/DS/AaDS AI won
17. AS/DS/AaDS AI won
18. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

Date: 3/1/2021
MODE: "ASAIandDSAI"

1. AS/DS/AaDS AI won
2. AS/DS/AaDS AI won
3. AS/DS/AaDS AI won
4. AS/DS/AaDS AI won
5. AS/DS/AaDS AI won
6. AS/DS/AaDS AI won
7. AS/DS/AaDS AI won
8. AS/DS/AaDS AI won
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won

Date: 3/1/2021

11. AS/DS/AaDS AI won
12. AS/DS/AaDS AI won
13. AS/DS/AaDS AI won
14. AS/DS/AaDSAI won
15. AS/DS/AaDS AI won
16. AS/DS/AaDS AI won
17. AS/DS/AaDS AI won
18. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

Date: 3/1/2021
MODE: "DSAIandASAI"

1. AS/DS/AaDS AI won
2. AS/DS/AaDS AI won
3. AS/DS/AaDS AI won
4. AS/DS/AaDS AI won
5. AS/DS/AaDS AI won
6. AS/DS/AaDS AI won
7. AS/DS/AaDS AI won
8. AS/DS/AaDS AI won
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won

Date: 3/1/2021

11. AS/DS/AaDS AI won
12. AS/DS/AaDS AI won
13. AS/DS/AaDSAI won
14. AS/DS/AaDS AI won
15. AS/DS/AaDS AI won
16. AS/DS/AaDS AI won
17. AS/DS/AaDS AI won
18. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

Date: March 2
MODE: "AaDSAIandASAI"

1. AS/DS/AaDS AI won
2. AS/DS/AaDS AI won
3. AS/DS/AaDS AI won
4. AS/DS/AaDS AI won
5. AS/DS/AaDS AI won
6. AS/DS/AaDS AI won
7. AS/DS/AaDS AI won
8. AS/DS/AaDS AI won
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won

Date: March 2

11. AS/DS/AaDS AI won
12. AS/DS/AaDS AI won
13. AS/DS/AaDS AI won
14. AS/DS/AaDS AI won
15. AS/DS/AaDS AI won
16. AS/DS/AaDS AI won
17. AS/DS/AaDS AI won
18. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

Note: the winning AI will be highlighted in yellow; the mode will be highlighted green

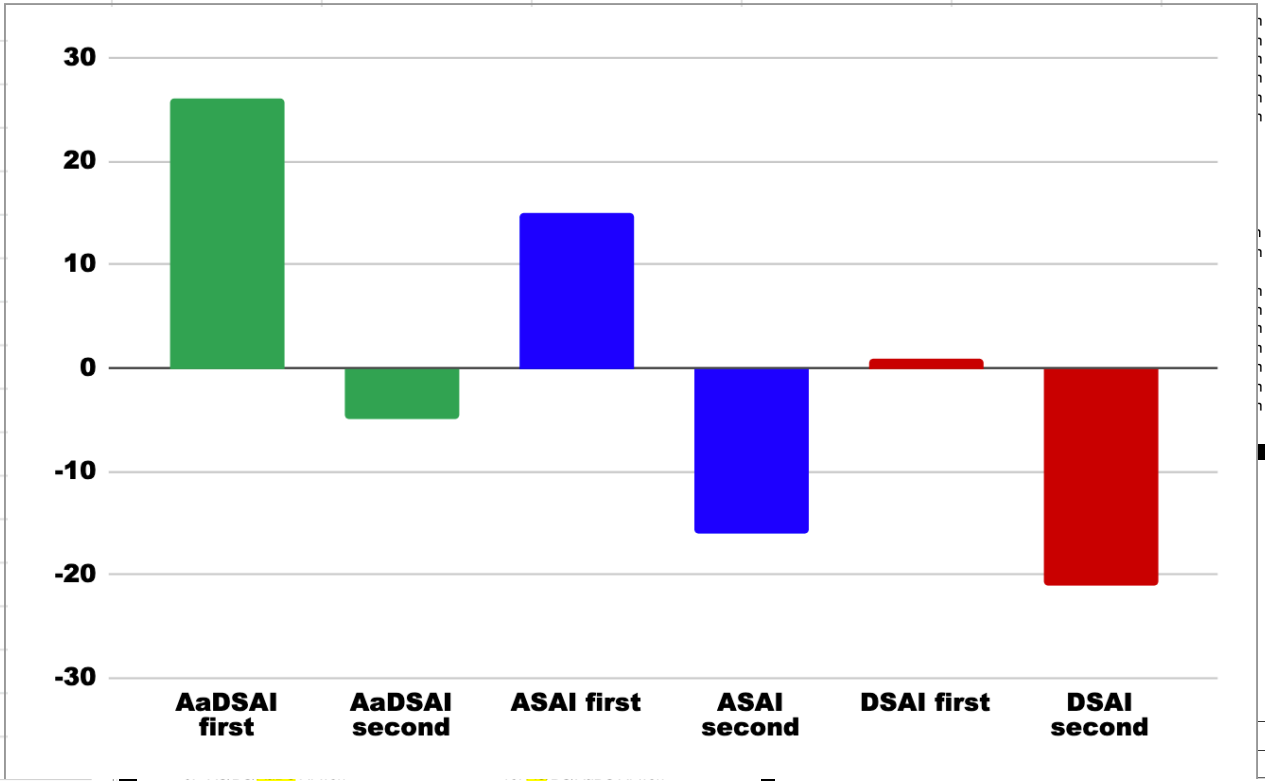
AS AI	DS AI
Overall Count (pts.): -1	Overall Count(pts.): -20

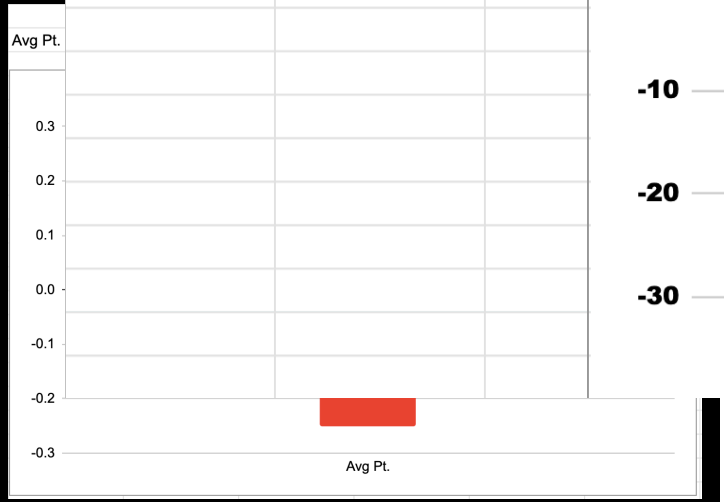
Note: the winning AI will be highlighted in yellow; the mode will be highlighted green

AaDS AI
Overall Count(pts.): 21

Results

	Pt
AaDSAI first	26
AaDSAI second	-5
ASAI first	15
ASAI second	-16
DSAI first	1
DSAI second	-21





Date: 3/1/2021
MODE: "ASAIandDSAI"

Date: 3/1/2021
11: AS/DS/AaDS AI won

7. AS/DS/AaDS AI won
8. AS/DS/AaDS AI won
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won

17. AS/DS/AaDS AI won
18. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

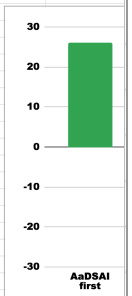
Note: the winning AI will be highlighted in yellow; the mode will be highlighted green

AaDS AI
Overall Count(pts.): 21

Results

	ASAI	DSAI	AaDSAI		
Avg Pt.	-0.0125	-0.25	0.25		

Pt	
AaDSAI first	26
AaDSAI second	-5
ASAI first	15
ASAI second	-16
DSAI first	1
DSAI second	-21



Date: 3/1/2021
11: AS/DS/AaDS AI won
12: AS/DS/AaDS AI won
13: AS/DS/AaDS AI won
14: AS/DS/AaDSAI won
15: AS/DS/AaDS AI won
16: AS/DS/AaDS AI won
17: AS/DS/AaDS AI won
18: AS/DS/AaDS AI won
19: AS/DS/AaDS AI won
20: AS/DS/AaDS AI won

Date: 3/1/2021

11: AS/DS/AaDS AI won
12: AS/DS/AaDS AI won
13: AS/DS/AaDSAI won
14: AS/DS/AaDS AI won
15: AS/DS/AaDS AI won
16: AS/DS/AaDS AI won
17: AS/DS/AaDS AI won
18: AS/DS/AaDS AI won
19: AS/DS/AaDS AI won
20: AS/DS/AaDS AI won

Date: March 2

AS/DS/AaDS AI won
AS/DS/AaDS AI won
AS/DS/AaDS AI won
AS/DS/AaDS AI won
AS/DS/AaDS AI won
AS/DS/AaDS AI won
AS/DS/AaDS AI won
AS/DS/AaDS AI won
AS/DS/AaDS AI won
AS/DS/AaDS AI won

ode will be highlighted green

l

all Count(pts.): -20

ode will be highlighted green

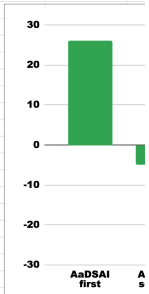
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

AaDS AI

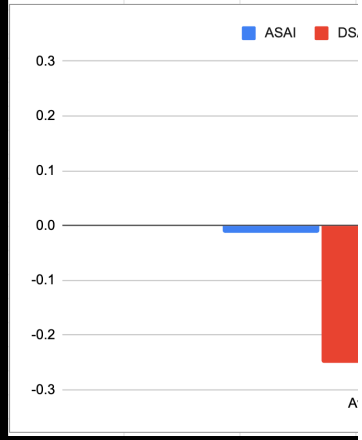
Overall Count(pts.): 21

Results

	Pt
AaDSAI first	26
AaDSAI second	-5
ASAI first	15
ASAI second	-16
DSAI first	1
DSAI second	-21



	ASAI	DSAI
Avg Pt.	-0.0125	-0.25



Date: March 2

MODE: “AaDSAIandDSAI”

1. AS/DS/AaDS AI won
2. AS/DS/AaDS AI won
3. AS/DS/AaDS AI won
4. AS/DS/AaDS AI won
5. AS/DS/AaDS AI won
6. AS/DS/AaDS AI won
7. AS/DS/AaDS AI won
8. AS/DS/AaDS AI won
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won

Date: March 2

11. AS/DS/AaDS AI won
12. AS/DS/AaDS AI won
13. AS/DS/AaDS AI won
14. AS/DS/AaDS AI won
15. AS/DS/AaDS AI won
16. AS/DS/AaDS AI won
17. AS/DS/AaDS AI won
18. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

Date: March 2

MODE: “ASAIandAaDSAI”

1. AS/DS/AaDS AI won
2. AS/DS/AaDS AI won
3. AS/DS/AaDS AI won
4. AS/DS/AaDS AI won
5. AS/DS/AaDS AI won
6. AS/DS/AaDS AI won
7. AS/DS/AaDS AI won
8. AS/DS/AaDS AI won
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won

Date: March 2

11. AS/DS/AaDS AI won
12. AS/DS/AaDS AI won
13. AS/DS/AaDS AI won
14. AS/DS/AaDS AI won
15. AS/DS/AaDS AI won
16. AS/DS/AaDS AI won
17. AS/DS/AaDS AI won
18. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

Date: 3/1/2021

11: AS/DS/AaDS AI won
12: AS/DS/AaDS AI won
13: AS/DS/AaDS AI won
14: AS/DS/AaDSAI won
15: AS/DS/AaDS AI won
16: AS/DS/AaDS AI won
17: AS/DS/AaDS AI won
18: AS/DS/AaDS AI won
19: AS/DS/AaDS AI won
20: AS/DS/AaDS AI won

Date: 3/1/2021

11: AS/DS/AaDS AI won
12: AS/DS/AaDS AI won
13: AS/DS/AaDSAI won
14: AS/DS/AaDS AI won
15: AS/DS/AaDS AI won
16: AS/DS/AaDS AI won
17: AS/DS/AaDS AI won
18: AS/DS/AaDS AI won
19: AS/DS/AaDS AI won
20: AS/DS/AaDS AI won

Date: March 2

11. AS/DS/AaDS AI won
12. AS/DS/AaDS AI won
13. AS/DS/AaDS AI won
14. AS/DS/AaDS AI won
15. AS/DS/AaDS AI won
16. AS/DS/AaDS AI won
17. AS/DS/AaDS AI won
18. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

yellow: the mode will be highlighted green

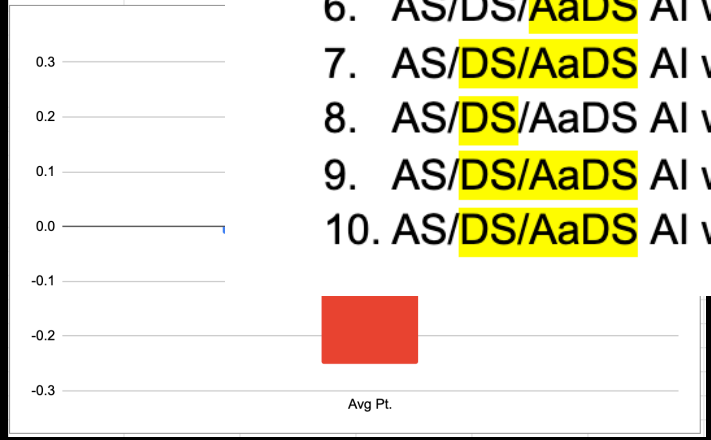
	DS AI
	Overall Count(pts.): -20

yellow: the mode will be highlighted green

Results

	Pt
AaDSAI first	26
AaDSAI second	-5
ASAI first	15
ASAI second	-16
DSAI first	1
DSAI second	-21

	ASAI
Avg Pt.	-0.012



Date: March 2

MODE: “DSAIandAaDSAI”

1. AS/DS/AaDS AI won
2. AS/DS/AaDS AI won
3. AS/DS/AaDS AI won
4. AS/DS/AaDS AI won
5. AS/DS/AaDS AI won
6. AS/DS/AaDS AI won
7. AS/DS/AaDS AI won
8. AS/DS/AaDS AI won
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won

Date: March 2

11. AS/DS/AaDS AI won
12. AS/DS/AaDS AI won
13. AS/DS/AaDS AI won
14. AS/DS/AaDS AI won
15. AS/DS/AaDS AI won
16. AS/DS/AaDS AI won
17. AS/DS/AaDS AI won
18. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

Date: 3/1/2021

MODE: “ASAIandDSAI”

1. AS/DS/AaDS AI won

Date: 3/1/2021

11: AS/DS/AaDS AI won
12: AS/DS/AaDS AI won
13: AS/DS/AaDS AI won

5. AS/DS/AaDS AI won
6. AS/DS/AaDS AI won
7. AS/DS/AaDS AI won
8. AS/DS/AaDS AI won
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won

15. AS/DS/AaDS AI won
16. AS/DS/AaDS AI won
17. AS/DS/AaDS AI won
18. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

d green

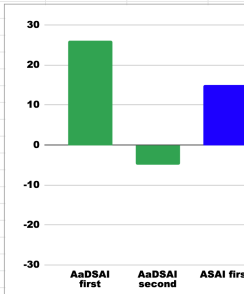
Overall Count (pts.): -1	Overall Count(pts.): -20

Note: the winning AI will be highlighted in yellow; the mode will be highlighted green

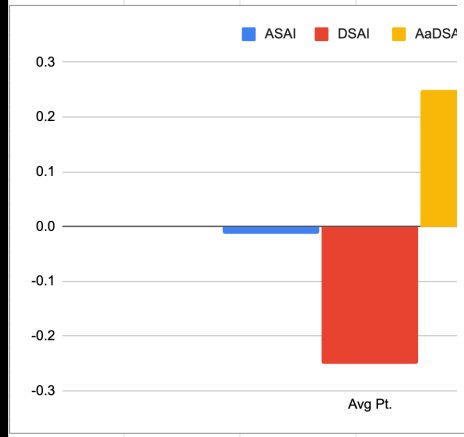
AaDS AI
Overall Count(pts.): 21

Results

	Pt
AaDSAI first	26
AaDSAI second	-5
ASAI first	15
ASAI second	-16
DSAI first	1
DSAI second	-21



	ASAI	DSAI	AaDSAI
Avg Pt.	-0.0125	-0.25	0.2



Date: 3/1/2021

MODE: "ASAIandDSAI"

1. AS/DS/AaDS AI won
2. AS/DS/AaDS AI won
3. AS/DS/AaDS AI won
4. AS/DS/AaDS AI won
5. AS/DS/AaDS AI won
6. AS/DS/AaDS AI won
7. AS/DS/AaDS AI won
8. AS/DS/AaDS AI won
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won

Date: 3/1/2021

11: AS/DS/AaDS AI won
12: AS/DS/AaDS AI won
13: AS/DS/AaDS AI won
14: AS/DS/AaDSAI won
15: AS/DS/AaDS AI won
16: AS/DS/AaDS AI won
17: AS/DS/AaDS AI won
18: AS/DS/AaDS AI won
19: AS/DS/AaDS AI won
20: AS/DS/AaDS AI won

Date: 3/1/2021

MODE: "DSAIandASAI"

1. AS/DS/AaDS AI won
2. AS/DS/AaDS AI won
3. AS/DS/AaDS AI won
4. AS/DS/AaDS AI won
5. AS/DS/AaDS AI won
6. AS/DS/AaDS AI won
7. AS/DS/AaDS AI won
8. AS/DS/AaDS AI won
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won

Date: 3/1/2021

11: AS/DS/AaDS AI won
12: AS/DS/AaDS AI won
13: AS/DS/AaDSAI won
14: AS/DS/AaDS AI won
15: AS/DS/AaDS AI won
16: AS/DS/AaDS AI won
17: AS/DS/AaDS AI won
18: AS/DS/AaDS AI won
19: AS/DS/AaDS AI won
20: AS/DS/AaDS AI won

rch 2

AaDSAIandASAI"

aDS AI won

aDS AI won

aDS AI won

aDS AI won

aDS AI won

aDS AI won

aDS AI won

aDS AI won

aDS AI won

aDS AI won

Date: March 2

11. AS/DS/AaDS AI won
12. AS/DS/AaDS AI won
13. AS/DS/AaDS AI won
14. AS/DS/AaDS AI won
15. AS/DS/AaDS AI won
16. AS/DS/AaDS AI won
17. AS/DS/AaDS AI won
18. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

g AI will be highlighted in yellow; the mode will be highlighted green

	DS AI
pts.): -1	Overall Count(pts.): -20

g AI will be highlighted in yellow; the mode will be highlighted green

10. AS/DS/AaDS AI won

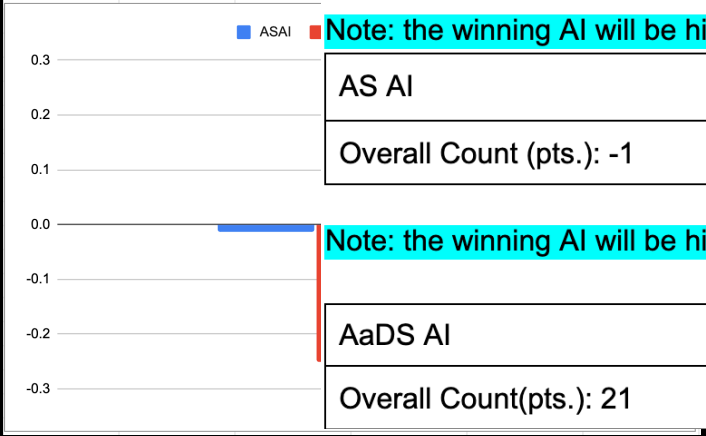
20. AS/DS/AaDS AI won

Overall Count(pts.): 21

Results

	Pt
AaDSAI first	26
AaDSAI second	-5
ASAI first	15
ASAI second	-16
DSAI first	1
DSAI second	-21

	ASAI	DSAI
Avg Pt.	-0.0125	-(



Date: March 2

MODE: "AaDSAIandASAI"

1. AS/DS/AaDS AI won
2. AS/DS/AaDS AI won
3. AS/DS/AaDS AI won
4. AS/DS/AaDS AI won
5. AS/DS/AaDS AI won
6. AS/DS/AaDS AI won
7. AS/DS/AaDS AI won
8. AS/DS/AaDS AI won
9. AS/DS/AaDS AI won
10. AS/DS/AaDS AI won

Date: March 2

11. AS/DS/AaDS AI won
12. AS/DS/AaDS AI won
13. AS/DS/AaDS AI won
14. AS/DS/AaDS AI won
15. AS/DS/AaDS AI won
16. AS/DS/AaDS AI won
17. AS/DS/AaDS AI won
18. AS/DS/AaDS AI won
19. AS/DS/AaDS AI won
20. AS/DS/AaDS AI won

Date: 3/1/2021

11: AS/DS/AaDS AI won
12: AS/DS/AaDS AI won
13: AS/DS/AaDS AI won
14: AS/DS/AaDSAI won
15: AS/DS/AaDS AI won
16: AS/DS/AaDS AI won
17: AS/DS/AaDS AI won
18: AS/DS/AaDS AI won
19: AS/DS/AaDS AI won
20: AS/DS/AaDS AI won

Date: 3/1/2021

11: AS/DS/AaDS AI won
12: AS/DS/AaDS AI won
13: AS/DS/AaDSAI won
14: AS/DS/AaDS AI won
15: AS/DS/AaDS AI won
16: AS/DS/AaDS AI won
17: AS/DS/AaDS AI won
18: AS/DS/AaDS AI won
19: AS/DS/AaDS AI won
20: AS/DS/AaDS AI won

Note: the winning AI will be highlighted in yellow; the mode will be highlighted green

AS AI	DS AI
Overall Count (pts.): -1	Overall Count(pts.): -20

Note: the winning AI will be highlighted in yellow; the mode will be highlighted green

AaDS AI
Overall Count(pts.): 21

Results Summary (TABLE)

Graph 1

Data Analysis and Results

Graph #1	Mode/ Method	ASAI and DSAI	DSAI and ASAI	ASAI and AaDSAI	DSAI and AaDSAI	AaDSAI and DSAI	AaDSAI and ASAI	Win/ Lose Avg.	Total Pt.
AI	×80 ea.	×20	×20	×20	×20	×20	×20	×20	×240
ASAI		9	-2	6			-14	-0.0125	-1
DSAI		-9	2		-1	-12		-0.25	-20
AaDSAI				-6	1	12	14	0.25	21

Graph #2	AI	ASAI	DSAI	AaDSAI
F/S	Total	-1	-20	21
First in Cycle	×80 ea.	15	1	26
Second in Cycle	×80 ea.	-16	-21	-5

Graph 2

Mode-Specified Results

Mode: "ASAIandDSAI"

Tie	ASAI Won
ASAI Won	DSAI Won
Tie	DSAI Won
ASAI Won	ASAI Won
ASAI Won	ASAI Won
ASAI Won	ASAI Won
DSAI Won	ASAI Won
DSAI Won	ASAI Won
ASAI Won	ASAI Won
Tie	ASAI Won

Mode: "DSAIandASAI"

Tie	DSAI Won
DSAI Won	ASAI Won
DSAI Won	ASAI Won
DSAI Won	Tie
ASAI Won	ASAI Won
ASAI Won	DSAI Won
Tie	DSAI Won
DSAI Won	DSAI Won
Tie	Tie
ASAI Won	Tie

Mode: "ASAIandAaDSAI"

AaDSAI Won	ASAI Won
Tie	ASAI Won
ASAI Won	ASAI Won
AaDSAI Won	ASAI Won
Tie	Tie
AaDSAI Won	ASAI Won
ASAI Won	AaDSAI Won
ASAI Won	AaDSAI Won
ASAI Won	ASAI Won
Tie	ASAI Won

Mode: "DSAIandAaDSAI"

Tie	AaDSAI Won
Tie	Tie
Tie	Tie
AaDSAI Won	DSAI Won
Tie	Tie
AaDSAI Won	Tie
Tie	Tie
DSAI Won	AaDSAI Won
Tie	DSAI Won
Tie	Tie

Mode: "AaDSAIandASAI"

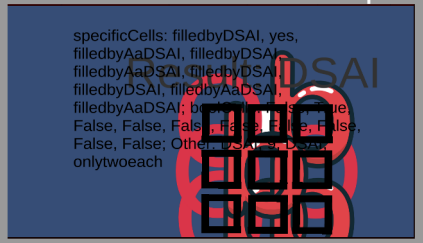
AaDSAI Won	Tie
AaDSAI Won	Tie
AaDSAI Won	AaDSAI Won
Tie	Tie
AaDSAI Won	AaDSAI Won
AaDSAI Won	AaDSAI Won
Tie	AaDSAI Won
AaDSAI Won	AaDSAI Won
AaDSAI Won	AaDSAI Won
AaDSAI Won	Tie

Mode: "AaDSAIandDSAI"

AaDSAI Won	Tie
AaDSAI Won	AaDSAI Won
AaDSAI Won	AaDSAI Won
AaDSAI Won	AaDSAI Won
Tie	Tie
Tie	DSAI Won
AaDSAI Won	AaDSAI Won
AaDSAI Won	AaDSAI Won
Tie	Tie
AaDSAI Won	AaDSAI Won

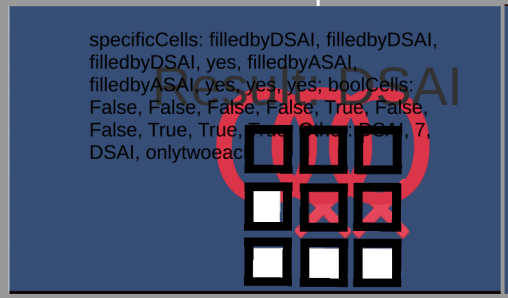
Example Results

Gamemode: AaDSAI|DSAI

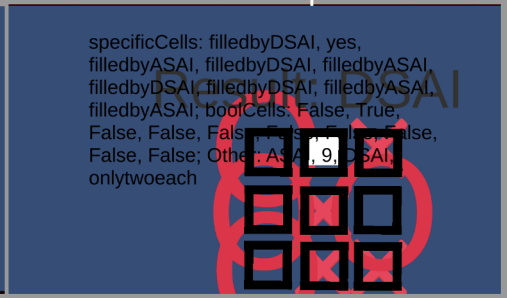


Rare/“Accidental” Win (DSAI)

Gamemode: DSAI|ASAI

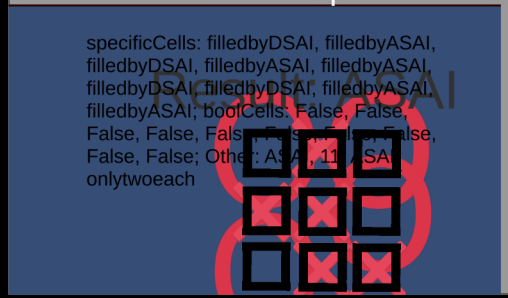


Gamemode: ASAI|DSAI



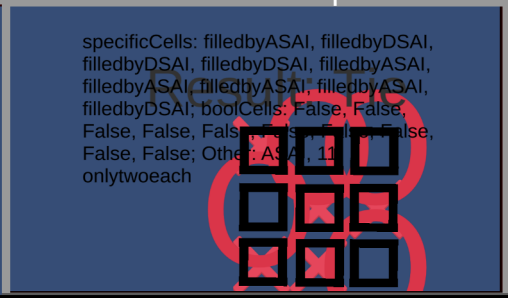
Win (ASAI)

Gamemode: ASAI|DSAI

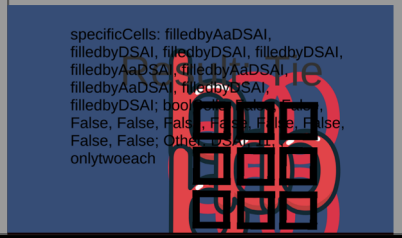


Tie(ASAI|DSAI)

Gamemode: ASAI|DSAI

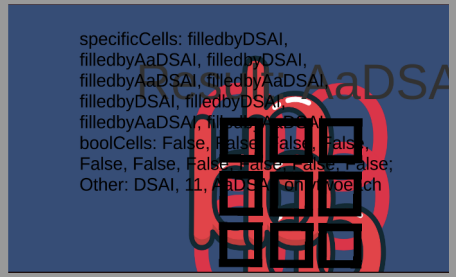


Tie (AaDSAI|DSAI)

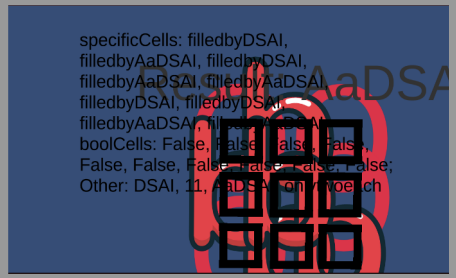


Win (AaDSAI)

Gamemode: AaDSAI|DSAI



Gamemode: DSAI|AaDSAI



Conclusion

In conclusion, based on my results, the AaDSAI has the most points and win/lose average, thus we should target attack/defense strategy competitive AIs as we move forward. As I hypothesized, this was correct, since it provides multiple functions for multiple concurrencies in a normal tic-tac-toe game, instead of limited fundamentals of a competitive/logical game. Some very important variables we should consider when creating competitive AI logic systems are which system/AI goes first in an experiment associating with logical games, such as Tic Tac Toe, Chess, and many others. This significantly changed the results of which AI wins, as shown in the graph 2 of the Results slide. Another variable we should consider is that humans do not have a constant skill level, so in experiments that are associated with competing with an AI, the AI's skill evaluation will be very inaccurate. Though I have concluded based on my data analysis, there may be limitations to my results, as there may be unintentional bugs, though it occurred rarely in my software. To be safe, I did not count the occasions.

Future Studies

In future studies, more advanced topics can be explored, such as number character recognition, using various conditionals for searching a specific number character. There can be image simplification of turning the input images into black/white images, then doing image comparison of the amount of pixels matching/the total resolution to the dataset image. Each of the 9 input images is defined as a list of bool variables representing each pixel (black = true; white equals false), can be implemented. The ideas and methods are limitless. Image processing through learning with a dataset/comparing the dataset image to the input image, and image processing without learning with specific code for each digit methods can be compared.

Bibliography

[1.0]: HCAI (Human-Centered.AI)

[1.1]: neuralnetworksanddeeplearning.com/chap1

[1.2]: Neural Network image: cs231n by Stanford

[2]: purnasai gudikandula: A Beginner Intro to Neural Networks

[3]: towards data science.com- “Everything you need to know about ‘Activation Functions’ in Deep learning models”