Global Journals Inc.

Global Journal of Computer Science and Technology: D
Neural & Artificial Intelligence
Volume 18 Issue 2 Version 1.0 Year 2018
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

The Role of Shologuti in Artificial Intelligence Research: A Rural Game of Bangladesh

By Samia Nawshin & Mohd. Saifuzzaman

Daffodil International University

Abstract- Shologuti has three major component: move generation, search and evaluation. Each components are pretty much necessary, though evaluation with its quiescence analysis is the main part which makes each program's play unique. To make this game more striking, most reliable algorithms and its many supporting aids are used here. Main components of the game tree search and pruning are analyzed here and the performance refinements such as aspiration variation search, assists like transposition and history table are compared here.

Keywords: *artificial intelligence. design, game, extinct, shologuti.*

GJCST-D Classification: *I.2.1*

The Role of Shologuti in Artificial Intelligence Research: A Rural Game of Bangladesh

Samia Nawshin[α] & Mohd. Saifuzzaman[σ]

Abstract- Shologuti has three major component: move generation, search and evaluation. Each components are pretty much necessary, though evaluation with its quiescence analysis is the main part which makes each program's play unique. To make this game more striking, most reliable algorithms and its many supporting aids are used here. Main components of the game tree search and pruning are analyzed here and the performance refinements such as aspiration variation search, assists like transposition and history table are compared here.

Keywords: artificial intelligence. design, game, extinct, shologuti.

I. Introduction

Shologuti is very famous and popular game in Bangladesh and some of the Indian regions. It actually a checker type game which basically played between two players. So from that view, it can be called as a rural version or innovation of chess. The local people play this game for passing their leisure period even sometimes they arrange tournament of this game. Generally the board of this game is very simple like the people can draw it in table or ground and use stones, dry branch or a small piece of cane as pawns. After drawing, two players take their place at the opposite direction of the board with their 32 guti's altogether of which everyone try to achieve opponent's sixteen beads. After placing 16 pawn from the edge of the board, the game starts by making a move on the empty space. The pawn can move right, left, forward, backward and crosswise where each opponents actually try to take or occupy others pawns. If any player's pawn can cross opponent's pawn, then the pawn will be out of the board. After that, only that pawn has the ability to have a bonus move if that pawn can find out opponents pawn can be crossed over. By this process, which player can take or capture all the pawns of opponents, he or she will be the winner.

Now a days, computer has become a part of our life and almost everyone like to play game in it. This led us to develop this attractive game of Shologuti. Due to lack of well-designed boards in market, city's people can't play it or don't even actually well known about the game. So, the problem can be solved by developing this game with an attractive UI. After designing the basic part of this research [16], we are intended to push artificial intelligence part in this program to make it more fascinating.

II. Literature Review

In general, this research cover the literature review from different sources which is targeted various aspect of designing game, installation & learning.

Whitby et.al [1] concerned in his research about the creation of a game installation and a reviewed a case study to test the game. Coleman et.al [2] described the initiatives they took to develop & design a game for improving their skills. Symborski et.al [3] & Martey et.al. [5] also showed the game development and design strategy guided by experimental test and evaluation. Sethi et.al [4] described "Don't just play a game, design a game & build a game and improve your programming skills". Amory et.al [9], Cheng et.al [10] & Dondlinger [12] developed a game which is actually a part of social & mental development. Schrier et.al [11] designed a game which was filled with moral scenarios & moral choices. Nergiz et.al [13] & Strzalkowski et.al [14] described the experiences from the implementation of a computer game development that it could help a student to improve in problem solving, the application of previously learned knowledge, the utilization of independent learning and learning by doing.

Then Reidl et.al. [15], Thune et.al. [16], Nasr et.al. [17] and Harrison et.al. [18] described in their research that Artificial intelligence has long been an important part of game design and development practices. Treanor et.al. [19], Eladhari et.al. [20] and Magerko et.al. [21] identified design patterns for AI in games like AI-based games put AI in the foreground of the player experience in their research.

After that, we (Nawshin et.al. [22]) developed and reviewed the development part of Shologuti in our research document where we wanted to describe and analyze a computer based game (Player vs Player) which will help our culture to save an extinct rural game.

So, in extension, we are intended to use more interesting algorithms to develop the AI vs player environment which will increase the performance to make this game more accurate, unique and enjoyable.

In the coming chapters we will discuss about system architecture of the developed game in Chapter III, Chapter IV deals about the physical implementation and results and finally Chapter V shows us the conclusion.

Author α σ: Lecturer, Dept. of CSE, Daffodil International University, Dhaka, Bangladesh. e-mails: samia@daffodilvarsity.edu.bd, saifuzzaman.cse@diu.edu.bd

III. System Architecture

To explain the entire system as well as the game, an effective architectural system needs to be designed. This section will represent the basic design part. The control flow of the program among various functions is shown using a single flowchart. Each function in the program is well defined and their pseudo codes are given for better understanding of the program. This section contains Control flow diagram, Function description and Pseudo code which will represent a clear understanding about move generation, search and evaluation part of this game.

a) Control flow diagram

PvAI Algorithm Workflow in fig 1 shows the entire execution sequence of functions in the program. In this program, two players can play continuously that is PvP (Player vs. Player) which is already developed [16]. In this game, a single player can also play by a good contribution of AI.

b) Function description

i. Handover

Handover is the starting function for AI part of the game. When the player complete his turn handover function is called. First of all this function clears the array named points to make it ready to use again. Then it calls AI_Moves, Sorting, sets up AICsp (the starting position of AI's best move) and AICep (the ending position of AI's best move) based on the sorting and finally AI_Pawn_Loc.

Pseudo code:

```
Step 1: Clear array "Points"
Step 2: Call AI_Moves ()
Step 3: Call Sorting()
Step 4: Set
        AICsp = points[0,0,0]
        AICep = points[0,1,0]
Step 5: Call AI_Pawn_Loc(AICsp, AICep)
```

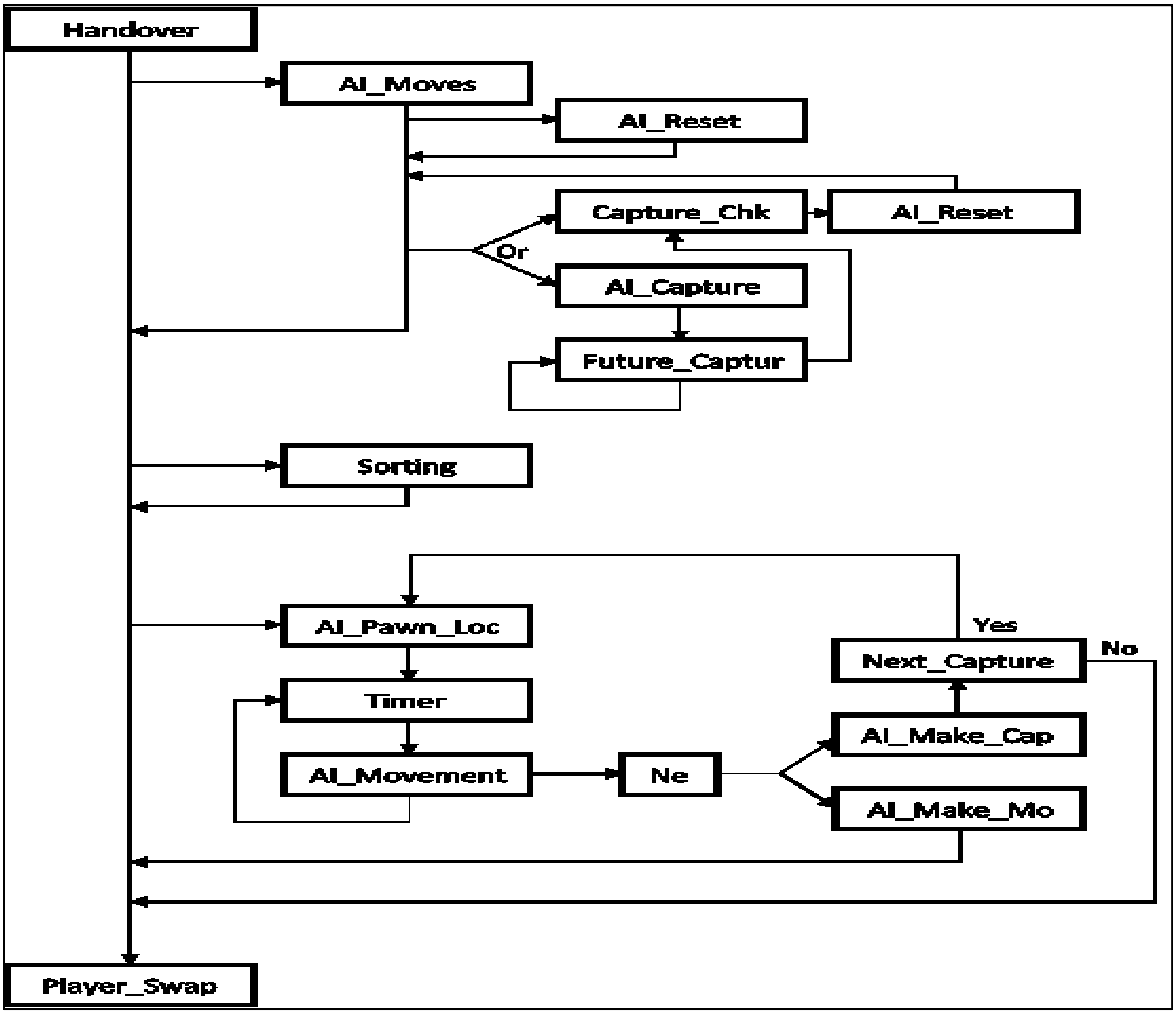


Fig. 1: PvAI Algorithm Workflow

ii. *AI_Moves*

AI_Moves function along with some other functions acts as the mini-max mechanism of the game. This function particularly focuses on each available AI's pawns adjacent positions. First it checks for all the available pawns on the board and store them in the array. Then takes a pawn each time, and searches its adjacent positions. Checks each adjacent positions value. If the value is 0 (represents open position). It stores that position as possible end point on array and calls Capture_Check with parameter as that position. if the adjacent position is 1 (represents enemy pawn), it calls AI_Capture function with parameter of starting position and that position. The function ends when each adjacent position for each pawn is calculated and points are distributed.

Pseudo code:

```
Step 1: Call AI_Reset()
Step 2: Set x=0
Step 3: For every position on the board
    Set z=0
    Set points[x,0,z]=current position
Step 4: For every current position
    Check every adjacent positions value
Step 5: If value of current adjacent position is = 0
    Set points[x,1,z] = current adjacent position
    Call Capture_Check(points[x,0,z], points[x,1,z])
    z=z+1, AI_Reset()
Step 6: If value of current adjacent position is 1
    Call AI_Capture with parameter points[x,0,z] and points[x,1,z]
    z=z+1, AI_Reset()
Step 7: x=x+1
```

iii. *AI_Reset*

AI_Reset function is very important function for the AI part of the game. To calculate and distribute points to each possible moves to evaluate the best move, we have to manipulate the variables that represents pawn on the board. But original variables can't be manipulated directly or the game would become a chaos. So a dummy board or replica is needed. This function replicates the board and lets the needed change happen for calculation. This provides with the ability to calculate using the same info as the board but not change it in the process.

Pseudo code:

- Copy every positions value to similar AI variables | ai0 = a0, ai2 = a2 etc.

iv. *Capture_Check*

This function is the first function that sets points as the mini-max mechanism. This function checks if an AI pawn is moved to a new position, is there any chance that it would be captured by an enemy pawn or not. If there is a chance the pawn may be captured, this will subtract 4 points from the array for that move. If not, it will add three points into it.

Pseudo code:

```
Step 1: Check for each possible enemy capture for this move.
Step 2: If there is a possible enemy capture
    Set points set points[x,2,z] = points[x,2,z] – 4;
Step 3: Else Set points [x,2,z] = points[x,2,z] + 3;
    Call Future_Capture with parameter[x,1,z]
```

v. *AI_Capture*

This is another function of the mini-max mechanism. This function takes the starting and the adjacent enemy position and checks for the third position's value to be 0 (represents open location). If the 3rd position is open, it adds 10 points in the array and calls Future_capture function with the 3rd position as parameter.

Pseudo code:

```
Step 1: Get starting and capture position from AI_Moves
Step 2: If the third position on the line is open
    Set points set points[x,2,z] = points[x,2,z] + 10;
    Set points[x,1,z] = Third Position
    Call Future_Capture with parameter points[x,1,z]
```

vi. *Future_Capture*

This function is used to detect any possible bonus capture moves. As per Sholo-Guti rules, a player can make several capture moves at a time with a single pawn if possible. If possible it adds 5 points to the array for that move, and recourses using the future endpoint for this bonus move. The recursion continues till there is no more bonus moves possible and then calls the Capture_Check function with the parameter of captured location (FSP) and ending position (FEP) it gets at the final recursion.

Pseudo code:

```
Step 1: Set FSP = null & FEP = null
Step 2: Check if any capture move is possible
Step 3: If possible,
    Set FSP = Capturable enemy pawn position
    Set FEP = Next open position on the line
     Call Future_Capture with parameter FEP
Step 4: If not,
    Call Capture_Check with parameter FSP and FEP
```

vii. *Sorting*

After the mini-max part is done, the array is full with each possible move and their respective points. The array then is sorted. Firstly the best move for each pawn is sorted out and brought to the top, i.e. z=0 of the array. And then the best move between all the pawns is sorted and brought to the beginning column i.e. x=0 of the array. After this process the first column of first page of the array holds the best moves starting position, ending position and its points. The array is string type array so to manipulate numeric value. After this the control returns to handover function.

Pseudo code:

```
➢ Sorts the array using Bubble short technique
  and ascends the best move based on the
  points assigned during previous function
  operation.
```

viii. *AI_Pawn_Loc*

AICsp is usually set as the first starting point and AICep is set as first ending point on the array by the function Handover. In case of multiple capture, this function is called again using modified AICsp and AICep. This function then finds out the X & Y coordination for both the AICsp and AICep positions and put them in SPx, SPy, EPx and EPy accordingly. Then it starts the timer function.

Pseudo code:

```
Step 1: Get AICsp, AICep
Step 2: Set
    SPx = X Co-ordination of AICsp
    SPy = Y Co-ordination of AICsp
    EPx = X Co-ordination of AICep
    EPy = Y Co-ordination of AICep
Step 3: Start timer function
```

ix. *Timer*

Timer function is used to recourse a function after a fixed time. Each time timer is activated it calls AI-Movement function till SPx = EPx and SPy = EPy. When the condition is met, timer stops and calls function Next.

Pseudo code:

```
Step 1: Call AI_Movement()
    After certain time till SPx = EPx and SPy = EPy
Step 2: Stop timer, call Next()
```

x. *AI_Movement*

AI_Movement function is responsible for the movement of AI pawn. It detects which pawn to move from and to where and which direction. It increases or decreases SPx & SPy by 4 each and moves the pawn to new SPx & SPy, time till it matches EPx & EPy. This creates the movement of AI's pawn.

Pseudo code:

```
Step 1: If SPx < EPx
    SPx = SPx + 4
    Else
    SPx = SPx - 4
Step 2: If SPy < EPy
    SPy = SPy + 4
    Else
    SPy = SPy – 4
```

xi. *Next*

After the mini-max mechanism and sorting happened, AI has the best move to make. Timer function moves the pawn to the desired position but the value of the board has to change. AI_Make_Move, AI_Make_Capture functions make that happen. After these functions are called and executed, the control again returns to next function and player swap function is called to swap the program control to the pvp part of the game for player to make his move. And the game progresses.

Pseudo code:

```
Step 1: Call AI_Make_Move(AICsp,AICep)
Step 2: Call AI_Make_Capture(AICsp,AICep)
Step 3: Call Player_Swap()
```

xii. *AI_Make_Move*

AI_Make_Move function is used to change the value in board when AI makes a normal move.

Pseudo code:

```
Step 1: Move pawn from AICsp position to AICep
        position
Step 2: Set AICep positions value as AICsp positions
        value
Step 3: Set AICsp positions value as default 0.
```

xiii. *AI_Make_Capture*

AI make capture function is used when AI makes a capture move. It sets AICep positions value as AICsp positions value. And AICsp positions value is set as default 0. The position in between value is set as default 0. Captured pawn count is increased by 1 and enemy pawn count is decreased by 1. Then the function calls Next_Capture function using parameter AICep.

Pseudo code:

```
Step 1: Move pawn from AICsp to AICep
Step 2: Hide pawn between AICsp and AICep
Count enemy pawn = pawn – 1
Step 3: Call Next_Capture using parameter AICep
```

xiv. *Next_Capture*

Next_Capture function checks each adjacent position of AICep position for enemy pawn, if found, it checks if capture move is possible. If possible it sets AICsp as AICep, AICep as new ending position, and calls function AI_Pawn_Loc using these parameter. This process keeps repeating till there is no more capture move possible. And returns control to function next.

Pseudo code:

```
➢ If any more capture possible
Set
AICsp = AICep
AICep = Ending Position
Call AI_Pawn_Loc() using parameter AICsp and
AICep.
```

IV. Implementation and Results

a) *Selecting the Game Type*

When the game starts if it detects a previously saved game, the game is loaded with previously saved properties. If not, a window shows up like this and asks the player to select a game type, either a PvP or PvAI.

Fig. 2. is the screenshot of Selecting the Game Type.

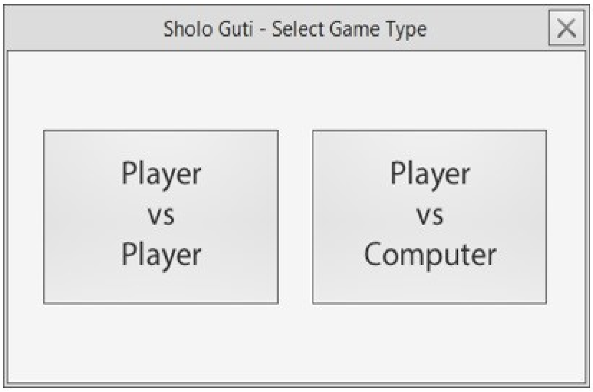


Fig.2: Selecting the Game Type

b) *Player vs. Computer*

If the player selects to play a new game of PvAI, the board is loaded with pawns set in their default starting position and then the player gets to play the first move always. When a move is made by the player, AI will take a move automatically. Fig.3. is the screenshot of Player vs. Computer.

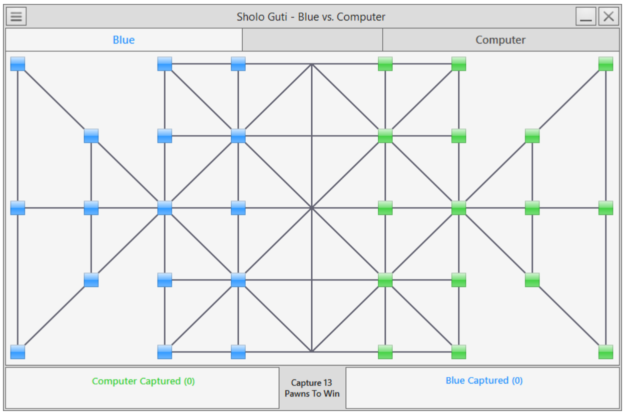


Fig. 3: Default Board - Player vs. Computer

c) *Pawn Movements*

i. *Player Pawn*

If the player selects a pawn of his own, then possible moves are shown in the board in Yellow and Red. Yellow color represents normal movable positions and Red color represents capturable positions on the board. Fig.4. is the screenshot of Pawn Movements of Player Pawn.

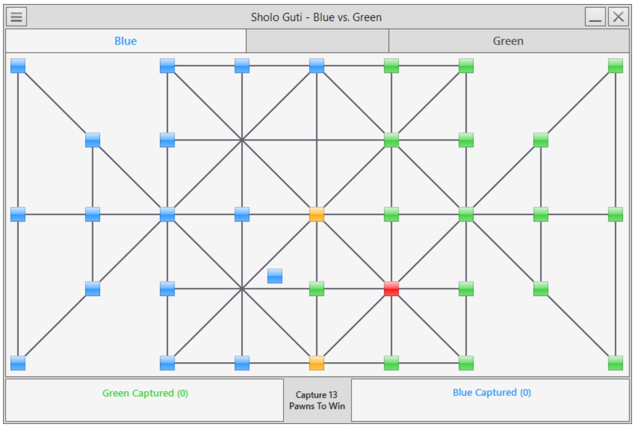


Fig. 4: Pawn Movements - Player Pawn

ii. *AI Pawn*

When a move is made by the user AI will make a move to best possible location among all of its own pawn and after making a move AI will hand over the game to the player.

If there is any capture move possible AI will take that move. And then if another capture is possible with that pawn again, AI will take that move too and will be looking to capture more pawns with that pawn if possible. Else, AI will hand over the game to the player. Fig.5. is the screenshot of Pawn Movements taken by AI Pawn.

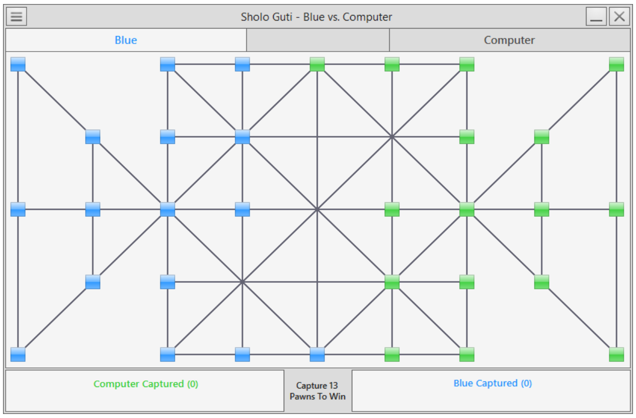


Fig. 5: Pawn Movements – AI Pawn

d) *Drop-Down Menu*

On top left of the UI there is a button which shows a dropdown menu if clicked. The menu has different options and submenus to customize game properties. Fig.6. is the screenshot of the game window with Drop-Down Menu.

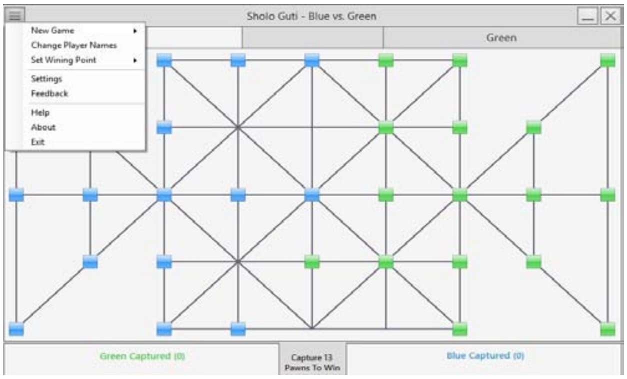


Fig. 6: Drop-Down Menu

e) New Game

On dropdown menu there is a submenu named as "New Game" which contains two options to start a new game, "Player vs. Player" and "Player vs. Computer". On the click to one of those menu option specific game starts with default board. Fig.7. is the screenshot of how to create a new game.

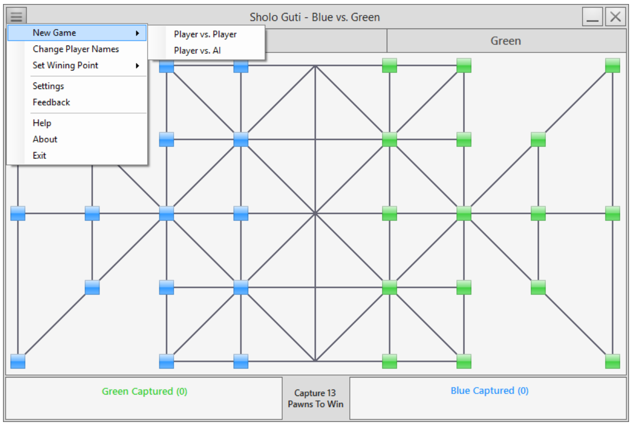


Fig. 7: New Game

f) Setting Wining Point

On dropdown menu there is a submenu named as "Set Wining Point". On the hover to this control, it shows five different numbers. Clicking on any of them sets particular value. Which indicates how many pawns have to be captured by users/computer to win the current game. Fig.8. is the screenshot of how to set wining point on a game.

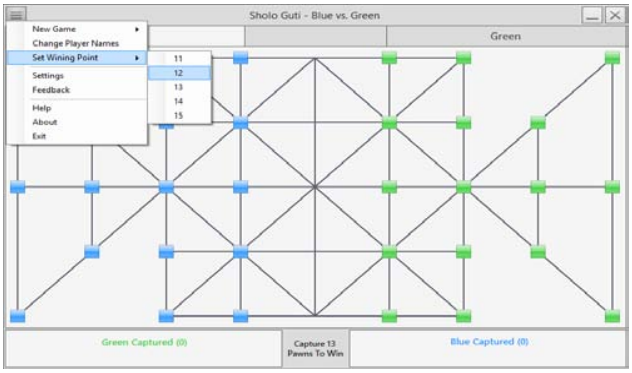
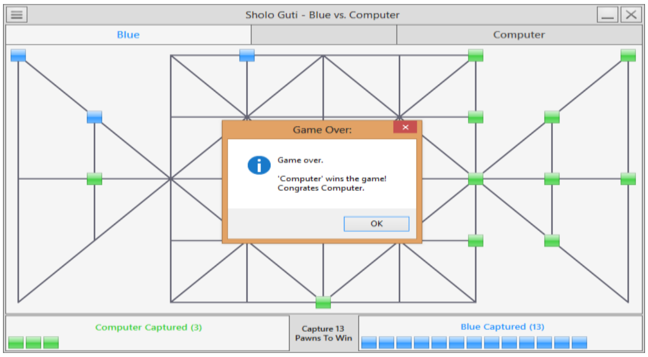


Fig. 8: Setting Wining Point

g) Winning the Game

When the specified numbers of pawn is captured by player/ computer, a message box appears on the screen showing the winners name on it. And then another message box is showed up if the player/players wants to start a new game on the current game mode. Fig.9. is the screenshot of what happens when a game is won by the AI and fig.10. is the screenshot of what happens when a game is finished.

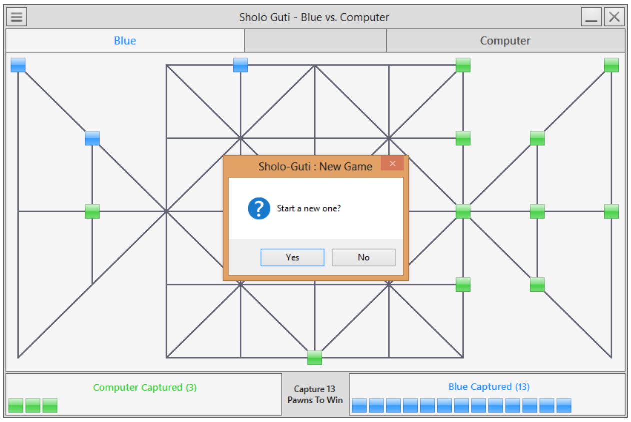


Game Over:

Game over.

'Computer' wins the game!
Congrates Computer.

Fig. 9: Wining the Game – Game Over



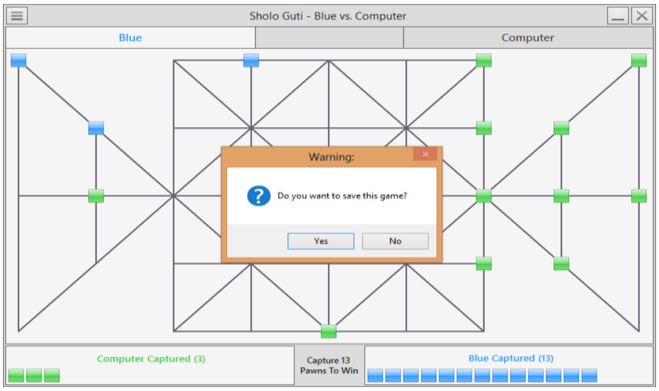
Sholo-Guti : New Game

Start a new one?

Fig. 10: Wining the Game – New Game

h) Save & Exit

When the close button on the top right is pressed or "Exit" is clicked from the dropdown menu, a message box appears to ask player if he wants to save the current game to play it later from the current state. If the button “Yes” is clicked then the game will be saved with current properties. And user can play the game from that saved state when he runs the game next time. Fig.11. is the screenshot of what happens when a close button is pressed during the gameplay.



Do you want to save this game?

Fig. 11: Save & Exit

V. Conclusion

Different person thinks in different way. This game is developed from a specific point of view and analysis. For developing or designing this game, many problems or difficulties actually came out. So there is a chance that the problems can be solved in a much more easy way by using advanced techniques.

a) *Challenges to overcome*

It is true that the development environment is totally unpredictable. Many things went wrong while working on the real project. Firstly, any well-defined structure of the game haven’t found. Secondly, Some problems arose while developing the system like creating database cost us more time fetching data from database for this a three dimensional array is used instead of using database. Finally, faced some problem in .NET platform but managed to fix it all.

b) *Future scope*

We need to apply more algorithms in future to make it more efficient and effective. In future this research can be enhanced into a more intelligent system.

References Références Referencias

1. Whitby, M. (2017) “Designing and Creating a Game Installation”, The Computer Games Journal, 6(3), 85–109. Doi: 10.1007/s40869-017-0039-2
2. Coleman, R., Krembs, M., Labouseur, A. & Weir, J. (2005) “Game design & programming concentration within the computer science curriculum”, ACM SIGCSE Bulletin, 37(1), 545-550, doi:10.1145/1047124.1047514
3. Symborski, C., Barton, M., Quinn, M.M., Korris, J.H., Kassam, K.S., & Morewedge, C.K. (2017), “The Design and Development of Serious Games Using Iterative Evaluation”, SAGE Journals: Games and Culture, 12(3), 252-264.doi: https://doi.org/10.1177/1555412016673262
4. Sethi, M. (2008), “Game Programming for Teens, 3rd Edition (Computer Game and Simulation Programming)”. Retrieved from: https://www.amazon.com/Game-Programming-Teens-Computer Simulation/dp/1598635182.
5. Martey, R.M., Galley,J.S., Shaw, A., McKernan,B., Saulnier, T., Mclaren, E., Rhodes, M., Folkestad, J., Taylor,S.M., Kenski,K., Clegg, B., & Strzalkowski, T. (2016), “Balancing Play and Formal Training in the Design of Serious Games”, SAGE Journals: Games and Culture,12(3),269-291.
6. Flores,E., Tobon,G., Cavallaro,E., Cavallaro,F.I., Perry,J.C. & Keller,T. (2008), “Improving patient motivation in game development for motor deficit rehabilitation”, ACE '08 Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology, 381-384
7. Goude, D., Björk, S., Rydmark, M. (2007).“Game Design in Virtual Reality Systems for Stroke Rehabilitation”, Stud Health Technol. and Inform. 125:146-148.
8. Cameirao, M. S., Badia, S. B., Zimmerli, L., Oller, E. D., Verschure, P. F. M. J., (2007),“The Rehabilitation Gaming System: a Virtual Reality Based System for the Evaluation and Rehabilitation of Motor Deficits.” Virtual Rehabilitation, 2007 (September 27-29) 29-33. doi: http://doi.acm.org/10.1109/ICVR.2007.4362125
9. Amory, A., Naicker, K., Vincent, J., & Adams, C. (1999),“The use of computer games as an educational tool: Identification of appropriate game types and game elements”, British Journal of Educational Technology 30(4),311–321.doi:10.1111/1467-8535.00121
10. Cheng, C.H., Ho Su, C. (2012),“A Game-based learning system for improving student's learning effectiveness in system analysis course”, Procedia - Social and Behavioral Sciences, 31,669-675. doi: https://doi.org/10.1016/j.sbspro.2011.12.122
11. Schrier,K. (2017), “Designing Games for Moral Learning and Knowledge Building”, SAGE Journals: Games and Culture. Doi: https://doi.org/10.1177/1555412017711514
12. Dondlinger, M.J. (2007), “Educational Video Game Design: A Review of the Literature”, Journal of Applied Educational Technology, 4(1). Retrieved from: https://www.researchgate.net/publication/238444705_Educational_Video_Game_Design_A_Review_of_the_Literature.
13. Cagiltay, N.E. (2007),”Teaching software engineering by means of computer-game development: Challenges and opportunities”, British Journal of Educational Technology, 38(3), 405–415.

14. Strzalkowski, T., & Symborski, C.(2016),"Lessons Learned About Serious Game Design and Development", SAGE Journals: Games and Culture,12(3), 292-298. Doi: https://doi.org/10.1177/1555412016673524
15. Riedl, M.O. & Zook, A, "AI for game production" in Computational Intelligence in Games (CIG), IEEE Conference, 2013
16. D. Thue, V. Bulitko, M. Spetch, and E. Wasylishen, "Interactive storytelling: A player modelling approach," in 3rd AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2007
17. M. Seif El-Nasr, A. Drachen, and A. Canossa, Eds., Game Analytics. Springer London, 2013
18. B. Harrison and D. L. Roberts, "Using sequential observations to model and predict player behavior," in 6th International Conference on the Foundations of Digital Games, 2011
19. Treanor et.al. "AI-based game design patterns" in 10th International Conference on the Foundations of Digital Games 2015
20. Eladhari, M.P. et al. 2011. {{AI}-Based} Game Design: Enabling New Playable Experiences.
21. Magerko, B, Permar, J, Jacob, M, Comerford, M, Smith, J. 2014. An Overview of Computational Co-creative Pretend Play with a Human. Proceedings of the Playful Characters workshop at the Fourteenth Annual Conference on Intelligent Virtual Agents (2014)
22. Samia, N., Saifuzzaman, M., Abdullah Al Mamun, M. and Al Muzahid, M. (2017), "Design and Implementation of Shologuti: A Rural Game of Bangladesh" on International Journal of Computer Applications 176(8), 20-25, DOI:10.5120/ijca2017915622

Global Journals Inc.

Global Journal of Computer Science and Technology: D
Neural & Artificial Intelligence
Volume 18 Issue 2 Version 1.0 Year 2018
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Texture Feature Abstraction based on Assessment of HOG and GLDM Features for Diagnosing Brain Abnormalities in MRI Images

By Sudheesh. K. V & L. Basavaraj

Vidyavrdhaka College of Engineering

Abstract- The brain tumors are increasing rapidly among the younger generation. The survival of the subject can gradually be increased if the tumors are detected at early stages. Magnetic Resonance Imaging (MRI) is an important technique in detecting the tumors. The images are corrupted by random unwanted information, complicating the automatic feature extraction and the analysis of clinical data. Many methods are existing in present day to remove the unwanted information from the images. Automatic classification is essential because it reduces the cause of human error and where the accuracy is not affected. The work emphasizes on removal of noises from the MRI using the hybrid KSL technique which is the combination of Kernel, Sobel and low pass filter. Features are the properties which describe the whole image. Features from these images are extracted using shape, texture and intensity based techniques. The feature extracted are HOG and GLDM.

Keywords: histogram of gradient, gray level difference method, feature extraction.

GJCST-D Classification: I.4.1