

Artificial Intelligence & Machine Learning(C09.307) - Бие Даалт 2

Тайлан

Гүйцэтгэсэн: Б.Наранбат

Хоёрдугаар бие даалтын хүрээнд Java программчлалын хэл дээр Java Swing хэрэглэгчийн интерфэйсийг график орчинд хөгжүүлэн гүйцэтгэв. Үүнд JFrame class -т “extend” хийн нэг файл дотор бүх үйлдлийг хийв. Программ нь 3x3 бүтэцтэй char төрлийн 2 хэмжээст массиваар O болон X тэмдэгтийг бөглөж босоо 3, хэвтээ 3, диаголоор 3 дараалласан тэмдэгт массивт ороход эсвэл 3x3 массив дүүрсэн тохиолдолд тоглоом дуусна. Программ ажиллаж эхлэхэд:

humanPlayer тэмдэгт хувьсагчаар X, aiPlayer тэмдэгт хувьсагчаар O тэмдэгтүүдийн утгыг өгөх ба humanPlayer эхний үйлдлийг хийнэ. Мөн үндсэн талбарыг буюу 2 хэмжээст массивын хэмжээг 3,3 гэж өгөх ба бүх утгуудыг “-” тэмдэгээр дүүргэнэ. Үүний дараагаар GUI ажиллуулах initComponents функцийг дуудна.

Талбарт mouse 1 дарах үед тухайн талбарт харгалзах массивын утга “-” тэмдэгттэй тэнцэх ба isGameOver() функцийн буцаасан утга false үед массивт “X” тэмдэгтийн утгыг өгч тухайн талбарын шошгын зургийг “x.jpg” -аар set хийнэ.

Үүний дараагаар aiMove() функцийг дуудах ба isGameOver() функцийн буцаасан утга false үед minimax(int depth, boolean isMaximizing) функцийг дуудаж, буцаасан массивын 0 болон 1 индекс дэх тоонууд компьютерын нүүдлээр үндсэн талбарын массивт “O” гэсэн тэмдэгтийг өгнө. Мөн харгалцах талбарын шошгын зургийг “o.jpg” -аар set хийнэ. isGameOver() функцийг дахин дуудаж true үед getWinner() функцийг дуудаж буцаасан утга

- “-” тэмдэгттэй тэнцүү бол тэнцсэн онооны утгыг нэгээр нэмэгдүүлж “Та тэнцлээ” гэсэн диалог мессеж харуулна.
- “X” тэмдэгттэй тэнцүү бол тоглогчийн онооны утгыг нэгээр нэмэгдүүлж “Та яллаа” гэсэн диалог мессеж харуулна.
- “O” тэмдэгттэй тэнцүү бол компьютерын онооны утгыг нэгээр нэмэгдүүлж “Та хожигдлоо” гэсэн диалог мессеж харуулна.

Дахин тоглох товчийг дарах үед restart() функц дуудагдаж бүх талбарын зургийг “null” болгож талбаруудын массивын бүх утгыг “-” тэмдэгт болгоно.

isGameOver() функц нь isBoardFull() буцаасан boolean хувьсагчийн true эсвэл getWinner() функцийн буцаасан утга “-” тэмдэгтээс өөр үед true утгыг буцаана.

getWinner() функцийн нь үндсэн талбарын массивын эхний хэмжээсээр давталт ажиллаж [i][0] == [i][1] ба [i][1] == [i][2] ба [i][2] == [i][0] ба массивын [i][0] утга “-” тэмдэгтээс өөр үед массивын [i][0] -ийн утгыг буцаана. Нөхцөл биелээгүй үед үндсэн талбарын массивын хоёрдох хэмжээсээр давталт ажиллаж [0][j] == [1][j] ба [1][j] == [2][j] ба [2][j] == [0][j] ба массивын [0][j] утга “-” тэмдэгтээс өөр үед массивын [0][j] -ийн утгыг буцаана. Нөхцөл биелээгүй үед үндсэн талбарын массивын [0][0] == [1][1] ба [1][1] == [2][2] ба [0][0] утга “-” тэмдэгтээс өөр үед массивын [0][0] -ийн утгыг буцаана. Нөхцөл биелээгүй үед үндсэн талбарын массивын [0][2] == [1][1] ба [1][1] == [2][0] ба [0][0] утга “-” тэмдэгтээс өөр үед массивын [0][0] -ийн утгыг буцаана. Нөхцөлүүд биелээгүй бол “-” тэмдэгтийг буцаана.

`minimax(int depth, boolean isMaximizing)` функц нь `int[]` төрлийн утга буцаадаг ба тоглоомын минимакс алгоритмыг ажиллуулж компьютерын нүүх үйлдлийг хийнэ. Минимакс алгоритм нь `гүн(int)` ба `isMaximizing(boolean)` гэсэн хоёр параметрийг авдаг. Гүн гэдэг нь алгоритмын шийдвэр гаргахын өмнө авч үзэх шаардлагатай алхмуудын тоо юм. `isMaximizing` гэдэг нь алгоритм нь одоогийн байдлаар нэмэгдүүлэх эсвэл багасгах хөдөлгөөнийг үнэлж байгаа эсэхийг илэрхийлдэг логик утга юм. Алгоритм тоглогчид давуу тал өгөхгүйгээр үйлдэл хийхийг зорино. Мөн алгоритм нь самбарын одоогийн төлөвөөс хийж болох бүх боломжит хөдөлгөөнийг үнэлэхийн тулд рекурсив аргыг ашиглана. Энэ нь тоглоом дууссан эсэх (жишээ нь, ялагч эсвэл тэнцсэн эсэх) эсвэл хамгийн их гүнд хүрсэн эсэхийг шалгах замаар эхэлнэ. Тоглолт дууссан бол тоглоомын үр дүнд үндэслэн оноог буцаана: компьютер хожвол 10, хүн хожвол -10, тэнцсэн бол 0. Хэрэв хамгийн их гүнд хүрсэн бол алгоритмд цаашид авч үзэх алхам байхгүй тул 0 оноог буцаана. Хэрэв тоглоом дуусаагүй бөгөөд хамгийн их гүнд хүрээгүй бол алгоритм нь самбар дээрх бүх хоосон нүднүүдийн жагсаалтыг `getEmptyCells()` функц дуудаж авна. Хэрэв `isMaximizing` үнэн бол алгоритм нь бүх хоосон нүднүүдийг давтаж, тус бүрд "О" тэмдгийг хийдэг. Дараа нь самбарын үүссэн төлөвийг үнэлэхийн тулд гүн параметрт `depth - 1`, `isMaximizing` параметрт `false` гэсэн утгатай `minimax` функцийг рекурсив байдлаар дууддаг. Рекурсиваар дуудах бүрийн дараа алгоритм нь нүдийг хоосон төлөв рүү дахин тохируулж, үр дүнд нь гарсан алхам нь өмнөх хамгийн сайн нүүдлээс илүү байвал `bestMove` хувьсагчийг шинэчилдэг. Хэрэв `isMaximizing` нь `false` бол, алгоритм нь ижил зүйлийг хийх боловч тоглогч компьютерт давуу тал өгөхгүйгээр үйлдэл хийнэ гэж таамаглаж, үр дүнд нь гарсан нүүдэл өмнөх хамгийн сайн нүүдлээс муу байвал `bestMove`-г шинэчилнэ. Эцэст нь алгоритм нь рекурсивээр ажиллаж олсон `bestMove`-ийг буцаана. `bestMove` нь гурван элемент бүхий массив юм: хамгийн сайн нүүдлийн мөрийн индекс, хамгийн сайн нүүдлийн баганын индекс, энэ нүүдэлтэй холбоотой оноо.

Үр дүн ба дүгнэлт: Энэхүү бие даалтын ажлаар `miniMax` алгоритмыг ашиглаж компьютерын үйлдлийн шийдвэрийг гаргасан ба ашигласан алгоритм нь хэрхэн рекурсивээр ажиллаж компьютерын хамгийн сайн үйлдлийг олж гаргаж байгааг ойлгож мэдэж авав. Мөн Java swing хэрэглэгчийн интерфэйсийг график орчин хөгжүүлэлтийн мэдлэгээ сэргээн бататгав.