**Вештачка интелигенција**

**Оператор промене стања**

                                 Члановитима:

                                                               Ђорђе Петровић, бр. инд. 17872

ЈованаСтојановић, бр. инд. 17980

                                                           Алекса Томић, бр. инд. 18013

Прва функција “inputDim()” служи за унос димензија табле. Димензије табле, “m” и “n”, морају да буду веће од 2, пошто плочица заузима два поља. У супротном би увек победио први играч. Такође димензије морају да буду мање или једнаке од 26, због абецеде која обележава колоне на табли. Ако нису унете валидне димензије штампа се порука и врши се поновни унос димензија. Као вредности враћа димензије табле.

Функција “pocetnoStanje()” прави иницијално или почетно стање табле на основу прослеђених димензија. Празна табла се том функцијом на почетку попуњава са “None”.Функција враћа таблу.

Опис коментара: Параметар “iksOks” је типа “bool” и у зависности од његовог стања одређује се који играч игра. Када је он “False” игра први играч, тј. “X“, у супротном игра играч “О”.

Функција “valjanPotez (mat, iksOks: bool, x, y, m, n)” проверава валидност потеза и прослеђују јој се табла, играч који је на потезу, координате потеза и димензије табле. Услов за координате је да су мање од димензија табле како плочица не би била ван табле. Вредност 26 се узима јер постоје 26 слова абецеде. Ако игра играч “X” попуњава се то поље и поље изнад. Ако игра играч “О” попуњава се то поље и поље десно од задатог. Да би потез био валидан оба поља морају бити празна. Функција враћа bool вредност која нам говори о томе да је потез ваљан када је “True“.

Функција “jedanKarakterProvera (x, y)” проверава да ли је као “х“ координата потеза унета бројчана вредност, односно да ли је као “у“ координата унет један карактер и да ли је карактер у опсегу ASCII вредности за слова. Ако није штампа се одговарајућа порука. Функцији се као параметри прослеђују координате “х“ и “у“, а враћа “-1“ уколико постоји нека невалидност, односно

Функција “potez (mat, iksOks: bool, x, y, m, n)” прво позивом функције “jedanKarakterProvera(x, y)” проверава да ли су координате унете у исправном формату. Затим на основу прослеђених координата проверава да ли су исправне позивом претходне функције “valjanPotez” и штампа поруку ако нису. У наставку функције врши се попуњавање таблице карактерима “X” и “О”. Парамети функције су табла, играч, координате потеза и диментије табле, а функција враћа стање табле након одиграног потеза, уколико је потез валидан, односно “-1“, ако није.

Функција “proveraNijeKraj (mat, iksOks: bool, m, n)” проверава да ли смо дошли до краја партије, тачније да ли играчи имају слободна поља за потезе. Овом функцијом се такође добијају и сви могући потези које тренутни играч може да одигра. Параметри функције су табла, играч и димензије табле. За играча “X” проверава да ли у таблици постоје два поља један изнад другог која су слободна, а за играча “О” проверава да ли постоје два слободна суседна поља. Притом се у променљивој “listPotezi“уписују поља која задовољавају претходно наведене критеријуме у случају да је играч “Х“, односно “О“ на потезу. Враћа се bool  вредност која нам говори да ли смо дошли до краја када је “True”, као и сама листа потеза.

Функција “formatirajTekst (lista, m, iksOks)“ служи само да се у исправном формату прикажу потези које може да одигра тренутни играч. Као параметре има листу у којој се налазе сви могући потези, дужину листе и играча који је на потезу.Сви потези се смештају у променљивој “string1“. Исправан приказ потеза је [<ознакаВрсте>, <ознакаКолоне>], где <ознакаВрсте> треба да се прикаже као број, док је <ознакаКолоне> слово, и то се постиже у for петљи.

Функција “zapocniIgru()” заправо покреће целу апликацију. Зову се функције за унос димензија и постављање почетног стања, након чега ће се одмах приказати почетно стање.

Закоментарисани део може да се користи ако желимо да на почетку одрадимо као који играч ће да игра рачунар, као икс или окс. Даља измена би била да и 164 линији кода поставимо играча на „False“ како би икс играо први. Такође у 172 другој линији услов би био да ли су играч и комп једнаки, тада би се позивао алгоритам и играо би рачунар.

Затим се уноси играч, који је представљен променљивом “igrac“. Могуће опције за унос су x, X, o и O, и све док се не унесе исправна вредност тражи се унос играча. За вредности x или X, на потезу је “Х“, а променљива “igrac“ добија вредност “False“, односно за вредности o или O, на потезу је “О“, а променљива “igrac“ добија вредност “True“.

Извршење осталих функција се врши у while петљи све док не дођемо до краја и док оба играча имају слободна места за потезе. Исписује се који играч је тренутно на потезу и сви његови могући потези. Уносе се координате потеза за сваког играча и зове функција за постављање потеза на таблу.

Ако је услов задовољен игра рачунар. Резултат алгоритма се смешта у променљиву „оптималниПотез“. То је заправо оптимални потез генерисан алгоритмом. Помоћу [0] и [1] приступамо првој или другој координати потеза. Проверава се да ли је тип оптималног потеза листа, ако није за координате се бирају потези из листе могућих потеза. Ако се не задовољава услов, то значи да је на потезу играч, а не компјутер. Остататак кода је исти као пре за унос координата.

Прелази се на следећег играча након завршетка уноса потеза и зове се следећа функција за графички приказ. Након извршења while петље, игра је готова и штампа се играч који је победио.

Функција “grafickiPrikaz (mat, m, n)” врши исцртавање табле. Параметри функције су табла и њене димензије. Променљива “chr123“ служи за приказ колона абецедним редоследом и на почетку има вредност “А“.Испис абецеде изнад и испод колона табле ће се наћи у променљивој “strPrvaLinija“.Функцијом ord() добијамо ASCII вредност карактера и при сваком проласку инкрементирамо га за 1 и тако прелазимо на следећи карактер. Променљива “linijice“ служи за унос знака = са горње и доње стране табле. Приказ бројева и усправних цртица са леве и десне стране табле, као и саме табле врши се у for петљи. Број се добија када се од ширине табле одузме стање бројача петље, пошто се бројеви са стране инкрементирају одоздо на горе. Са десне стране је иста логика, само другачији запис, прво цртице па број. Између приказа број-цртица и цртица-број, додају се поља табле. На тај начин се испитује цела врста табле и њене ознаке.

Функција „realniPotezi(mat, iksOks, m, n)“ приказује број реалних потеза. Број реалних потеза је максимални број потеза које је могуће одиграти у игри без међусобног преклапања. Функционише на начин да у зависности од играча пролазимо кроз врсте или колоне и тражимо потезе за ту врсту или колону. Ако је потез нађен инкрементирамо бројач за два, ако није инкрементирамо за један и проверавамо да ли следећа два поља празна.

Функција „sigurniPotezi(mat, iksOks, m, n)“ изачунава број сигурних потеза. „listaPoteza1” и „listaPoteza2” су листе потеза за Окс и Икс редом. У зависности играча проверавају се да ли у листи противникових потеза постоји потез који би играо на тим позицијама. Ако услов није задовољен, то би значило да су то сигурни потези, инкрементира се бројач. На тим позицијама играч би увек могао да игра, независно од противника.

Функција „heuristika(mat, potezKoor, iksOks, m, n)“ одређује хеуристику потеза и на основу тога алгоритам одређује који играч ће да игра. За почетак копирамо таблицу у резервну таблицу мат2, и у њу се за прослеђени потез уписују потези и рачуна хеуристика за њих. Рачунају се број реалних потеза за нас и противника, као и број сигурних потеза за нас и противника. На основу тога рачуна се хеуристика потеза.

Функција „Puna(mat, m, n)“ проверава да ли је таблица пуна. Пуна је ако ни један играч не може да игра.

Сам алгоритам алфа-бета је сличан алгоритму за икс-окс са предавања. На почетку се проверава да ли је крај игре или је таблица пуна. У листу потеза се смешта листа могућих потеза за играча. Затим се проверава да ли смо дошли до добине и проверава се листа потеза до кад се и извршава овај део алгоритма. Зове се хеуристика за сваки потез. Прво део у for петљи смо писали на тај начин јер се тако код нас уписују подаци у табелу, тог су типа. То је због координата које се у функцији потез наводе као слово и број. За играча за кога је овај алгоритам, игра се максимум. Хеуристика би представљала вредност самог потеза за тог играча. Он тражи максимално добар потез, док би за противника најболји потез био минимум или најлошији потез по тој хеуристици за противника. У зависности од играча, рачунамо максимум или минимум. Одсецање се извршава на крајевима тог дела кода, што убрзава сам алгоритам. Обилази се у просеку око 20% мање чворова.

На крају се позива функција “zapocniIgru()” и покреће се програм.