

Óbudai Egyetem
Kandó Kálmán Villamosmérnöki Kar
Automatika Tanszék



TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

OKOS SAKKTÁBLA

Szerző(k): **Hekli Dániel**
villamosmérnök BSc. szak, III. évf.

Konzulens(ek): **Sándor Tamás**
mestertanár

Budapest, 2022.

Tartalomjegyzék:

1. Bevezetés	3
1.1 Projekt ötlet [1]	3
1.2 Sakkgépek	3
2. Történelmi áttekintés	4
2.1 A játék fejlődése [2][3]	4
2.2 Sakkgépek és programok fejlődése [4]	5
3. Fizikai sakkgépek	6
3.1 Sakkgépek megvalósításai [5]	6
3.2 Ma kapható kiemelkedő sakkgépek [5]	7
4. Az első „sakkgép” [6][7]	8
5. Tervezett megvalósítás	9
6. Modulok	9
6.1 A sakkbábukat érzékelő szenzorok	10
6.2 Sakkfigurák	11
6.3 Egyedi sakktábla (17-18. ábra)	12
6.4 Visszajelző rendszer	13
6.5 Vezérlő perifériák	14
6.6 Sakkóra	15
6.7 Hőnyomtató	16
6.8 Mikrokontroller	17
7. Program megvalósítása	18
7.1 Beállítások	18
7.2 Ember-Ember játszma	18
7.3 Visszajátszó mód	19
7.4 Oktató mód	20
7.5 Ember-Gép játék [8]	21
8. Sakkprogram [9][10]	22
8.1 Játék tér	23
8.2 Lépés generálás	24
8.3 Lépés keresése	25
8.3.1 Minmax - Negamax	25
8.3.2 Alfa-Béta	26
8.3.2 Keresés sorrendje	27
8.3.2 Horizont Hatás	28
8.4 Sakkprogram felhasználása	29
9. Továbbfejlesztési lehetőségek	30
10. Következtetés [5]	31
Irodalomjegyzék	32

1. Bevezetés

Ez a TDK munka a 2021-es őszi továbbfejlesztett verziója.

1.1 Projekt ötlet [1]

Az okos sakktáblát egy az amatőr és haladó versenysakk során felmerülő probléma megoldására tervezem. A játékosok ezen a szinten gyakorta elfelejtik a verseny hevében dokumentálni a lépéseket. Ezen problémát az okos sakktábla oldaná fel, játék közben a táblába épített érzékelők segítségével rögzítené a megtett lépéseket, amiket képes lenne menteni, visszajátszani, vagy akár egy kompakt hordozható hőnyomtatóval a helyszínen kinyomtatni.

Természetesen az ehhez szükséges hardver már más funkciók betöltésére is alkalmas ezért az alap ötlet további bővítésre került. Ha már a rögzíti a lépéseket akkor érdemes lenne megvizsgálni, hogy az adott lépés szabályos-e, ehhez viszont szükséges megvizsgálni, hogy a felemelt bábú tud-e lépni, és ha igen akkor hova. Amennyiben tudom, hogy hova tud lépni akkor ezt visszalehetne adni valamely módon a játékosnak, ami nagy segítség lehet azoknak, akik még csak most ismerkednek a játékkal. Az ehhez szükséges megjelenítő rendszerrel viszont már képes lenne a rendszer egy sakkprogram által megtett lépést is megjeleníteni. Mindezen funkciókkal kiegészítve már egy teljesértékű sakkgépet kapunk. Végezetül még egy sakkóra integrálása is a tervbe került elvégre minden szükséges elem a rendelkezésre áll hozzá, pusztán egy megjelenítő és 2 gomb hozzáadása szükséges, és így nem kell külön sakkórát is szállítani.

1.2 Sakkgépek

Már évtizedek óta verik el az embereket a gépek, a mai sakkgépek már annyira erős ellenfelek, hogy a sakknagymesterek már nem is próbálkoznak legyőzni őket. A mai processzorok számítási teljesítménye ellen már egy emberi játékosnak nincs esélye, egy kötött szabályrendszerben működő játékban a gépek ellen. Ezért nem is az lenne a projekt célja, hogy egy legyőzhetetlen sakkprogramot készítsek, az már adott. Ehelyett az emberi játék segítése lenne a cél, az önálló vagy edzővel történő gyakorlás, valamint kezdők és haladók számára szórakozási szinten gép elleni játék megvalósítása.

2. Történelmi áttekintés

2.1 A játék fejlődése [2][3]

A sakk eredete egy indiai játékhoz vezethető vissza a chanturanga-ra, aminek múltja még a 600-as évek előttre nyúlik vissza. A sakk őse számos változtatáson esett át amíg elérte a ma ismert formáját, és több változata is született időközben, ilyen például a japán sakk a Shogi (sógi) is, bár ez különbözik a legjobban a nyugati sakktól ugyanaz az őse. A játék minden változatára jellemző, hogy a terület korabeli kultúrája honosodott meg a bábuk alakjában, illetve a szabályokban.

Három fő változási hullám terjedt végig a sakkszerű játékokon, ezek az egyik fő oka, hogy több játékváltozat alakult ki, egyes népek elfogadták és alkalmazták az indiai eredetű változtatásokat mások elutasították, vagy már nem voltak alkalmazhatóak a játék adott változatára, vagy csak saját szabályokat alkottak. Az előbb példának felhozott Shogi-nál csak az első lett elfogadva, míg a nyugati sakkon mindhárom változtatást alkalmazták és ezek a mai napig megmaradtak.

A nyugati sakk néven is emlegetett ma ismert sakk arab kereskedők révén jutott Európába, ahonnan aztán elterjedt az egész világon. A végleges formáját a 16. század környékén nyerte el, de a figurák alakja csak 19. századra lettek egységesítve. Az első megnyitásokról szóló könyvet 1843-ban adták ki, az első nemzetközi sakktornyt pedig 1851-ben rendezték. A játék, illetve sport népszerűsége azóta is töretlen, az egyik legismertebb táblajáték világon, bár nem mindenki tudja játszani, nehéz olyan embert találni, aki nem ismeri. Az emberek elég nagy része még a bábuk mozgási szabályait is tudja, még akkor is, ha szinte sose játszott vele.



1. ábra Chaturanga és a modern sakk

2.2 Sakkgépek és programok fejlődése [4]

Az első sakkjátékra alkalmas gép az 1950-es években jelent meg ami még egy vákuumcsöves számítógép volt, ezeket a gépeket egy kezdő még könnyűszerrel elverte, de már képes volt a szabályoknak megfelelően lépni, a tábla állapotát figyelembe véve. Az informatika rohamos fejlődésével 40 évvel később már arra a szintre jutottak a sakkgépek, hogy 1996-ban egy Depp Blue nevű sakk számítógép képes volt elverni Garry Kasparov-ot az akkori világbajnokot hatból egyszer és kétszer ért el döntetlent. Egy évvel és négy hónappal később 1997 májusában pedig 3,5-2,5-re legyőzte a világbajnokot.

A Deep Blue egy kifejezetten sakkozásra tervezett szuperszámítógép volt, ami megközelítőleg akkora volt, mint egy két ajtós ruhás szekrény. Csak 2005-2006 körül kezdtek nyerni azok a sakkprogramok, amik nem célszámítógépen, hanem egy általános felhasználású személyi számítógépen futottak.

Ezt követően több ember – gép játéka is sor került, az utolsó komoly játszma Vladimir Kramnik és a Fritz nevű sakkprogram között került sor, a gép 4-2-re győzött, 4 döntetlennel és 2 győzelemmel a gép javára, Kramnik egy győzelmet sem tudott elérni. Ezen a ponton egyértelművé vált, hogy többé nem nyerhet az ember a gép ellen sakkban. Ahogy a játszma lebonyolításában segédkező professzor Monty Newborn is kijelentett a játszma után:

“I don't know what one could get out of it at this point. The science is done.”

(Nem tudom valaki mit tud ebből szerezni ezen a ponton. A tudomány kész.)



2. ábra IBM Deep Blue



3. ábra 90-es évekbeli nyomásérzékeny sakkgép

3. Fizikai sakkgépek

3.1 Sakkgépek megvalósításai [5]

Az 1990-es évek óta vannak olyan sakkgépek, amik különféle módokon érzékeli a lépéseket, majd valamilyen módon közlik a gép lépését a játékossal, hogy lépjen a gép helyett. Így a játékos a valós térben láthatja a táblát a gép elleni játék során és nem csak egy kijelzőn. Az érzékelés többféleképp is megvalósítható, működhet kapacitív, induktív vagy mágneses alapon, de akár nyomásérzékenyek is lehetnek az egyes mezők. A komolyabbakban az érzékelés oly módon van megoldva, hogy nem csak a pozíciót érzékeli, hanem azt is, hogy az adott mezőn melyik bábu áll. Egy egyszerűbb táblában ez nincs megvalósítva. Azt, hogy melyik mezőn melyik bábu áll azt csak a kiinduló helyzetből és az azt követő lépésekből, tudja, tehát ha a például a kezdésnél felcserélem a bástyákat a futókkal akkor a játék hátralévő részében a futó figurája fogja a bástya szerepét betölteni és fordítva. A visszajelzés a mezőknél lévő LED-ekkel, LCD kijelzővel, vagy E-ink kijelzővel szokott történni, de léteznek olyanok is, amik hangutasítással közlik a gép vagy online ellenfél lépését.

Számos variációban léteznek, vannak önálló gépek, amiket nem lehet semmihez csatlakoztatni, vannak olyanok, amiket lehet csatlakoztatni PC-hez és/vagy az internetre és azon keresztül játszani, rögzíteni vagy analizálni a játszmát, de ha nem szeretnénk akkor önálló egységként is működnek. Valamint azok, amik pusztán a szenzorokat tartalmazzák és azt valamilyen kommunikációs csatornán, általában USB-n keresztül csatlakoztatni kell egy PC-hez vagy erre a célra készített külön „számítógéphez”. Azokat is ide sorolják, amiken csak egy beviteli eszköz segítségével lehet megadni a lépést, majd egy kijelzőn látható a válaszlépés, bár ezek közelebb vannak egy sakkprogramhoz mint egy tényleges sakkgéphez, mivel nem érzékeli magát a lépést, erre pedig egy notebookra telepített sakkprogram is képes, ami ugyanúgy hordozható.

Azok a gépek, amik képesek rögzíteni a játszmát általában PC-hez csatlakoztatva képesek erre. Olyan, ami képes rögzíteni és azt visszajátszani ezt egy PC segítségével nélkül nem nagyon van, csak a legdrágább luxus kategóriába tartozó gépek tudják, ezek 3-4000 dollár között kaphatóak, tehát Forintban már milliós nagyságrend.

3.2 Ma kapható kiemelkedő sakkgépek [5]

A ma kapható legjobb sakkgép a *DGT Revelation II*. Ez a gép teljesen önállóan képes működni, képes bármelyik profi sakk program futtatására. A lépéseket pedig mind LED-es visszajelzéssel mind a beépített OLED kijelzőn képes megjeleníteni. Képes csatlakozni PC-hez vagy telefonhoz, tablethez, élőben közvetíteni a játékot vagy akár az interneten keresztül más játékosok ellen is van lehetőség játszani.

Egy másik kiemelkedő sakkgép szintén a DGT fejlesztése, a DGT Centaur. Ez egy másik irányból közelíti meg az ember-gép játékot. Az ember, mint fentebb tárgyaltuk már nem tud nyerni a gép ellen, ha azt teljes „erejéből” engedjük játszani. Viszont a világbajnokokat leszámítva nem sok értelme így játszani a gép ellen, tehát valahogy visszakell fogni a gépet. Ehhez általában be kell állítani a nehézségi szintet vagy ELO-ban megadva, vagy pedig szimplán a kezdő-haladó-mester lehetőségek közül választva. Ezzel szemben a Centaur gépi tanulással felméri az ellenfél erejét és nekünk azt kell megadni, hogy barátságos, kihívást jelentő, vagy expert szinten szeretnénk játszani. A barátságosnál a program lehetőséget ad a győzelemre „szándékosan hibázik”, a kihívást jelentőnél pedig az általa felmért tudásszintünkönél kicsivel magasabb nehézségi szintre áll be. Expert fokozaton pedig a Stockfish teljes ereje ellen játszhatunk, ami ma az egyik legerősebb sakkprogram a világon.

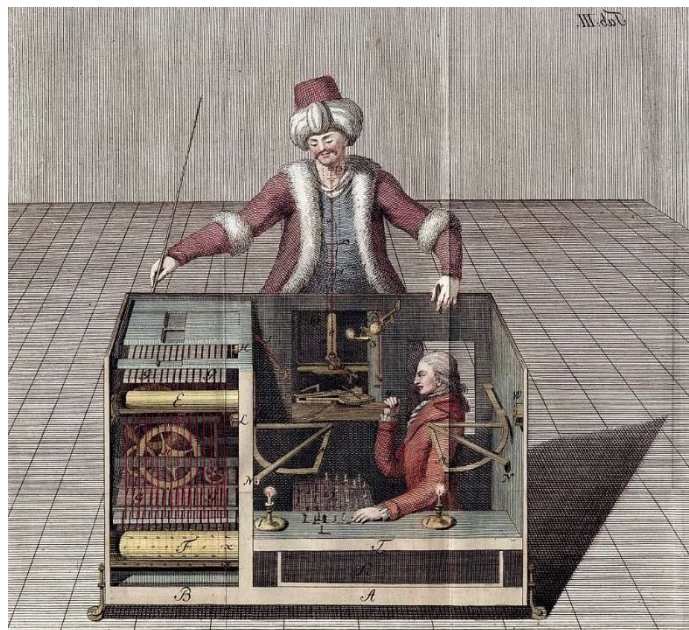


4. ábra DGT Revelation II

4. Az első „sakkgép” [6][7]

Bár valódi sakkgépekről csak az 1990-es évektől beszélünk volt egy korai szerkezet, amiről évtizedekig azt hitték, hogy képes sakkozni.

A szerkezetet egy magyar tudós Kempelen Farkas tervezte, akinek számos komoly mérnöki vívmányt köszönhetünk többek közt a pozsonyi hajóhidat, és a Száva és Adriai-tenger közötti csatornarendszert. A sakkozógépet Mária Terézia kérésére készítette 1769-ben. Több korabeli híres személyt legyőzött, mint Napóleont és Benjamin Franklint majd bejárta a világot miközben mindenki a csodájára járt. Kempelen azzal győzte meg a világot, hogy a játék előtt kinyitotta a gép elejét és hátulját majd egy gyertyát tartott a gép mögé, ezzel demonstrálva, hogy a gépen át lehet látni és nincs benne semmi csak mechanika. Természetesen ez csak egy optikai illúzió volt, a kor technikájával lehetetlen volt ténylegesen sakkgépet építeni, valójában egy profi sakkozó ült a gépben, és ellene játszottak. Volt egy saját sakktáblája a gép hasában és lépéseket mechanikusan tette meg a látható táblán egy Törököt ábrázoló bábuval. A megtévesztés része volt az is, hogy a szerkezet mellé kandelábereket állított így nem látszott, hogy a bábú feje füstöl, ugyanis a gépben ülő embernek látnia kellett, ehhez pedig akkoriban csak gyertya ált rendelkezésre, aminek a füstjét viszont valahol el kellett vezetni, hogy a sakkozó ne fulladjon meg a gép kis légterében. A leleplezésére csak 1804-ben került sor az alkotója halála után. Sajnos az eredeti 1854-ben egy múzeumi tűzben megsemmisült, egy rekonstruált változata viszont ma is megtekinthető a Műcsarnok kiállításában.



5. ábra Kempelen sakkozógépe

5. Tervezett megvalósítás

A sakktábla megvalósításánál elsősorban az egyetemi projektmunkának való megfelelést, majd szakdolgozattá való tovább fejleszthetőséget vettem figyelembe. A készülő okos sakktáblánál nem cél, hogy a prototípus vagy annak kismértékben módosított változata a piacon megállja a helyét, a kereskedelmi forgalomban lévő sakkgépek mellett. A felhasznált anyagok, eszközök és technológiák nem feltétlenül tükrözik a legoptimálisabb megoldást, sok esetben a számomra hozzáférhető és anyagilag elfogadható megoldások mellett döntöttem. Ezekben az esetekben feltüntettem a választás okát, illetve, hogy melyik megoldást találnám optimálisnak és miért. Törekedem arra, hogy a sakktábla logikája, működési elve és programja relatív könnyen átemelhető legyen esetleg egy tényleges termékbe a jövőben.

6. Modulok

Az okos sakktábla funkcióit jól elkülönülő modulok látják el. Így az egyes funkciók könnyen módosíthatók vagy kiválthatóak. Ezek az alábbiak:

- A sakkbábukat érzékelő szenzorok
- Sakkfigurák
- Egyedi sakktábla
- Visszajelző rendszer
- Vezérlő perifériák
- Sakkóra
- Hőnyomtató
- Mikrokontroller

6.1 A sakkbábukat érzékelő szenzorok

A sakkbábuk érzékelésére több megoldást is megfontoltam, az egyik fő szempont az volt, hogy az érzékelő kész egységként kapható legyen és a kimeneten könnyen feldolgozható logikai 1 és 0 értéket kapjak, hogy ne azzal menjen el a projektmunkára jutó idő, hogy az egyedileg készített érzékelőt kalibrálom.

Az első ötlet a mágneses érzékelés volt, ehhez pedig REED relékre gondoltam. A tesztelés során viszont nem bizonyult megfelelőnek, az érzékelés nem volt kellően megbízható, gyenge mágnesnél nem érzékelt kellő távolságból, erősebbnél mágnesnél, ahol már megfelelő magasságból érzékelt volna a bábukat a szomszédos bábuk is működésbe hozhatták. Ezenkívül annak a verzióknak, aminek az ára elfogadható lett volna túl gyengének bizonyult az üvegsőve és a lábak hajlításánál könnyen eltört. Ezeket figyelembe véve elvetettem a REED relét.

A következő lehetőségnek az unipoláris Hall szenzort választottam, a mágneses érzékelés vonalán maradva. A magasabb árkategóriás REED relé már elérte volna a Hall szenzor árát, ami egy sokkal megbízhatóbb és modernebb megoldás. A tesztelés során megfelelőnek bizonyult, ezért emellett döntöttem, a szenzor kimenete Open Collector-os, ami ideális számomra, könnyedén létre tudok hozni 0-5V-os logikát.

Felmerült a kapacitív vagy induktív érzékelés is, de a könnyen hozzáférhető szenzorok ipari alkalmazásra készültek, nem lettek volna alkalmasak számomra, mert túlzottan megnövelték volna a tábla vastagságát, amellet az áruk is igencsak meghaladja a Hall szenzorokét, ezért maradtam a mágneses érzékelésnél. A mágnesek által okozható kellemetlenségek miatt viszont egy piaci verzióhoz a kapacitív érzékelés lenne ideális, mivel a versenysakkbábuk nagyméretűek és tömör műanyagból készülnek, az induktív érzékeléshez pedig szükség lenne egy fémkorongot illeszteni a bábu aljához, ami megnövelné a bábuk súlyát, és egy további lépéssel bővítené a gyártási folyamatot.

6.2 Sakkfigurák

A sakkfigurák kiválasztásánál több problémába ütköztem, egy internetes sorozat népszerűsége miatt komoly készlethiány alakult ki a sakk készletekben. Számos boltban egyáltalán nem kapható, a többségben pedig igencsak szűk a választék, főleg az alacsonyabb árkategóriában.

Tömör műanyag készletet nem találtam, így egy fa készlettel kísérleteztem először. Mivel a Hall szenzorok úgy vannak kalibrálva, hogy csak a merőleges mágneses erővonalak képesek billenteni a kimenetet ezért a bábuk alját minél nagyobb részben kell a mágnesnek takarnia, és a mező méretet úgy kell megválasztani, hogy lehetőleg ne lehessen a bábút helyezni oly módon, hogy ne takarja mágnes a szenzor érzékelési felületét. Ezen felül a mágnesnek elég erősnek kell lennie ahhoz, hogy kellő távolságból is érzékelje a bábukat.

A teszteléshez vásárolt fa sakkbábuknál több probléma is felmerült, az érzékeléshez kellő erősségű mágnes túl erősnek bizonyult a bábuk súlyához képest, amennyiben akkorára csökkentettem a mező méretet, hogy a mágnes mindenképp takarja a szenzort akkor a szomszédos bábuk ellökték egymást. A probléma megoldására próbáltam különféle anyagokkal növelni a súrlódást, mint a bőr, szilikon, filc. A filc nem eredményezett előrelépést, a bőr és a szilikon meggátolta a bábuk eltolását viszont a bábúnak olyan kicsi volt a súlya, hogy ha túl közel raktam egy bábút a másikhoz akkor felborította. Ez nem lett volna elfogadható ezért alternatív megoldást kerestem. Kipróbáltam ferrit és neodímium mágneseket is különböző méretekben, a legjobbnak a 10x2 mm N35-ös neodímium korong bizonyult, de még ez is felborította a bábukat, ha nem elég óvatosan helyeztem őket egymás mellé, ezért más anyagú, de hasonló méretű sakkbábukra volt szükség.

A készlethiány miatt szűkek voltak a lehetőségeim, végül az üvegbábuk mellett döntöttem, az üvegbábuk súlya és mérete pont megfelelő. A készletben, amihez hozzájutottam a bábuk alja üreges volt feltehetőleg könnyítés céljából. Ez nekem kifejezetten jól jött így könnyedén beletudtam helyezni a mágneset. A mágnesek epoxi gyantával való rögzítése után felvettem a szükséges méreteket. 25x25mm-es mezőméretnél a bábuk se nem tolják se nem borítják fel egymást, valamint bár lehetséges úgy rakni a bábukat, hogy a mágnes ne takarja az érzékelési felületet, ha nem szándékosan úgy helyezi el a játékos akkor ez nem okoz problémát. Mellesleg ebben az esetben már kérdéses, hogy a bábú mely mezőn áll, ami amúgy sem tekinthető szabályos lépésnek. (15. ábra)

6.3 Egyedi sakktábla (17-18. ábra)

A tábla ötletét az összecukható fa sakktáblákból merítettem, bár azokkal szemben ez lényegesen nagyobb lett. A játék 8x8-as területe egy oldalon lesz és nem pedig a két oldalra szétosztva. A belsejében a játéktér oldalán rögzítettem a nyáklapot, amin a Hall szenzorok, a visszajelző LED-ek és a hozzájuk tartozó „Slave” áramköri elemek találhatóak. Ezen az oldalon került beépítésre a sakkóra gombjaiként használt kapacitív érintés szenzorok, a hétszegmenses kijelzők, a mátrix billentyűzet és a karakteres LCD kijelző.

Az alsó oldalon kapnak helyet a prototípusban felhasznált fejlesztői panelek és egy kiegészítő panel, valamint a sakkbábuk tárolására kialakított rész. A bábuk mivel üvegből vannak nem lehet őket ömlesztve tárolni, szállítani, ezért az aljába két fémlemez került rögzítésre. A felhasznált neodímium mágnesek elég erősek, ezért stabilan lehet vele rögzíteni a bábukat a fémlemezekhez. Ezen az oldalon található még a nyomtató, és más jövőbeni külső bővítés számra fenttartott tükörsor és a két Micro USB csatlakozó. Az egyik USB a felhasználói, amin keresztül lehet majd a korábbi játékokat lementeni, míg a másik a tápellátásért felel, illetve azon keresztül lehet a mikrokontrollert programozni, ha számítógépre van kötve.

Maga a tábla fából készült, a jelölések és a mezők plotterrel lettek kivágva egy nagy kopásállóságú matricából, ezzel szebb lett az eredmény mintha festve lett volna a játéktér. A LED-ek helyein papírfúróval készült lyukak találhatók, a hagyományos fa fúróval szemben vékony falemezt szebben, kevesebb roncsolással lehetett így elkészíteni.

A mérete elég robosztus, ez elsősorban abból adódik, hogy a játéktér alá helyezendő nagyméretű nyáklapot legyártatni aránytalanul drága lenne, ez természetesen nagyobb mennyiségnél már nem okozna problémát, de csupán egy példánynál ez nem volt elfogadható kiadás számomra. Ehelyett lehetőségem adódott, a szükséges nyáklap 1 rétegű, maratással történő elkészítésére, mivel viszont pusztán 1 rétegű lehet a nyák nem lehet olyan sűrűségben elhelyezni az alkatrészeket mint 2 vagy több réteg felhasználása esetén, ezért a szükséges terület nagyobb lett, aminek egyértelmű következménye, hogy a tábla is megnőtt az eredeti tervekhez képest.

Egy kereskedelmi forgalomba hozható változatban minden helyet kaphatna pusztán egy nyáklapon, természetesen a mikrokontroller is azon lenne és nem egy fejlesztői panelen, így nem lenne szükség a kinyitható táblára, a bábukat külön tárolóban szállítva csupán pár cm vastag lenne a kész tábla, és a játéktéren kívül csak a perifériáknak szükséges helyel lenne nagyobb a felülete.

6.4 Visszajelző rendszer

Többféle képpen is lehetőség lesz a visszajátszás, oktatás, valamint a gép elleni játék során a következő lépés jelzésére. A leglátványosabb és kézenfekvőbb mód magán a játéktéren, az adott mezőkhöz tartozó LED-el történő visszajelzés. Ezt egyszínű LED-ekkel fogom megvalósítani, felmerült az RGB LED használata is de az további bonyodalmak okozna mivel az egy rétegű nyáklapon további vezetősávokat kéne a LED-hez juttatni. Bár látványosabb lenne, ha a színekkel lehetne a szükséges információt átadni, fényjelzésekkel is megoldható a feladat. A piros, zöld és kék helyett egyszerűen gyors és lassú villogással, valamint folyamatos világítással történik majd a visszajelzés. A LED-es visszajelzés természetesen opcionális lesz, igény esetén kikapcsolható a menüben.

A LED-ek működtetéséhez demultiplexereket fogok használni, kihasználva a szem tehetetlenségét egymás után rövid időre bekapcsolgatom a szükséges LED-eket, kellő gyorsaságnál a játékos folyamatosan világítónak fogja látni az amúgy villódzó fényt.

Mivel másra nem tervezem felhasználni az LCD kijelzőt játszma közben ezért azon is leolvasható lesz a következő lépés algebrai módon megjelenítve, a LED-es visszajelzés mellett. Illetve normál játék során ezen lehet majd ellenőrizni, hogy a tábla megfelelően érzékelt-e a lépést és így szükség esetén orvosolni lehet a problémát.



6. ábra DGT Centaur aktív visszajelző LED-el

6.5 Vezérlő perifériák

A vezérléshez egy LCD kijelzőt és egy mátrix billentyűzetet fogok felhasználni. A beállításokat, információ megadást, játékindítást, valamint a játékmód kiválasztását mindent ezekkel a lehet majd megadni.

A kijelző egy 2x16-os kék/fehér LCD kijelző, amit 4 bites adatátviteli módban fogok felhasználni, mivel nem időkritikus a megjelenített információ ezért az R/~W lábat fixen földre kötöm így spórolva még egy lábat a mikrokontrolleren, így a busy flag-et nem tudom vizsgálni, de nem is fontos a jelenlegi felhasználásnál, hogy a maximális sebességén használjam az LCD kijelzőt.

Nincs szükség hosszú szövegek megjelenítésére ezért elegendő ez a relatív kisméretű karakteres kijelző. Egy nagyobb és nem karakteres kijelzővel meglehetne jeleníteni a tábla aktuális állapotát is, ami gyakorlásnál megkönnyítené a tábla állás felvételét, viszont ezt a funkciót a nyomtatónál szándékomban állt megvalósítani. A tábla állás felvétele a LED-es rendszerrel és ezzel az egyszerű karakteres kijelzővel talán még gyorsabb és egyszerűbb is lehet, elvégre pont az lenne a lényege egy sakkgépnek, hogy ne egy kijelzőn keljen nézni a táblát, hanem azt fizikálisan lássuk. Valamint egy komolyabb, nem karakteres kijelző lényegesen megnövelte volna a projekt költségét.

A billentyűzet 4x3-as lesz, ezzel lehet majd a menüben lépkedni, menüpontot kiválasztani értéket megadni. Ezzel lehet majd megadni a dátumot, az időt, a sötét és világos játékos nevét, a sakkórába beállítandó időt, ha nem a hagyományos játékmódokhoz előre definiáltak közül szeretnénk választani.

A könnyebb kezelhetőség végett egy ESP32 modul felhasználásával Bluetooth kapcsolaton keresztül, okostelefonnal is vezérelhető lesz a sakktábla. Így mindkét játékos a saját szemszögéből láthatja az információkat, 2 külön LCD kijelző hiányában az egyik játékos a kijelzett információt csak fejfel lefelé látná enélkül, valamint a nevek és egyéb szövegek bevitele is egyszerűbb a mobiltelefon érintőképernyőjével.

6.6 Sakkóra

A legtöbb sakkgépben van beépített sakkóra, ami teljesen logikus, hiszen rendelkezésre áll hozzá minden. Nem kötelező a használata, ha valaki a különálló sakkórát preferálja akkor egyszerűen kilehet kapcsolni, de még akkor is jól jöhet, ha elromlik, vagy csak elfelejti magával vinni.

A sakkórához én hozzáillesztettem a rendszerhez 2-2 hétszegmenses LED kijelzőt is, bár az LCD kijelző is alkalmas lenne erre a célra, de így szerintem könnyebb az idő leolvasása, és így játék közben az LCD-t fel tudom használni a lépések algebrai visszajelzésére. Az órát és a percet egy nagy 4 modulból álló kijelző fogja mutatni, míg a másodpercet egy kisebb 2 modulos kijelző. A nagy kijelzőn található az óra és a perc között egy kettőspont is, aminek félmásodperces villogtatásával tervezem szemléltetni, hogy melyik játékos órája megy. Erre azért lehet szükség mert a másodperc kijelzést opcionálissá teszem, ha valakit zavar ki lehet kapcsolni, vagy csak az utolsó 3-5 percben jelenik meg. Ezt a beállítást játékosonként külön is lehet majd állítani.

A sakkóra gombjához több megoldást is végig gondoltam, a mikrokapcsolós megoldás túl kicsi lett volna a kényelmes használathoz. A nyomógombos megoldások túl sok helyet igényeltek volna a tábla belsejében és szintén nem a legkényelmesebb megoldás.

Megoldás után keresve találtam, hogy kaphatóak előre összeszerelt, kalibrált kész áramkörök, amik kapacitív érintés szenzorként használhatóak. Az érintés idejére a Q kimeneten megjelenik a tápfeszültség, ami lehet olyan kicsi is, amit a használt mikrokontroller közvetlenül képes fogadni. Az áramkör nyáklapját csak belülről rá kell ragasztanom a tábla lapjára és kívülről csak meg kell jelölnöm az érzékelési területet. Az áramkör kellően érzékeny így nem kell pontosan ráhelyeznie a játékosnak az ujját, akár tenyérrel vagy több ujjal is működésbe hozhatja, ami kényelmes használatot biztosít és a végeredmény is esztétikus.

A gombok helyett használt áramkör felhasználható lett volna a bábuk érzékelésére is de ezen a ponton már túl sok pénz és idő lett a mágneses érzékelésbe fektetve, hogy előlről kezdjem. Tekintve, hogy a mágnesesek által okozott problémákat a tesztek alapján sikerült megoldani. Ezen kívül a gomb helyett használt kapacitív érzékelő ára a többszöröse, mint a Hall szenzoré és csak kis mennyiségben volt készleten.

6.7 Hőnyomtató

A nyomtató feladata, hogy vagy a játék közben folyamatosan, vagy pedig utólagosan kinyomtassa a rögzített játszmát. Tintasugaras vagy lézernyomtatóból nincs olyan kis méretű, amit könnyű lenne szállítani, ezért választottam a hőnyomtatót, amiből nagyon kis méretű is létezik. A felhasznált nyomtató kifejezetten mikrokontrollerekhez van tervezve, ezért könnyű az illesztése, és a programozása. A nyomtatót egyszerűen UART kommunikációval lehet vezérelni, alapvetően karakteres nyomtatásra készült, de képes grafika nyomtatására is.

A hőnyomtatásnak viszont van egy lényeges hátránya, márpedig, hogy a készült nyomatok nem időtállóak, és igényesek. Ideális arra, hogy gyorsan a helyszínen mindenkinek készüljön egy példány, akinek kell. Viszont ezt vagy le kell másolni egy időtállóbb technikával, vagy valamilyen módon rögzíteni, ha nem szeretnénk elveszíteni az információt. Ezért a nyomtatáshoz felhasznált adatok egy txt fájlban is rögzítésre kerülnek, amiket majd pendrive-ra is átlehet menteni, hasonló formátumban, mint amilyenben kinyomtatásra kerülnek.

A kinyomtatott dokumentáción minden adat megtalálható lesz, amit a hagyományos kézzel írott verzióban is fel kell tüntetni, mint a dátum, idő, nevek és a verseny adatai. Persze ha nem verseny rögzítésére készül a nyomtatott dokumentáció akkor csak a megadott adatok kerülnek nyomtatásra.

A lépéspárok sorszáma mellett lehetőség lesz az adott lépéshez felhasznált idő, vagy a sakkórában maradó idő kinyomtatására is. Utólagos nyomtatásnál lehetőség lesz lépésenként való haladásra is, ekkor az aktuális táblaállást vizuálisan is kilehet majd nyomtatni, ez segítség lehet az oktatásnál, így könnyebben fel lehet venni az adott állapotot hagyományos sakktáblán is, ha valaki gyakorolni akar.

Nyomtató hiányában a rögzített játszma később a memóriából is kinyomtatható lesz, ehhez a funkcióhoz mindenképpen meg kell majd adni a dátumot és az időt, hogy a tárolt játszmák között könnyebb legyen a keresés.

6.8 Mikrokontroller

Mikrokontroller választásnál elsődlegesen az döntött, hogy mi áll a rendelkezésemre. A globális chiphiány miatt nem lehet előre látni, hogy mikor kapnám meg a rendelést és már a projekt kezdetekor rendelkeztem a projekthez végül felhasznált fejlesztői panellel.

Egy sakkprogram jelentős számítási teljesítményt igényel, még optimalizált megoldás esetén is, a kezdeti verziókban viszont kifejezetten nagy sebességű mikrokontrollerre van szükség, hogy értékelhető eredményt adjon, ha korlátozott a lépés kiválasztására szánt idő, vagy belátható időn belül végezzen, az összes lehetőség megvizsgálása esetén.

A rendelkezésemre állt egy 8 bites és két 32 bites mikrokontrollerrel felszerelt fejlesztői panel, az előbb tárgyalt sebességigény miatt csak a 32 bites mikrokontrollerek jöhettek szóba. Az egyikre egy következő féléves kurzushoz szükségem lesz, ezért állandóan szét és össze kellett volna szerelnem, továbbá ez a gyengébb is a kettő közül. Az erősebb mikrokontroller egy Nucleo-144-es fejlesztői panelen van, egy STM32H755-ös. A H7-es széria az ST 32 bites ARM családjának a legerősebb tagja, így nem valószínű, hogy kevés lenne a teljesítménye a feladatra. A panellel más célom nincsen így végig az okos sakktáblához kapcsolva maradhat és ha nem oldódik fel a készlethiány akkor végleges megoldásként is alkalmazható.

Egy kész termékhez természetesen a fejlesztés befejezése után választanám ki a megfelelő mikrokontrollert, jelenleg még képlékeny a megvalósítás, a fejlesztés idejére ideális a fejlesztői panellel való munka. A fogyasztás nem szempont, így a választást elsősorban a szükséges teljesítmény, a lábak száma és egyéb funkciók határoznak meg. Mint például az USB és az Ethernet natív támogatottsága befolyásolná.

A H7-es mikrokontrollerben minden szükséges funkció megtalálható amennyiben feleslegesen erősnek bizonyul a végleges mikrokontrollernek lehet gyengébbet választani, vagy akár maradhat a fejlesztésnél felhasznált modell is, az árkülönbség nem olyan jelentős amennyiben nem lényegesen gyengébb modellre történne a váltás, a sakkprogram számítási igénye miatt viszont az nem lenne érdemes.

7. Program megvalósítása

7.1 Beállítások

Az indítás után lehet megadni majd az aktuális dátumot és időt. Ezt követően a program a főmenübe lép, ahol a beállítások menüpontban lehet majd átállítani az alapértelmezett beállításokat, mint a nyomtatási módot, a LED-ek funkcióit, sakkóra kikapcsolását és minden további működéssel kapcsolatos funkciót. Lehetőség lesz az alapértelmezett beállításokat módosítani ezeket a beépített Flash memóriába tárolom így a tápfeszültség megszűnése után is megmaradnak, majd az indításnál beolvasásra kerülnek.

A főmenüben lehet még játékmódot kiválasztani, vagy, hogy csak egy régebbi játszmát szeretnénk visszajátszani, nyomtatni, pendrive-ra menteni.

7.2 Ember-Ember játszma

Ebben a játékmódban elsődlegesen csak a megtett lépések rögzítése történik, a lépés szabályosságának ellenőrzése minden lépés után végbemegy, szabályos lépés esetén a memóriában tárolja a lépést, szabálytalan lépés esetén hibajelzést ad. Ha a játékos az LCD kijelző, vagy az okostelefon applikáció alapján hibásnak ítéli a tábla által érzékelt lépést akkor lehetősége van a '*' megnyomása után kijavítani azt, a bábu felemelésével majd újbóli, pontosabb lehelyezésével. A lépésjog átadása a sakkóra lenyomásával történik, ha az nincs kikapcsolva, egyébként automatikusan történik. A sakkórához a szokásos játékmódok ideje előre be lesz táplálva, de egyedileg is belehet majd időt állítani akár játékosonként külön-külön.

Amennyiben a nyomtató csatlakoztatva van a beállítás függvényében lépéspáronként nyomtathatja a lépéseket. Egy lépéspár akkor kerül véglegesítésre, amikor, a következő fehér lépés megtörtént. Ez alól kivételt képez a játék vége, matt vagy pedig feladás által, ebben az esetben az utolsó lépés automatikusan végleges lépésként kerül tárolásra és megjelennek a kijelzőn a játék utáni lehetőségek. Például teljes játszma nyomtatása, új játék, visszalépés a főmenübe, vagy a játszma adatai.

A beállítás függvényében a játékosoknak lehetősége van arra, hogy a felemelt bábú lehetséges lépéseit megjelenítse a LED-ek segítségével, vagy a gép tippet adjon a következő lépésre.

7.3 Visszajátszó mód

Ebben a játékmódban lépésről lépésre végig vezet az okos sakktábla egy korábbi játszmán. A lépéseket ugyanazon a módon közli a játékoskal, mint a gép elleni játék során. A visszajátszás bármikor megszakítható és onnantól folytatható, mint normál ember-ember elleni játszma vagy mint ember-gép elleni játék.

Kétféle megjelenítés lesz kivitelezve, az azonnali és a késleltetett. Azonnali módban a tábla a lépés megtétele után rögtön megjeleníti a következő lépést, késleltetett módban pedig csak valamelyik sakkóra gomb lenyomásával történik a következő lépés megjelenítése. Utóbbi az oktatás során lehet hasznos, egy tábla állás edzővel vagy egyedül történő kiértékelése megszokott gyakorlási mód, és zavaró lehet, ha a következő lépés már megvan jelenítve mikor a játékos próbálja kitalálni ő mit lépett volna az adott szituációban.

Lehetőség lesz arra, hogy ne kelljen mindig az elejétől végig vinni a játszmát, ha tudjuk, hogy hányadik lépéstől szeretnénk kezdeni az LCD kijelző és a LED-es rendszer segítségével adott utasítások alapján felvehető az adott tábla állás. Az LCD-ről leolvasható, hogy melyik mezőre milyen bábú kerüljön és az milyen színű legyen, a LED pedig az adott mezőn világítani fog. Ezt a folyamatot addig ismétli amíg minden bábú a helyére nem kerül.

A tábla adott állása vizuálisan is nyomtatható lesz. A visszajátszás során lehetőség lesz a lépések páronként nyomtatására is, pont, mint a játék során, de itt lehetőség lesz az algebrai leírásba beszúrni a vizuális táblaállás képét.

Gyakorlási célzattal lehet majd kérni a gépet, hogy mutasson egy jobb lehetőséget a megtett lépésnél. Más játékosok, vagy általunk korábban megtett lépések megjelenítésére is lesz lehetőség. Ezekre a gép szöveges visszajelzést ad, hogy jó, elfogadható, vagy rossz volt e.

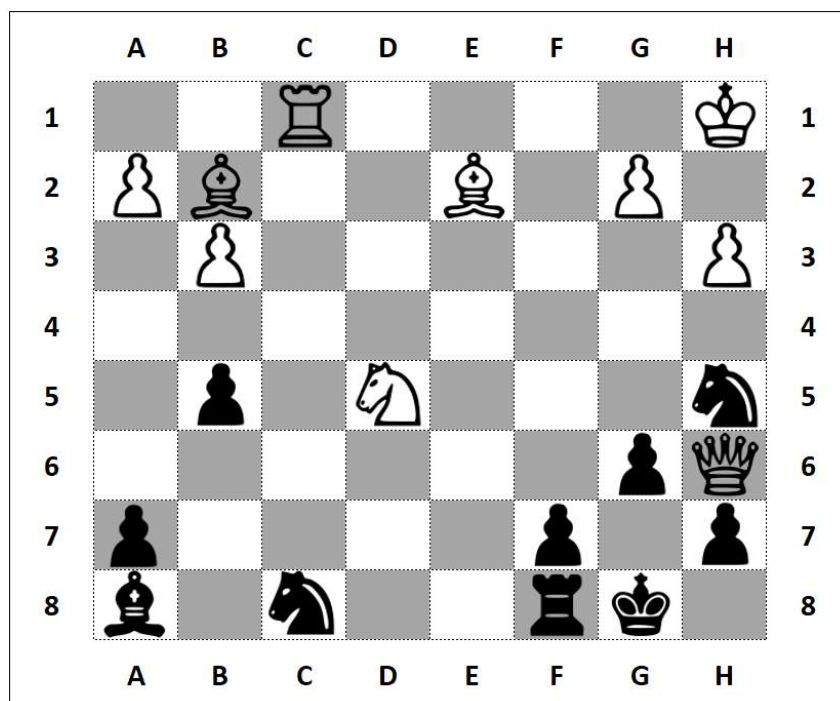
A projektben nem lesz túl látványos, kicsi a valószínűsége, hogy pont ugyanaz az állapot már fent állt korábban, és ugyanez igaz lesz a gép elleni játék során az emberi lépések beiktatásánál is a rendkívül sok lehetőség miatt.

7.4 Oktató mód

Az oktató módban előre beprogramozott edzéseket lehet végig csinálni. A játékosnak a megadott feltételek mellett végre kell hajtania az adott feladatot. Mint például adott állásból egy, kettő vagy három lépésen belül mattot kell adnia, vagy meg kell találnia melyik az a lépés, amivel anyagot tud nyerni, vagyis olyan helyzetet teremti, ahol az ellenfél mindenképpen elveszíti az egyik bábuját, mégpedig úgy, hogy értéküket tekintve megérje a csere vagy lehetőleg ne is veszítsen bábut.

Ehhez a visszajátszó módban ismertetett módon fel kell venni az adott tábla állást, majd megtenni a szükséges lépéseket. Az egylépéses megoldásokhoz egy előre megadott válaszlépést fog adni a tábla, míg a többlépéses megoldásoknál lehet választani, hogy vagy mindig ugyanazt csinálja az ellenfél, vagy a sakkprogram tegyen egy lépést. Viszont ezek a feladványok már nyert helyzeteket ábrázolnak, még a legkomolyabb sakkgép se tudna változtatni rajta, csak annyi értelme lehet a programot igénybe venni, hogy változatos legyen a reakció, illetve egy rossz lépés esetén hatékony válaszlépést adjon a gép.

A feladvány nyomtatható lesz, hogy később a táblától távol is lehessen gondolkozni rajta, és igény szerint a megoldást is a papír aláírára lehet majd nyomtatni.



7. ábra Világos 2 lépésen belül mattot ad (Vxf8 KxF8; BxC8#)

7.5 Ember-Gép játék [8]

A nehézségi szint kiválasztása után kezdődik a játék, itt is használható a sakkóra pont, mint a normál ember elleni játékmódban. Azt, hogy mit lépett a gép az LCD kijelzőről leolvashatjuk, illetve a LED-ek fényjelzések segítségével is mutatják majd.

A terv az, hogy a lépés kiválasztásánál nem csak a program által talált lépéseket veszem figyelembe, hanem az adott tábla álláshoz tárolom az ember által megtett lépéseket és összehasonlítom, hogy az mennyivel gyengébb, mint a gép által talált legjobb értékű lépés. Tekintve, hogy az emberi lépést is a gép fogja értékelni ezért a legjobb esetben is csak meg fog egyezni az érték a gép által talált legjobb lépéssel. A beállításoknál lesz lehetőség megadni, hogy hány százalékkal lehet gyengébb az emberi lépés a gép által választottnál ahhoz, hogy a program inkább azt válassza.

Mivel a gép csak adott mélységig számol előre, így akár lényegesen jobb is lehet egy lépés, mint amilyennek a program megállapította, nem feltétlen lesz gyengébb a gépi ellenfél, ha relatív nagy eltérés esetén is inkább az emberi lépés kap prioritást. Ezzel a módszerrel azt szeretném elérni, hogy kellően sok játszma után olyan lépéseket tegyen, amit korábban emberek is megtettek, így szimulálva egy emberi játékos hatását. Igény szerint ez a funkció kikapcsolható lesz a beállításokban, ha 0%-ra állítja a felhasználó a megengedett eltérést akkor mindenképpen a program által választott lépés kerül kiválasztásra, ellenben 100%-on függetlenül, hogy mennyire rossz egy tárolt lépés akkor is azt fogja választani, amennyiben van.

További haszna, hogy különböző megnyitások és azokra adott válasz reakciók tárolódnak az adatbázisba, így a játék elején amikor még nem nagyon vannak egyértelműen jó lépések az általános keresési módok szerint, a program ahelyett, hogy szimplán azt venné figyelembe, hogy melyik bábunak „hol jó” lennie, konkrét megnyitásokat tud végezni. Ezt a módszert a legtöbb sakkprogram szintén alkalmazza, csak egy előre létrehozott adatbázist használnak, és akkor vált vissza az általános keresésre amikor a megtett lépés már nem része egy megnyitásnak sem.

Ha a gép csupán egyetlen embertől „tanulna” nem lenne túl effektív, ezért mindenképp érdemes lenne ezt egy terméknél internetes adatbázisban tárolni. Érdemes figyelembe venni azt is mennyire gyakori egy lépés.

8. Sakkprogram [9][10]

A sakkprogram „engine” -je egy interneten talált verzióra alapoztam. Több lehetőség megvizsgálása után döntöttem emellett. A választásnál figyelembe vett fő szempontjaim, hogy ne legyen túl komplexen megvalósítva és elfogadható szinten lehessen ellene játszani.

Ha túl komplex lenne a választott program, mint például Stockfish akkor nehéz lenne, az elképzelésemet implementálni vagy túl erőforrás igényes lenne, hogy megfelelően működjön a felhasznált eszközökön, az átalakítása pedig időigényes lenne.

Azt is megfontoltam, hogy abszolút nulláról készítem el a saját verziómat, viszont a szakdolgozatig erre a célra rendelkezésemre álló időből túl sokat emésztett volna fel. Amíg a lépésgenerálás ellenőrzésére használt Perft teszt eredménye nem adja vissza pontosan ugyanazt az eredményt, ami előre megvan határozva, nem lehet tovább haladni, ezen a ponton a hibakeresés rendkívül sok időt vehet igénybe. Azáltal, hogy egy működő kódot vettem alapul, nem maradt ki egyetlen kritikus aspektus sem a lépésgenerálásból így lényegesen lerövidült a fejlesztési idő és a projekt más területeire is jut elég.

A választott verzió kifejezetten a sakkprogram készítés bemutatására készült, így bár nem a legoptimálisabb és legerősebb, viszont jól áttekinthető és könnyen módosítható.

8.1 Játék tér

A tömb alapú táblakezelés könnyebben áttekinthető és könnyebb vele dolgozni, a bitboard-os programok ezzel szemben gyorsabb működést eredményeznek, így ugyanazzal az erőforrással nagyobb mélységig lehetne vizsgálatot végezni. Mivel elsősorban nem a táblaállás tárolásának módja határozza meg a program hatékonyságát, ezért inkább egy könnyebben kezelhető tömb alapú verziót kerestem.

A tömb alapú játéktér egy 120 elemű tömb, minden eleme a tábla egy mezőjét reprezentálja az értéke pedig, hogy üres e vagy milyen bábu áll rajta. Egy nyugati sakktábla csupán 64 mezőből áll, a fentmaradó 56 biztonsági hálóként funkcionál és az értéke mutatja, hogy a sakktáblán kívüli mezőről van szó. Ezzel a módszerrel a lépésgenerálás egyszerűsödik, nem kell külön feltételekhez kötni a vizsgálatot a tábla szélén álló bábukhoz, amennyiben a generált lépés célmezőjének az értéke OFFBOARD (táblán kívüli) abban az esetben nem kerül tárolásra, mint lehetséges lépés.

	A	B	C	D	E	F	G	H			
	0	1	2	3	4	5	6	7	8	9	
	10	11	12	13	14	15	16	17	18	19	
1	20	21	22	23	24	25	26	27	28	29	1
2	30	31	32	33	34	35	36	37	38	39	2
3	40	41	42	43	44	45	46	47	48	49	3
4	50	51	52	53	54	55	56	57	58	59	4
5	60	61	62	63	64	65	66	67	68	69	5
6	70	71	72	73	74	75	76	77	78	79	6
7	80	81	82	83	84	85	86	87	88	89	7
8	90	91	92	93	94	95	96	97	98	99	8
	100	101	102	103	104	105	106	107	108	109	
	110	111	112	113	114	115	116	117	118	119	
	A	B	C	D	E	F	G	H			

	A	B	C	D	E	F	G	H	
1	0	1	2	3	4	5	6	7	1
2	8	9	10	11	12	13	14	15	2
3	16	17	18	19	20	21	22	23	3
4	24	25	26	27	28	29	30	31	4
5	32	33	34	35	36	37	38	39	5
6	40	41	42	43	44	45	46	47	6
7	48	49	50	51	52	53	54	55	7
8	56	57	58	59	60	61	62	63	8
	A	B	C	D	E	F	G	H	

8. ábra Játéktér

A fenti ábrán látható a tábla vizualizálása. A 120 eleműnél a zöld mező jelöli a táblán kívüli részt. A 64 eleműre többször szükség van az egyes mezőkhöz való adott érték hozzárendelésére, ezekben az esetekben felesleges egy 120-elemű tömböt használni, helyette csak egy 64 elemű tömb kerül tárolásra és az indexek között egy egyszerű makróval történik a megfeleltetés.

8.2 Lépés generálás

A gyalogok lépésének generálásához és sáncoláshoz feltételvizsgálatot használok, a gyalogokkal speciális lépés is lehetséges (kezdő pozícióból dupla lépés, „En passant” ütés,) viszont alapvetően a mozgásuk egyszerű és így könnyebben áttekinthető a kód, és nem is hosszabb számottevően. A fejlesztés során az „En Passant” szabályból és a Sáncolásból is kimaradt egy apró, alapvetően ritka feltétel, amit csak a Perft teszt egy speciális állapota hozott ki. A hiba felderítése relatív hamar megtörtént, a megvalósítás módja miatt.

A többi bábunak a lépéséhez egy tömböt használok, amiben le van tárolva, hogy melyik bábú hova, vagy milyen irányba tud elmozdulni, index alapon. Például a Huszár az 'L' alakú lépésével 8 mezőre tud lépni, ebbe beletartoznak a „nemlétező mezők” is. Ha a B1-es mezőn áll (23-as index) akkor a -8 és -19 is egy lehetséges 'L' alakú lépés, de lelépne vele a tábláról. Mivel úgyis meg kell vizsgálni, hogy mi van a cél mezőn, ezért a lépést ugyanúgy létrehozuk, és amennyiben a kapott érték OFFBOARD abban az esetben nem tároljuk, mint lehetséges lépés. Ha az ellenfél bábuja van rajta akkor mint lehetséges ütés kerül tárolásra, ha saját bábú akkor szintén elvetésre kerül a lépés, mint mikor táblán kívülre lépnénk, illetve, ha üres akkor pedig „csendes” lépésként kerül rögzítésre.

A „csúszó” bábuknál (Vezér, Bástya, Futó) hasonló elven történik a lépésgenerálás, az egyetlen különbség, hogy a tárolt értéket folyamatosan hozzáadjuk vagy kivonjuk az indexből addig amíg a célmező értéke „üres”. Minden egyes üres mezőre való lépés egy külön lehetőségként kerül tárolásra. Az irány vizsgálata véget ér, ha OFFBOARD, vagy valamilyen bábú értéke van a vizsgált mezőn, ezután jön a következő irány vizsgálata.

	A	B	C	D	E	F	G	H		
	0	1	2	3	4	5	6	7	8	9
	10	11	12	13	14	15	16	17	18	19
1	20	21	22	23	24	25	26	27	28	29
2	30	31	32	33	34	35	36	37	38	39
3	40	41	42	43	44	45	46	47	48	49
4	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69
6	70	71	72	73	74	75	76	77	78	79
7	80	81	82	83	84	85	86	87	88	89
8	90	91	92	93	94	95	96	97	98	99
	100	101	102	103	104	105	106	107	108	109
	110	111	112	113	114	115	116	117	118	119
	A	B	C	D	E	F	G	H		

9. ábra Világos Huszár lépésgenerálása

	A	B	C	D	E	F	G	H			
	0	1	2	3	4	5	6	7	8	9	
	10	11	12	13	14	15	16	17	18	19	
1	20	21	22	23	24	25	26	27	28	29	1
2	30	31	32	33	34	35	36	37	38	39	2
3	40	41	42	43	44	45	46	47	48	49	3
4	50	51	52	53	54	55	56	57	58	59	4
5	60	61	62	63	64	65	66	67	68	69	5
6	70	71	72	73	74	75	76	77	78	79	6
7	80	81	82	83	84	85	86	87	88	89	7
8	90	91	92	93	94	95	96	97	98	99	8
	100	101	102	103	104	105	106	107	108	109	
	110	111	112	113	114	115	116	117	118	119	
	A	B	C	D	E	F	G	H			

10. ábra Világos Vezér lépés generálása

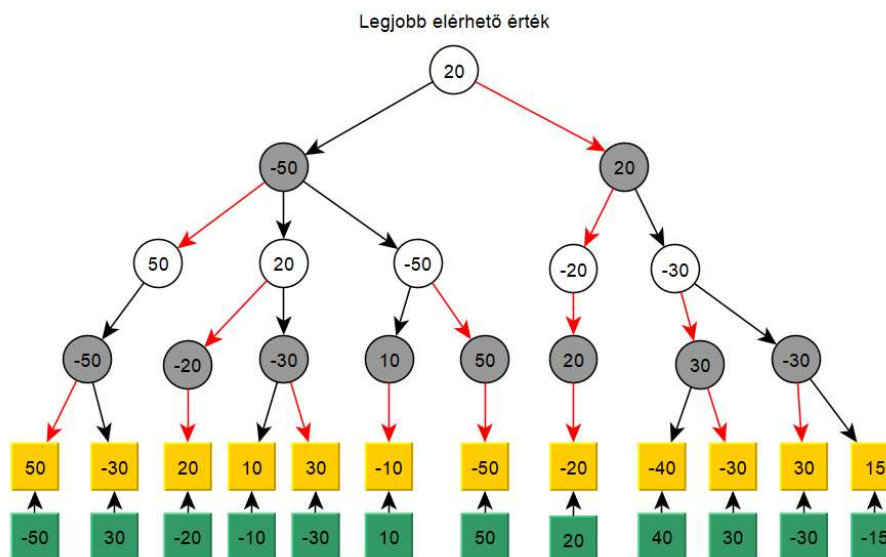
8.3 Lépés keresése

8.3.1 Minmax - Negamax

A lépés keresése a Minmax algoritmuson alapszik. A világos oldal a Maximum elérhető értéket keresi, míg a sötét a Minimum értéket. Ez két különálló függvényt lenne, viszont a bemenő paraméterek és a visszatérési érték negálásával összevonhatóak egy Negamax függvénnyé.

$$\max(a, b) == -\min(-a, -b)$$

Ezt kihasználva nem kell külön lekezelni mikor kell a maximum, illetve a minimum érték. A leafnode (keresési ág végén lévő tábla állapot) értékét mindig a vizsgálandó oldal szemszögéből adja vissza. Tehát ha sötét következik akkor a kiértékelés eredményét negálva vesszük figyelembe. A végeredmény, ha az adott oldal előnyben van/lesz pozitív lesz, ha hátrányban akkor negatív, de minidig a legnagyobb elérhető érték a legjobb lépés.



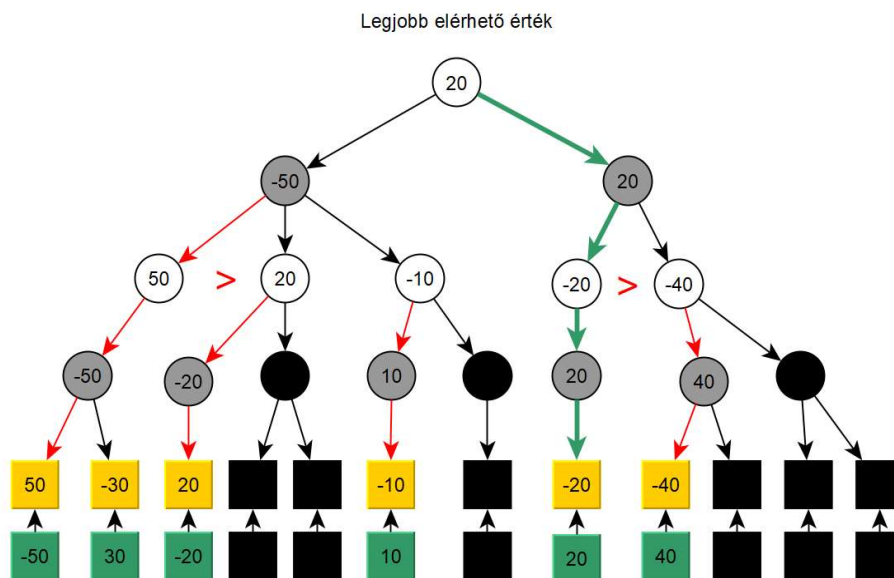
11. ábra Negamax vizsgálat 4-es mélységig (világos részére)

A fenti ábrán látható egy keresési fa. A zöld négyzetben a leafnode tényleges értéke található, mivel a kiértékelési résznél a sötét választ lépést ezért a kapott értéket átkell fordítani a sötét szemszögébe az értékek negálásával. Ezekből a negált értékekből választja ki a sötét a legnagyobbat (a saját szemszögéből a legjobb lehetőség), a választás eredményét negálva visszaadja a világosnak, így minden keresésnél a kereső oldalnak megfelelően tudja a függvény visszaadni, a legjobb lehetséges lépést.

8.3.2 Alfa-Béta

A következő lépés az Alfa-Beta bevezetése, bár a Negamax képes megtalálni a legjobb lépést, de ehhez minden lehetőséget átnéz a program. Bizonyos stádiumában a játéknak ez már ötös mélységnél is elérheti a milliárdos nagyságot, minden node érték kiszámítása és összehasonlítása ekkora számoknál már számottevő időt vesz igénybe, és ezeknek közel 90%-a felesleges. Az Alfa-Beta lényege, hogy a feleslegesen számolt ágak számát minimalizáljuk.

A 11-es ábrán látható, hogy melyik szituációban melyik oldal mit fog választani, milyen szituációba fogja a másik oldal „kényszeríteni” a játékost. Amennyiben a fehér sorban a balszélső 50-es érték megállapításra került, onnantól azon az ágon nem érdemes tovább keresni amennyiben az 50-es értéknél rosszabbra tud kényszeríteni az ellenfél.



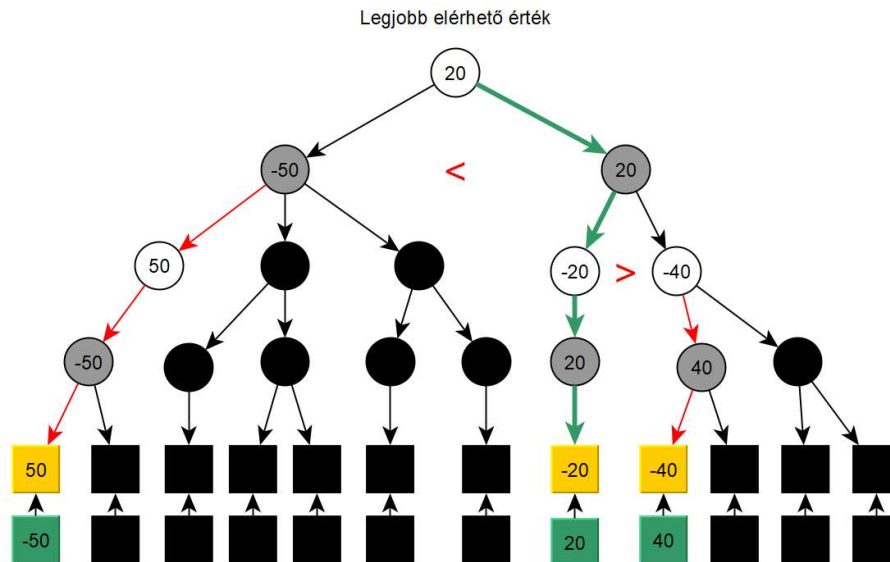
12. ábra Alfa-Béta keresés balról jobbra

A 12-es ábrán jól látható, hogy egy optimalizálatlan Alfa-Béta alkalmazásával is jelentősen csökkenthető az átvizsgálandó node-ok száma.

A módszer alapja, hogy az maximalizáló (alfa) és a minimalizáló (béta) számára is tároljuk az eddig talált legjobb értéket amíg az adott ágon vagyunk úgy, hogy a maximalizáló mínusz végtelenről és a minimalizáló plusz végtelenről indul. A Negamax miatt viszont itt is minden vizsgálatot az oldalnak megfelelő szemszögből vizsgálunk, az alfa és béta értéke minden mélységváltásnál negálva és felcserélve megy tovább. Amennyiben találunk nagyobb értéket, mint az eddigi legjobb, akkor alfa frissítésre kerül, viszont az első találatnál, ahol az érték nagyobb, mint béta (előző mélység negált alfája) az ág maradék részét eldobjuk. (Beta-cutoff)

8.3.2 Keresés sorrendje

Ha a keresést a potenciálisan értékeesebb node-okkal kezdjük akkor drasztikusan lecsökkenthető a vizsgálatok száma. Ha az eddig használt keresési fából a zölddel jelölt útvonalat járjuk be először, ami a példában a legjobb lehetőség, akkor a 13-as ábrán látható a vizsgálandó node-ok száma mennyire lecsökkent.



13. ábra Alfa-Béta Legjobb lehetőség elsőre kiválasztva

A lépések kiválasztásánál az alábbi sorrend követendő:

1. PV Move (principal variation) [1% javulás]
Adott táblaálláshoz előzőleg már megtalált és tárolt legjobb lépés
2. Ütés [98% javulás]
Legértékesebb áldozat, legértéktelenebb támadó sorrendjében
3. Killer Move [23% javulás]
Olyan tárolt csendes lépés, ami korábban már Beta-cutoff-ot eredményezett
4. Heurisztikus lépések [1% javulás]
Azok a lépések, ahol alfát frissíteni kellett.

A megvalósítás során egy adott állapothoz teszteltem hány százalékkal csökkent az átvizsgált node-ok száma a legjobb lépés megtalálásáig. Az ütésekkel kezdve, és a PV lépéseket utoljára implementálva.

Folyamatosan mélyülő keresésénél (Iterative Deepening) a PV tábla tudja a legjobban felgyorsítani, viszont ez rendkívül memóriaigényes megoldás abban a formában ahogy a mintakódban történt a megvalósítása. Ezzel a módszerrel bőven meghaladtam volna a mikrokontrolleremben rendelkezésre álló SRAM-ot. Így ezt a későbbiekben megpróbálom átalakítani és időalapon tesztelni, hogy megéri-e. Jelenleg korlátozott tárterülettel működik, de számottevő gyorsulást így nem nyújt.

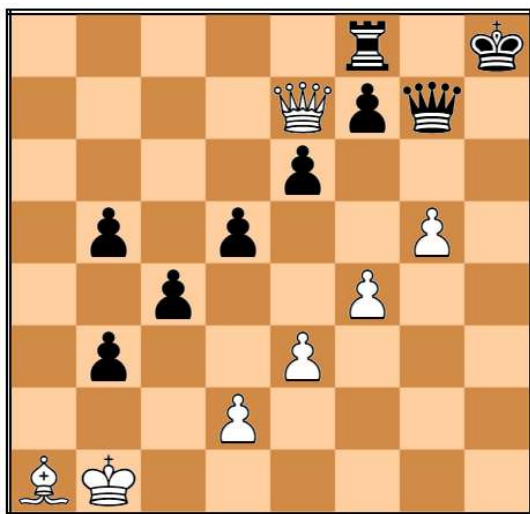
8.3.2 Horizont Hatás

A horizont hatás abból adódik, hogy csak korlátozott mélységig történik a keresés. A jelenség nagyon szembeütő, ha nincs lekezelve. Amennyiben a keresés azzal ér véget, hogy egy gyalogot lehet nyerni vezérrel való ütéssel és nincs jobb lépés akkor a gép azt fogja megtenni. A következő lépésként viszont a vezért egy másik gyalog leüti. Az oka, hogy a program nem lát messzebbre, mint a beállított mélység (Horizont), így nem tudja, hogy a következő lépésként az ellenfél visszatudja ütni a tisztünket.

Természetesen ez nagyobb mélységnél már nem ilyen látványos, viszont ilyenkor a program igyekszik a kritikus lépést elkerülni, ahogy halad előre a játék, mindig 1 lépéssel a kritikus lépés mögé látva „késlelteti az elkerülhetetlent”. Áldozatokat hoz, hogy megmentsen egy magasabb értékű tisztet, ami „menthetetlen”, ahelyett, hogy elkönyvelné a veszteséget és más irányban tervezne, így értékes anyagot és lépéseket pazarolva tovább gyengítve a pozícióját.

Ez a jelenség minden sakkprogramot érint, különféle módszerekkel a hatása csökkenthető, de nem eltüntethető. Az elsődleges módszer Quiescence keresések alkalmazása. Ami lényegében a fő keresés befejezésekor indít egy újabb korlátozott keresést, lényegesen nagyobb mélységben és a leafnode csak Quiescence végén kap értéket.

A felhasznált kódban a fő keresés végeztével tovább folytatódik a keresés maximum 64-es mélységig, illetve addig amíg létezik olyan lépés, ami ütést tartalmaz. Ha nem tudott több ütést generálni azt jelenti, hogy eljutottunk egy csendes lépésig és elvégzi a kiértékelést, vagy amennyiben valamilyen extrém eset folytán nincs csendes lépés 64 fél lépés után sem, akkor ott végzi el a kiértékelést. Tekintve, hogy 32 bábu lehet maximum a táblán erre rendkívül kicsi az esély.



A 11-es ábra a CPW oldaláról egy olyan szituációt mutat be, ahol a sötét oldal feláldoz 4 gyalogot, hogy „megmentsse” a vezért a horizont hatás miatt, még egy 8 mélységű vizsgálat során is, annak ellenére, hogy a vezér elveszett, viszont még cserélhető lenne egy Futóra.

(Pb2; Bxb2, Pc3; Bxc3, Pd4; Bxd4, Pe5; Bxe5)

14. ábra Horizont effekt szemléltetése

8.4 Sakkprogram felhasználása

Játékmódtól függetlenül a lépések tárolása, ellenőrzése, megtétele, kiértékelése a fentebb tárgyalt sakkprogramon belül fog működni. Csak bemeneti értékek létrehozására, illetve a kimeneti értékek feldolgozására/átalakítására van szükség.

Felesleges lenne külön kezelni többi játékmódot mivel a gépi játékhoz már minden megvalósításra került. Az egyetlen különbség a gépi játékhoz képest, hogy nem indul keresés automatikusan egyik oldal részére sem, illetve a keresés eredménye nem kerül rögzítésre, mint lépés csak kijelzésre kerül, ha valamelyik játékos tippet kér a géptől.

A lépés szabályosságának ellenőrzéséhez a lépés generálással létrehozom az összes lehetőséget, ha a lépés megtalálható a listában akkor megpróbálom megtenni, amennyiben hiba nélkül lefut szabályos volt a lépés, ellenben valamelyik feltételnek nem felelt meg.

A döntetlen és a matt vizsgálata pedig egy 1 mélységű keresés, amennyiben nem létezik szabályos lépés és a király nincs támadva akkor a visszatérési értéke 0 (döntetlen), ha a király támadva van akkor pedig 29000/-29000 (matt) annak függvényében melyik szempontból végzem a keresést.

9. Továbbfejlesztési lehetőségek

Az alap elképzeléshez több ötlet is adódott a feladat megfogalmazása során. Az elsőszámú fejlesztési lehetőség egy saját Android applikáció készítése. A másik két elképzelés egy kamerával történő képfeldolgozás és egy SCARA vagy Delta robot által való bábumozgatás.

A jelen állapotában csak egy letölthető terminálos applikáció fog vezérlési funkciókat ellátni, saját alkalmazás fejlesztésével több funkció is megvalósítható lenne például a játék dokumentációk telefonra mentése, visszajátszás lépésenként grafikus megjelenítéssel, lépés bázis szinkronizálása szerverrel, távoli játék internet használatával.

A kamerás rendszer segítségével elkerülhető, hogy a hibás lépésregisztrálást a játékosnak kelljen észlelnie. Amennyiben a kamera mást érzékel, mint amit a táblába épített szenzor abban az esetben automatikusan kérni fogja a lépés megismétlését. Valamint a „megérintett bábú” szabálya is betartatásra kerülhet, amely szerint, ha a játékos megérinti az egyik figuráját azzal kell lépnie, ha a szabályok szerint erre lehetősége van, illetve az ellenfél figuráját megérintve köteles azt leütni. Ez alól kivétel, ha az „igazítók” felszólalás után nyúl hozzá, ezt a mátrix billentyűzettel lehetne kiváltani hangfelismerő hiányában. A szabály másik része szerint pedig az, hogy elengedett bábú már nem mozgatható, amit szintén képes lenne figyelni a sakktábla a kamera segítségével. Ezen kívül a feladás is történhetne a hagyományos módon a perifériák használata nélkül, a király feldöntésével.

A Delta vagy SCARA robot felhasználásával a tábla állások felvételét, és minden egyéb lépést, amit az embernek kellett a gép által adott visszajelzések alapján megtennie már maga az okos sakktábla tenné meg. Ez a két robot fajta a legideálisabb a síkon való elmozdításra, a sakkbábukat ugyanis csak felemelni kell majd az X/Y tengelyek mentén való elmozdulás után lerakni. A SCARA robot lenne a legjobb választás mivel az készenléti állapotban kihúzódhat a tábla mellé így nem lenne zavaró, míg a delta robot folyamatosan a játéktér felett lenne a kialakításából adódóan, ami kifejezetten zavaró lenne.

A mobil applikáció elkészítése mindenképp tervben van, idő függvényében a képfeldolgozás is megvalósításra kerülhet még a Bsc. képzés alatt. Elenben a robot elkészítésére sem anyagi sem idő keret nem áll jelenleg rendelkezésre, ez a funkció csak későbbiekben kerülhet megvalósításra a szakdolgozat elkészítésig nem.

10. Következtetés [5]

A tervezett okos sakktáblához alkalmazott megoldásokat kezdetben csak a saját tapasztalataim és elképzeléseim alapján, állítottam össze. A kutatás során fedeztem fel, hogy az elképzeléseim nagyrésze már megvalósításra kerültek, főleg a magasabb árkategóriás sakkgépekben.

A végeredmény, mint fentebb is említettem nem lesz alkalmas kereskedelmi forgalomba hozásra. Egy módosított változata, aminél nem a projektmunkára való házilagós és olcsó elkészítés a fő tervezési szempont, viszont alkalmas lenne. Ebben az összeállításban nincsen jelenleg a piacon sakkgép, találhatóak ennél komolyabb képességekkel rendelkező gépek, és gyengébbek is. Megfelelő árással szerintem lenne rá kereslet.

Az elkészült sakkprogram még közel sem tökéletes, több keresés optimalizálási megoldás is létezik, ami nincs belefoglalva, ezek megvalósításával ugyanazon idő alatt mélyebb vizsgálatot lehetne végezni, illetve csak egy nagyon alapvető kiértékelési módszert használ. Egy kész termékben érdemes lenne egy profi sakkprogramot alkalmazni. A Stockfish, amit a DGT Centaur-ban is alkalmaznak ideális lenne. Ingyenes és nyílt forráskódú, ezért módosítható, hogy figyelembe vegye az emberi lépésekkel feltöltött adatbázist.

A sakkgépet kiegészítve a kamerával és a robottal mindenképp egyedi helyet foglalna el a piacon a sakkgépek között.

Irodalomjegyzék

[1] Bartos Ferenc, a Csongrád Megyei Sakkszövetség elnöke; Szabálytalan játszmajegyzések;

2020.03.05. <http://www.sakkolo.hu/aktualitasok/szabalytalan-jatszmajegyzesek.html>

[2] Wikipédia Sakk

2021.11.05. <https://hu.wikipedia.org/wiki/Sakk>

[3] Wikipédia Sógi

2021.11.05. <https://hu.wikipedia.org/wiki/Sógi>

[4] Wikipédia Computer chess (Számítógépes sakk)

2021.11.05. https://en.wikipedia.org/wiki/Computer_chess

[5] Chess-site.com The Best Electronic Chess Boards (A legjobb elektronikus sakktábla)

2021.11.05. <https://www.chess-site.com/chess-boards/electronic-chess-boards/>

[6] Wikipédia Kempelen Farkas

2021.11.05. https://hu.wikipedia.org/wiki/Kempelen_Farkas

[7] Kultúra.hu A titokzatos Kempelen és a sakkozógép története

2007.04.01. <https://kultura.hu/gyerek-titokzatos-kempelen/>

[8] freeCodeCamp A step-by-step guide to building a simple chess AI

(Lépésről-lépésre útmutató egyszerű sakk AI készítéséhez)

2017.03.30. <https://www.freecodecamp.org/news/simple-chess-ai-step-by-step-1d55a9266977/>

[9] Chess Programming Wiki (Sakkprogramozási Wiki)

2022.04.01 <https://www.chessprogramming.org/>

[10] Vice Chess Engine (Vice Sakkmotor)

2014.03.13 <https://github.com/peterwankman/vice>

Mellékletek



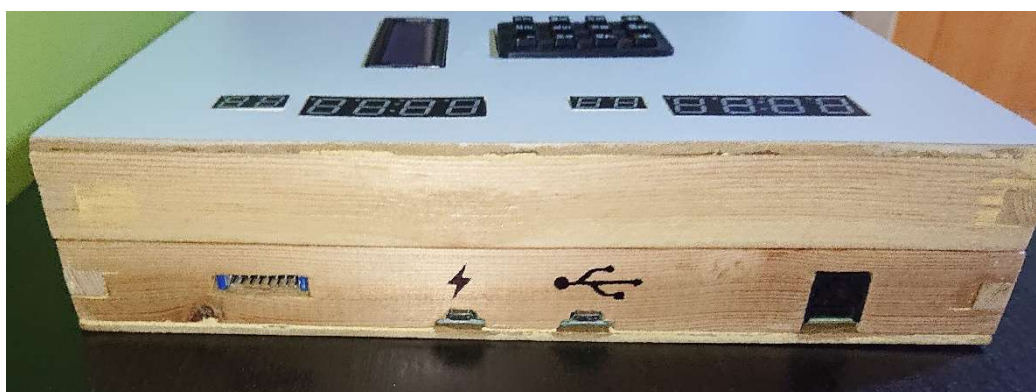
15. ábra Elkészült sakkbábuk



16. ábra Megvalósítás alapjául vett forma



17. ábra Elkészült sakktábla



18. ábra Sakktábla csatlakozói