

Ökopoly

Virtuális társasjáték

1-4fő részére

Készítette: Soltész Tamara, Popof Edina

Eseményvezérelt programozás

Szombathely 2019/2020 I. félév

Fejlesztői dokumentáció

A természet varázsát ontja bőven.
A fűben, a virágban és a kőben.
Ó nincs a földön oly silány anyag,
Mely így vagy úgy ne szolgálná javad;
De nincs oly jó, melyben ne volna vész,
Ha balga módra véle visszaélsz!

William Shakespeare

A projekt létrejöttének célja:

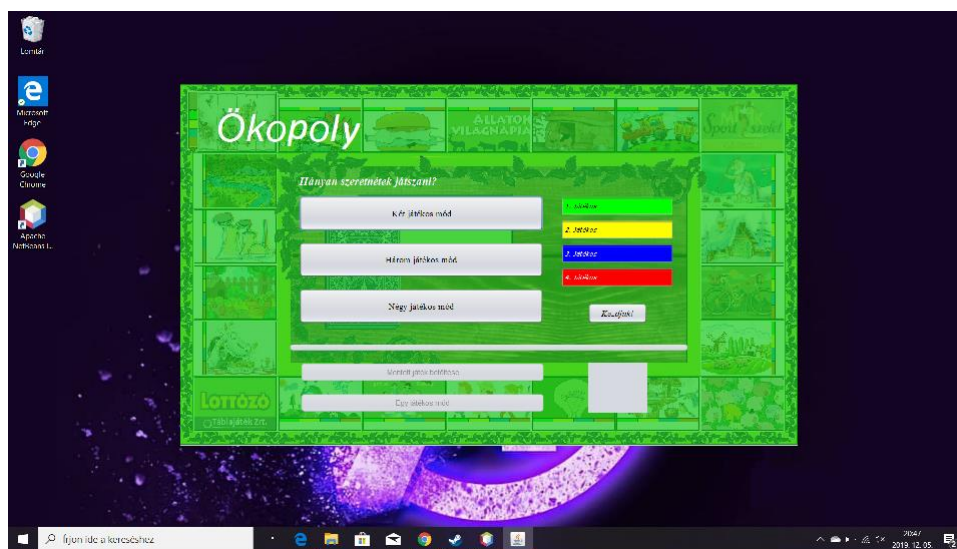
A 2019/2020-as tanév első félévének Eseményvezértelt programozás elnevezésű tantárgyának sikeres teljesítéséhez követelményként szerepel, egy darab nagy házi feladat elkészítése. Mivel a feladat elvégezhető csoport munkában is, így ez a házi 2 hallgató munkájával készült el. A projektben részt vett Soltész Tamara és Popof Edina. Első körben a téma megválasztása, az ötlet kigondolása volt feladva. A résztvevők az alábbiakban állapodtak meg:

A Projekt neve és célja:


A program kezdetben az Ökonómy projekcímet viselte, de becéztük Monopolynak is, bár a játék alapját a Magyarországon jól ismert táblás játék, a Gazdálkodj okosan adta. Ennek a játéknak a fő vonalait tartottuk meg, és színeztük ki olya elemekkel, amik a játékosokat a környezettudatos életvitel irányába próbálja terelni, és megmutatni nekik, hogy mik azok az apró kis figyelmességek, amik betartásával már tehetnek valamit környezetünk védelmében. Egyfajta "anti monopoly"-ként tekintünk rá, bár ambivalens módon, itt is szerepet játszik valamilyen mértékben a pénz, és a monetáris társadalom. Érdekesség, hogy –mint számunkra utólag kiderült - létezik már egy eredeti Anti-Monopoly társasjáték, amit a San Franciscoi Egyetem professzora Ralph Anspach alkotott meg 1973-ban. Természetesen, mivel akkor még nem voltak elterjedőben a számítógépek, ez a játék is papírtáblán jelent meg. Az általunk elképzelt játék - ami 100% -ban papírmintes formában jelenik meg - célja, hogy a játékosok által lelépett mezők és körök folyamán olyan alapvető életmódbeli tanácsokkal és tippekkel ismerkedjenek meg, amik egyéni szinten is hozzá járulnak a fenttartható, tudatos életvitelhez. A játékot az nyeri, aki a legmagasabb környezeti ponttal fejezi be a saját tábláját. Mi bízunk abban, hogy a program nem csak szórakoztató, hanem használata során a felhasználók sajátítottak el némi ismeretet a körültekintőbb életmódhoz, és tudatosabban fognak hozzáállni a rendelkezésre álló javak felhasználásához.

I. Program felépítése, fő grafikus felületek

1. Kezdő képernyő:



1. Ábra - képernyőkép: kezdő képernyő

A program  ikon segítségével indítható el, és az Ökopoly névvel rendelkezik. Kiterjesztése: .jar. A játék az indítást követően a fenti ablakkal köszönti a játékosokat. Csak ezen a felületen van lehetőség arra, hogy kiválasztásra kerüljön a résztvevők száma. Ez multiplayer játék esetén minimum 2 főt, maximum 4 főt jelent. Emelet lehetőség van egyjátékos módra is, ebben az esetben egy gépi játékoskal lehet játszani. Alapértelmezett kezdésként a négy játékos mód van kiválasztva. Minden nem gépi játék esetében a játékosokhoz előre rendelt szín tartozik. 4-es mód esetében a zöld, sárga, kék, piros színek használhatóak; 3-as mód esetén a zöld, sárga, kék; 2-es módnál pedig a zöld és a sárga. Lehetőség van a játékosok nevének megadására is, bár ez nem feltétele a játék elkezdésének. Kezdő képernyőről érhető el a mentett játék betöltése is. Ez csak abban az esetben működik, ha a mentést követően még nem játszottak le egy teljes játékot. Amennyiben az új tábla befejezésre kerül, a meglévő mentés törlődik. Csak egy darab mentési hely áll rendelkezésre. A program kezdő képernyőjéről az Esc gomb megnyomásával léphetünk ki.

Kezdő képernyőkép helye a kódban:

A játékot az Apache NetBeans IDE 11.2 programban készítettük el. A kezdő képernyő kódjának megírásához az alábbi metódusok felhasználására került sor:

| | | | |
|------------------------------|----------------------|-------------------------|----------------------|
| java.awt.Color | Grafikai megj. felel | java.awt.Dimension | Grafikai megj. felel |
| java.awt.Graphics | | java.awt.Graphics | |
| java.awt.image.BufferedImage | | java.awt.Toolkit | |
| javax.imageio.ImageIO | | javax.swing.JTextField | |
| java.awt.BorderLayout | Formázásért felel | java.io.File | File kezelés |
| javax.swing.Box | | java.io.FileInputStream | |
| javax.swing.BoxLayout | | java.io.FileReader | |

```

21 public class KezdoGUI extends javax.swing.JFrame {
22
23     Vector<String> nevok=new Vector<String>();
24     int jatekosokszam=4;
25     Dimension dim = Toolkit.getDefaultToolkit().getScreenSize(); // A képernyő nagyságának lekérdezése
26     int maxSzam=100; // A maximális lehetséges pontszám
27     int szamokSzam=(int) (dim.getHeight() * 0.8); // A szamok száma
28     Graphics g;
29     BufferedImage kocka;
30     BufferedImage kocka2;
31
32     public KezdoGUI() {
33         try {
34             kocka=ImageIO.read(new File("kocka01.png"));
35         } catch (Exception e) {
36             //
37         }
38         initComponents();
39         try {
40             BufferedImage kocka2=ImageIO.read(new File("kocka02.png"));
41             kocka2.setEnabled(true);
42         } catch (Exception e) {
43             //
44         }
45     }
46 }

```

1. kódrészlet-KezdoGUI.java

```

321 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
322
323     try {
324         //
325         //
326         //
327         //
328         //
329         //
330         //
331         //
332         //
333         //
334         //
335         //
336         //
337         //
338         //
339         //
340         //
341         //
342         //
343         //
344         //
345         //
346         //
347         //
348         //
349         //
350         //
351         //
352         //
353         //
354         //
355         //
356         //
357         //
358         //
359         //
360         //
361         //
362         //
363         //
364         //
365         //
366         //
367         //
368         //
369         //
370         //
371         //
372         //
373         //
374         //
375         //
376         //
377         //
378         //
379         //
380         //
381         //
382         //
383         //
384         //
385         //
386         //
387         //
388         //
389         //
390         //
391         //
392         //
393         //
394         //
395         //
396         //
397         //
398         //
399         //
400         //
401         //
402         //
403         //
404         //
405         //
406         //
407         //
408         //
409         //
410         //
411         //
412         //
413         //
414         //
415         //
416         //
417         //
418         //
419         //
420         //
421         //
422         //
423         //
424         //
425         //
426         //
427         //
428         //
429         //
430         //
431         //
432         //
433         //
434         //
435         //
436         //
437         //
438         //
439         //
440         //
441         //
442         //
443         //
444         //
445         //
446         //
447         //
448         //
449         //
450         //
451         //
452         //
453         //
454         //
455         //
456         //
457         //
458         //
459         //
460         //
461         //
462         //
463         //
464         //
465         //
466         //
467         //
468         //
469         //
470         //
471         //
472         //
473         //
474         //
475         //
476         //
477         //
478         //
479         //
480         //
481         //
482         //
483         //
484         //
485         //
486         //
487         //
488         //
489         //
490         //
491         //
492         //
493         //
494         //
495         //
496         //
497         //
498         //
499         //
500         //
501         //
502         //
503         //
504         //
505         //
506         //
507         //
508         //
509         //
510         //
511         //
512         //
513         //
514         //
515         //
516         //
517         //
518         //
519         //
520         //
521         //
522         //
523         //
524         //
525         //
526         //
527         //
528         //
529         //
530         //
531         //
532         //
533         //
534         //
535         //
536         //
537         //
538         //
539         //
540         //
541         //
542         //
543         //
544         //
545         //
546         //
547         //
548         //
549         //
550         //
551         //
552         //
553         //
554         //
555         //
556         //
557         //
558         //
559         //
560         //
561         //
562         //
563         //
564         //
565         //
566         //
567         //
568         //
569         //
570         //
571         //
572         //
573         //
574         //
575         //
576         //
577         //
578         //
579         //
580         //
581         //
582         //
583         //
584         //
585         //
586         //
587         //
588         //
589         //
590         //
591         //
592         //
593         //
594         //
595         //
596         //
597         //
598         //
599         //
600         //
601         //
602         //
603         //
604         //
605         //
606         //
607         //
608         //
609         //
610         //
611         //
612         //
613         //
614         //
615         //
616         //
617         //
618         //
619         //
620         //
621         //
622         //
623         //
624         //
625         //
626         //
627         //
628         //
629         //
630         //
631         //
632         //
633         //
634         //
635         //
636         //
637         //
638         //
639         //
640         //
641         //
642         //
643         //
644         //
645         //
646         //
647         //
648         //
649         //
650         //
651         //
652         //
653         //
654         //
655         //
656         //
657         //
658         //
659         //
660         //
661         //
662         //
663         //
664         //
665         //
666         //
667         //
668         //
669         //
670         //
671         //
672         //
673         //
674         //
675         //
676         //
677         //
678         //
679         //
680         //
681         //
682         //
683         //
684         //
685         //
686         //
687         //
688         //
689         //
690         //
691         //
692         //
693         //
694         //
695         //
696         //
697         //
698         //
699         //
700         //
701         //
702         //
703         //
704         //
705         //
706         //
707         //
708         //
709         //
710         //
711         //
712         //
713         //
714         //
715         //
716         //
717         //
718         //
719         //
720         //
721         //
722         //
723         //
724         //
725         //
726         //
727         //
728         //
729         //
730         //
731         //
732         //
733         //
734         //
735         //
736         //
737         //
738         //
739         //
740         //
741         //
742         //
743         //
744         //
745         //
746         //
747         //
748         //
749         //
750         //
751         //
752         //
753         //
754         //
755         //
756         //
757         //
758         //
759         //
760         //
761         //
762         //
763         //
764         //
765         //
766         //
767         //
768         //
769         //
770         //
771         //
772         //
773         //
774         //
775         //
776         //
777         //
778         //
779         //
780         //
781         //
782         //
783         //
784         //
785         //
786         //
787         //
788         //
789         //
790         //
791         //
792         //
793         //
794         //
795         //
796         //
797         //
798         //
799         //
800         //
801         //
802         //
803         //
804         //
805         //
806         //
807         //
808         //
809         //
810         //
811         //
812         //
813         //
814         //
815         //
816         //
817         //
818         //
819         //
820         //
821         //
822         //
823         //
824         //
825         //
826         //
827         //
828         //
829         //
830         //
831         //
832         //
833         //
834         //
835         //
836         //
837         //
838         //
839         //
840         //
841         //
842         //
843         //
844         //
845         //
846         //
847         //
848         //
849         //
850         //
851         //
852         //
853         //
854         //
855         //
856         //
857         //
858         //
859         //
860         //
861         //
862         //
863         //
864         //
865         //
866         //
867         //
868         //
869         //
870         //
871         //
872         //
873         //
874         //
875         //
876         //
877         //
878         //
879         //
880         //
881         //
882         //
883         //
884         //
885         //
886         //
887         //
888         //
889         //
890         //
891         //
892         //
893         //
894         //
895         //
896         //
897         //
898         //
899         //
900         //
901         //
902         //
903         //
904         //
905         //
906         //
907         //
908         //
909         //
910         //
911         //
912         //
913         //
914         //
915         //
916         //
917         //
918         //
919         //
920         //
921         //
922         //
923         //
924         //
925         //
926         //
927         //
928         //
929         //
930         //
931         //
932         //
933         //
934         //
935         //
936         //
937         //
938         //
939         //
940         //
941         //
942         //
943         //
944         //
945         //
946         //
947         //
948         //
949         //
950         //
951         //
952         //
953         //
954         //
955         //
956         //
957         //
958         //
959         //
960         //
961         //
962         //
963         //
964         //
965         //
966         //
967         //
968         //
969         //
970         //
971         //
972         //
973         //
974         //
975         //
976         //
977         //
978         //
979         //
980         //
981         //
982         //
983         //
984         //
985         //
986         //
987         //
988         //
989         //
990         //
991         //
992         //
993         //
994         //
995         //
996         //
997         //
998         //
999         //
1000        //

```

2. Kódrészlet-KezdoGUI.java

2. Játék tábla - játékos képernyő:

A játékosok számának kiválasztását követően elkezdhetjük a játékot. Ekkor egy új képernyőkép fog megnyílni. Ennek a neve a programban GUI.java. A felületen játszható ténylegesen maga a játék. A megjelenésében az alábbi grafikai elemek kaptak helyet.

- játéktábla: elkészítése Paint programmal történt, előre megszerkesztett kép;
- dobókocka: ennek megalkotásához egy Button itemet használtunk fel, a kocka értékét, azaz a pöttyöket véletlenszerűen jeleníti meg;
- játékos információk: JTree felhasználásával készült, itt szerepel lenyíló ágban a játékos neve, az aktuális pozíciója - ahol éppen tart a játék táblán-, bankszámlájának egyenlege és az általa megszerzett környezeti pontok száma;
- ház: az otthon, és az egyéb háztartási eszközök képe egy JPanel rétegen áll;
- információs mező: itt található, hogy éppen melyik játékos van soron, és hogy a mező amire érkezett, milyen tulajdonsággal rendelkezik. Kivitelezése JPanel segítségével történt.



3. Ábra - képernyőkép: Tábla képernyő

II. Játék menete

A játékosok a Start mezőről indulnak, az előre meghatározott sorrendben. A kezdő lapon kiválasztott játékos számtól függ, hogy a játékban milyen színek fognak játszani. A minimális 2 fős játék esetében a zöld és a sárga színek használhatók. Azt, hogy épp melyik játékos van soron, a tábla háttérszíne fogja mutatni, illetve a bal alsó sarokból induló információs ablak jelzi. Azt hogy ki melyik mezőn áll, az leolvasható a jobb oldalon lévő játékos információk panelről - ami a valóságban egy JTree - és a játéktábláról is. A megfelelő mezőn egy színes karika jelenik meg. Egy mezőn több játékos is állhat. Kezdetben a felhasználók 500.000 Ft-os egyenleggel, és 0%-os környezeti értékkel indulnak. A játék a dobókocka megnyomásával veszi kezdetét, és ezt használva haladunk tovább.

Minden körre jellemző, hogy a rendelkezésre álló egyenleghez hozzá íródik a havi fizetés, azaz 500.000 Ft. Az egyenlegen természetesen levonások is jóváíródnak, a költségek megfelelően. A felhasználó mindig csak annyi pénzt költhet el, amennyi a rendelkezésére áll, mínuszba nem mehet. Vásárolni kizárólag a felső sávban elhelyezett menüben található Boltban lehet, és ennek a műveletnek meg kell előznie a játékos dobását. A felszerelések különböző értékekkel érhetők el, és itt megtalálható összes tárgyból be kell szerezni minimum egy félért, hogy a játékot meg lehessen nyerni. Mivel a játékot az nyeri, aki több környezetvédelmi pontot gyűjt, érdemes azt a tárgyat megvásárolni, ami több pontot ad, igaz ezek beszerzési ára is magasabb. Felszerelést máshol vásárolni nem lehet, a nyerhető eszközök köre pedig korlátozva van a bútorokra.

A táblán előforduló mezőfajták:

A játékban összesen 22 mező szerepel a Start/Köreleje mezővel együtt. Ezek között különböző szerepkört betöltő mezők vannak:

- Start/kör eleje mező: a kezdő mező a játékban a hónap elejét szimbolizálja. Minden kör elején itt kapja meg a havi bérédet, egyéb grafikus felület nem tartozik hozzá.
- Közönséges mező(): ezeken a mezőkön nem történik semmi, átlagos cselekvést ír le a JPanel(1)-en tartózkodó JLabel(2-4) felületekre, a mező képi és szöveg világa része a Paint-ban megszerkesztett táblának .
- Tanács mező(): az ilyen típusú mezőkön a környezetvédelem egy-egy részterületének felmerülő problémájáról kap tájékoztatást egy felugró új ablakban. Az ehhez tartozó programkód, és az osztály megtalálható a Tanacs.java-ban. Grafikusan a JPanel1-re felfekvő képre olvasunk be egy random szöveget a JTextArea-ba. A feliratot a tanács mezőtől függően tölti be a megfelelő .txt állományból.
- Cselekvés mező: ezek a mezők a játék folyamatához adnak hozzá(3,9,16): a 3. mező kapott egy külön felugró ablakot, és egy osztályt, ami az UjAllasGUI.java-ban található meg. A mező funkciója az, hogy felkínál egy lehetőséget a játékos számára. Amennyiben az elfogadásra kerül, az adott játékos elveszti a játékot. Abban ez esetben, ha a gépi játékról, vagy kétszemélyes játékról volt szó, akkor ez automatikus befejezést jelent, és a játék mentés nélkül kilép. 9. mező: a játékos hozzájut egy kisebb

összeghez jutalom gyanánt.16. mező: Triton király elégedett hogy kitakarították az óceánt, és a MÉH is boldog, hogy vittek vasat. Jutalom hozzá íródik az egyenleghez;

- Szerencse mező(6,12,17): 3 különböző típusa elérhető a táblán, mindegyikhez saját külön osztály tartozik. Közös rájuk hogy minden mező tartalmaz magában minimum egy minijátékot.

Lottózó(6): Egy mező ára 250 Ft-t ami levonásra kerül az egyenlegéből. A 7 szám megjelölése után Ön esélyes lesz arra, hogy milliókat nyerjen. Legalábbis ez áll a reklámszövegben. A minijáték automatikusan felugrik egy új ablakban, amint a játékos az adott mezőre érkezik. Grafikaiag JPanel-ek és JLabel-ek segítségével lett megrajzolva. Háttérként egy erre a célra létrehozott .jpeg fekszik fel az ablak Jpalel(37)-jére. Abban az esetben, ha a játékos az adott számhoz tartozó JLabel(1-35)-re kattint a réteg háttérszíne pirosra vált, ezzel jelképezve, hogy az aktuális szám kiválasztásra került. Ezalatt a számot feltölti egy, a számunkra előre létrehozott - de üres - 7 elemű tömbjébe.

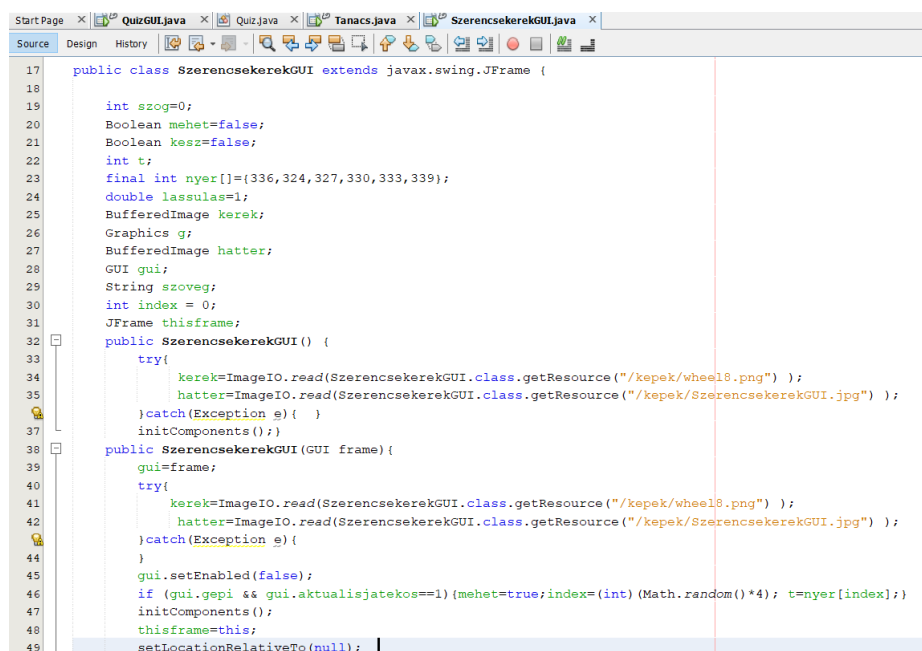
A lottózó matematikája: Első körben nem a véletlen szám generálás fut le, hanem azt döntjük el, hogy akarjuk-e hogy a játékos nyerjen. Ha nem, akkor - számok létrehozása nélkül - kiírjuk, hogy a játékos nem nyert. Abban az esetben, ha azt akarjuk, hogy érjen el nyereményt: az ő által feladott számok közül véletlenszerűen kiválasztunk 3 darab számot, és ezekhez hozunk létre további 4 számot. Természetesen a tartományon belül, a már felhasznált számok kizárásával. Így juthat hozzá a játékos különböző nyereményekhez, amik aztán jóváíródnak az egyenlegében. A lottózás kimenetele minden esetben megjelenik a tábla képernyő információs mezőjében.

```
Source Design History
1458 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
1459     feladott=new int[7];
1460     int idb=0;
1461     int eddigi=0;
1462     for (int i=0;i<szamok.length;i++){
1463         if(szamok[i]){
1464             feladott[idb]=i+1;
1465             idb++;}}
1466     this.setVisible(false);
1467     int nyere=(int) (Math.random()*10);
1468     if(nyere%3==0){ //Nem nyert semmit
1469         framegui.setTextJLabel("de nem nyert semmit");
1470     }else{
1471         int egyik=(int) (Math.random()*7); // A felhasználó által kiválasztott egyik szám
1472         int masik=(int) (Math.random()*7);
1473         while(egyik==masik){
1474             masik=(int) (Math.random()*7);
1475             int harmadik=(int) (Math.random()*7);
1476             while(egyik==harmadik || masik==harmadik){
1477                 harmadik=(int) (Math.random()*7);
1478             }
1479             int lottoszamok[]=new int[feladott.length];
1480             lottoszamok[0]=feladott[egyik];
1481             lottoszamok[1]=feladott[masik];
1482             lottoszamok[2]=feladott[harmadik];
1483             eddigi=3;
1484             for (int i=3;i<feladott.length;i++){
1485                 lottoszamok[i]=(int) (Math.random()*35+1);
1486                 for(int j=0;j<eddigi;j++){
1487                     while(lottoszamok[i]==lottoszamok[j]){
1488                         lottoszamok[i]=(int) (Math.random()*35+1);
1489                         j=0;}}
1490             }
1491             int seged;
1492             for(int i=0;i<lottoszamok.length-1;i++){
```

3. Kódrészlet- LottozoGUI.java

Szerencsekerék(12): képszerkesztő program segítségével van előre elkészítve egy szerencsekerék, amin különböző feliratok szerepelnek. Ezek a következők: Elektromos autó, +100 ökopont(kp), Háztartási eszközök, 1.000.000 Ft, Quiz, Jótanács.

- Elektromos autó: a boltban a játékos eszköztárában az autók közül az elektromos autó megvételre kerül, abban az esetben, ha a játékos rendelkezik gépjárművel, a nyeremény értéke íródik jóvá.
- 100 Ökopont : A játékos kp értéke növekszik 100 ponttal.
- Háztartási eszközök: a boltban a játékos eszköztárában megvételre kerül a háztartási eszközök egységcsomag. Amennyiben az játékos már rendelkezik a nyereménnyel, annak értéke íródik jóvá a játékos egyenlegén.
- 1.000.000 ft,-: az adott játékos egyenlege gyarapszik ezzel az összeggel.
- Quiz: Minijáték. Csak a Szerencsekerék-ből érhető el. Rendelkezik külön osztállyal. Működéséért a Quiz.java és a QuizGui.java felel. A QuizGUI a grafikus környezetet biztosítja, elkészítésekor a JPanel-en JTextArea-t és JButton(1-4) elemeket alkalmaztunk. A háttérként szolgáló képet úgy olvastattuk be a JPanel-re. A játék során véletlenszerű kérdéseket tesz fel a program, amikre a válasz lehet 2 vagy 4 kimenetelű is. Abban az esetben, ha a kimenetek száma 2, akkor csak ennek megfelelő számú nyomógomb kerül megjelenítésre. A játékos választásának következményeként sárga háttérszínezést kap, majd ezután késleltetve válaszreakció is érkezik. A válasz, ha jó zöld hátteret kap. Amennyiben nem talált, piros színt vesz fel a háttere, a jó válasz környezete pedig zöldre vált. A helyes válasz +5 ökopont jóváírását eredményezi.
- Jótanács: ismételten felhasználásra kerül a Tanacs.java osztály, a hozzá tartozó ablakával együtt. Erről a felületről csak akkor lehet tovább menni, ha a területén belül MouseClickedEvent hajtottódott végre.



```

17 public class SzerencsekerékGUI extends javax.swing.JFrame {
18
19     int szog=0;
20     Boolean mehet=false;
21     Boolean kesz=false;
22     int t;
23     final int nyer[]={336,324,327,330,333,339};
24     double lassulas=1;
25     BufferedImage kerek;
26     Graphics g;
27     BufferedImage hatter;
28     GUI gui;
29     String szoveg;
30     int index = 0;
31     JFrame thisframe;
32     public SzerencsekerékGUI() {
33         try{
34             kerek=ImageIO.read(SzerencsekerékGUI.class.getResource("/kepek/wheel8.png"));
35             hatter=ImageIO.read(SzerencsekerékGUI.class.getResource("/kepek/SzerencsekerékGUI.jpg"));
36         }catch(Exception e){ }
37         initComponents();
38     }
39     public SzerencsekerékGUI(GUI frame){
40         gui=frame;
41         try{
42             kerek=ImageIO.read(SzerencsekerékGUI.class.getResource("/kepek/wheel8.png"));
43             hatter=ImageIO.read(SzerencsekerékGUI.class.getResource("/kepek/SzerencsekerékGUI.jpg"));
44         }catch(Exception e){ }
45         gui.setEnabled(false);
46         if (gui.gepi && gui.aktualisjatekos==1){mehet=true;index=(int) (Math.random()*4); t=nyer[index];}
47         initComponents();
48         thisframe=this;
49         setLocationRelativeTo(null);

```

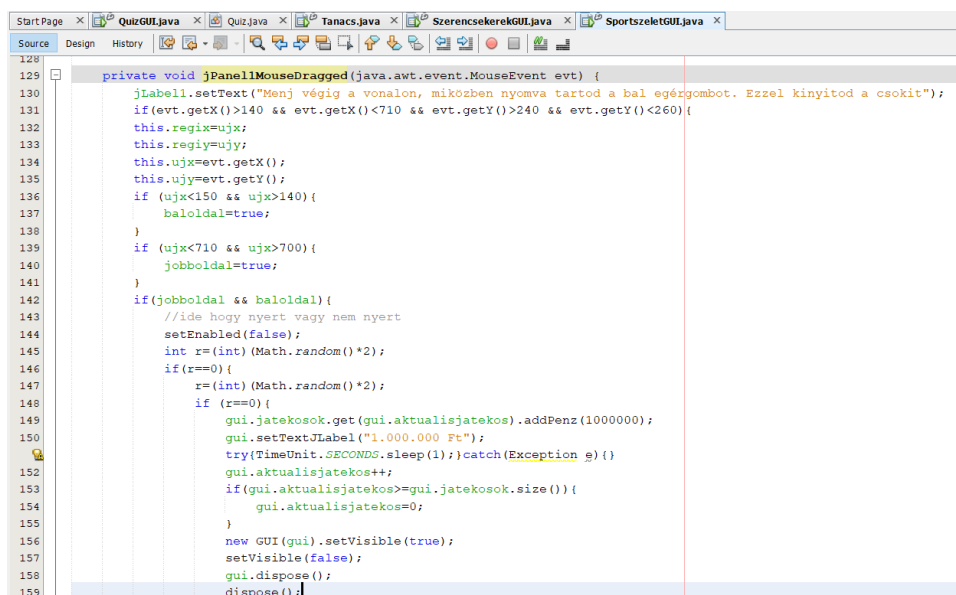
4. Kódrészlet - SzerencsekerékGUI.java

Miután lezajlott a megfelelő interakció, a játék visszalép a tábla képernyőre, az esetleges minijátékok pedig bezárásra kerülnek.

Nyerő csoki(17): Ehhez a mezőhöz is készült külön osztály a hozzá tartozó felugró ablakkal együtt - SportszeletGUI.java. Ez is egy minijáték. Lényege, hogy a csoki közepén vízszintesen végig húzott vonal felett, folyamatos egérnyomás kíséretében vezessék végig az egeret(MouseDragged a JPanel-en). Megengedett eltérés van, 5 pixelszélességben a vonaltól. Fontos kritérium még, hogy a vonal rajzolása az egyik végponttól a másik végpontig tartson, és folyamatos legyen. A minijáték lehetséges végkimenetelei: Amennyiben játékosnak nem sikerül végig a vonalon végig-vezetni az egeret, vagy nem az elejétől a végéig haladt : a csoki nyitás újra kezdődik. A játékos sikeresen kinyitotta:

Nyert még egy csokit. / Sajnos nem nyert semmit. / Pénznyeremény.

Ezekben az esetekben az információs sáv közli a megfelelő kimenetelt a tábla lapon.



```
128 private void jPanel1MouseDragged(java.awt.event.MouseEvent evt) {
129     JLabel1.setText("Menj végig a vonalon, miközben nyomva tartod a bal egérgombot. Ezzel kinyitod a csokit");
130     if(evt.getX()>140 && evt.getX()<710 && evt.getY()>240 && evt.getY()<260){
131         this.regix=ujx;
132         this.regiy=ujy;
133         this.ujx=evt.getX();
134         this.ujy=evt.getY();
135         if (ujx<150 && uyx>140){
136             baloldal=true;
137         }
138         if (ujx<710 && uyx>700){
139             jobboldal=true;
140         }
141         if(jobboldal && baloldal){
142             //ide hogy nyert vagy nem nyert
143             setEnabled(false);
144             int r=(int) (Math.random()*2);
145             if(r==0){
146                 r=(int) (Math.random()*2);
147                 if (r==0){
148                     gui.jatekosok.get(gui.aktualisjatekos).addPenz(1000000);
149                     gui.setTextJLabel("1.000.000 Ft");
150                     try(TimeUnit.SECONDS.sleep(1);){catch(Exception e){}
151                     gui.aktualisjatekos++;
152                     if(gui.aktualisjatekos>gui.jatekosok.size()){
153                         gui.aktualisjatekos=0;
154                     }
155                     new GUI(gui).setVisible(true);
156                     setVisible(false);
157                     gui.dispose();
158                     dispose();
159                 }
```

5. kódrészlet - SportszeletGUI.java

III. GUI.java-hoz tartozó egyéb osztályok szerepe és működése

1. Sugo.java:

A sűgó elérhető a játékos tábla menüjében. Szerepe, hogy információt adjon a játékosnak arról, hogy hogyan működik a játék, melyik mező mit takar. A sűgóhoz egy .txt file tartozik ezt olvassuk be.

```
(BufferedReader br=new BufferedReader(new InputStreamReader  
(SugoGUI.class.getResourceAsStream("/feliratok/Sugo.txt"), "UTF8")) ;
```

Grafikus megjelenítéséről a a JPanel-re illesztett kép gondoskodik.

2. *MentesGUI.java*:

Ez az osztály több eseményhez is tartozik. Őt hívja meg a program, ha a felhasználó(k) már a táblán van(nak), és a menüben akarnak menteni, új játékot létrehozni, vagy Esc-el kilépni a játékból/X-el kilépni. Míg a sima mentésre való klikkelés csak azt eredményezi, hogy a játék elmentésre kerül egy erre a célra létrehozott .txt dokumentumban, de maga az osztályhoz tartozó felület nem nyílik meg, addig az új játék, Esc, X esetében meghívódik a *MentesGUI.java* grafikus része is. Itt 3 opcióra van lehetősége a felhasználóknak: visszalép/kilép mentés nélkül/ment és kilép. Mentés esetén a már részletezett szabályok érvényesek.

```
173 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
174     gui.setEnabled(true);
175     dispose();
176 }
177
178 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
179     if (!uj){
180         System.exit(0);
181     }
182     else{
183         new KezdoGUI().setVisible(true);
184         gui.dispose();
185         dispose();
186     }
187 }
188
189 private void jPanel1KeyReleased(java.awt.event.KeyEvent evt) {
190     if (evt.getKeyCode()==27){
191         gui.setEnabled(true);
192         dispose();
193     }
194 }
195
196 private void formKeyReleased(java.awt.event.KeyEvent evt) {
197     if (evt.getKeyCode()==27){
198         gui.setEnabled(true);
199     }
200 }
```

6. kódrészlet - *MentesGUI.java*

3. *BoltGUI.java*:

Menüből érhető el. Itt szerezhetők be a játék teljesítéséhez szükséges eszközök. Grafikus felület. Szintén JPanel-ek, JTextArea-k, JLabel-ek és JToggleButton-ok felhasználásával készült felület. Autó és ház esetében 2 lehetőség közül lehet választani. Vásárlást követően a másik tárgyhoz tartozó gomb inaktívvá válik. A kiválasztott cikket az egyenleg terhére és annak a keretében szerzik be. Ezen tárgyak értéke a Boolean változóban van tárolva, így ez a *BoltGUI.java* szoros logikai kapcsolatban áll a *Jatekos.java*-val (a bolt játékosokként külön van kezelve).

```

39 public BoltGUI(GUI f) {
40     gmf;
41     initComponents();
42     g.setEnabled(false);
43     jToggleButton3.setEnabled(false);
44     jToggleButton4.setEnabled(false);
45
46     jToggleButton7.setEnabled(false);
47     jToggleButton6.setEnabled(false);
48
49     jToggleButton8.setEnabled(false);
50     jToggleButton9.setEnabled(false);
51     jToggleButton10.setEnabled(false);
52     jLabel2.setEnabled(false);
53     jLabel3.setEnabled(false);
54     jLabel4.setEnabled(false);
55     jLabel5.setEnabled(false);
56     setLocationRelativeTo(null);
57     if (g.jatekosok.get(g.aktualisjatekos).getBicikli()) {jToggleButton5.setEnabled(false);}
58     if (g.jatekosok.get(g.aktualisjatekos).getAuto() && !g.jatekosok.get(g.aktualisjatekos).getElektromos()) {jToggleButt
59     else if (g.jatekosok.get(g.aktualisjatekos).getElektromos()) {jToggleButton1.setEnabled(false);}
60     if (g.jatekosok.get(g.aktualisjatekos).getHaz().getVane() && !g.jatekosok.get(g.aktualisjatekos).getHaz().getNapelem
61     else if (g.jatekosok.get(g.aktualisjatekos).getHaz().getNapelem()) {jToggleButton6.setEnabled(false);jToggleButton7.s
62     if (g.jatekosok.get(g.aktualisjatekos).getHaz().getHuto()) {jToggleButton8.setEnabled(false);}
63     if (g.jatekosok.get(g.aktualisjatekos).getHaz().getFurdoszoba()) {jToggleButton10.setEnabled(false);}
64     if (g.jatekosok.get(g.aktualisjatekos).getHaz().getButorok()) {jToggleButton9.setEnabled(false);}
65     if (g.jatekosok.get(g.aktualisjatekos).getHaz().getNapelem() && g.jatekosok.get(g.aktualisjatekos).getHaz().getHuto()

```

7. kódrészlet - BoltGUI.java

4. Jatekos.java:

Nincs külön grafikus felülete. A játékoshoz tartozó változókat tároljuk itt. Minden felhasználóhoz tartozik : név, szín, sorrend, táblán való pozíció, számlaegyenleg, környezeti pont, utolsó dobott érték, boltban vagy nyeremény által szerzett tárgy(ak) külön-külön felsorolva. Ezek többsége boolean, vagy integer típusú változó. Szorosan kapcsolódik a BoltGUI.java-hoz, GUI.java-hoz, Mezo.java-hoz és a Haz.java-hoz.

5. Mezo.java:

Nincs külön grafikus felülete. Szorosan kapcsolódik a GUI.java-hoz és a Jatekos.java-hoz. Feladata, hogy tárolja a mezőket, és az azokhoz esetlegesen kapcsolódó különleges tulajdonságokat. Minden mező rendelkezik felirattal és számmal. Ezen kívül van, ami speciális eseményt hív meg(lásd lottózó), vagy a játékost jutalommal vagy büntetéssel látja el. A különleges események külön eseményszám változóval is rendelkeznek. A Mezo.java kezeli le azt is, hogy hány mező van használatban, azokon hány üres hely van és a nemüres pozíciókat kik töltik be.

6. Haz.java:

Játékosokként külön-külön tároljuk le. Boolean típusú változók, amelyek alapértéke false. Vásárlást vagy tárgynyeremény elérését követően értéke megváltozik. A játékosok csak akkor teljesítik a táblát(akkor fejezték be a játékot), ha minden logikai érték igazra vált.

7. Pont.java:

Mezo.java osztály használja. A játékos bábuk mozgatásához szükséges. Az x és az y által képzett koordinátákkal hivatkozási pontként szolgál.

8. *Tabla.java*:

A játék indításakor a játéktábla létrehozásáért felel. Hiba esetén a hibaüzenetet szolgáltatja. Itt került meghívásra a Mezo.java számára a mezőszövegek a Mezoszovegek.txt-ből.

9. *VegeGUI.java*:

Grafikus felület. Amint az első játékos teljesíti a játék megnyerésének feltételeit, felugrik egy új ablak gratuláló szöveggel. Lehetőség van kilépni és új játékot kezdeni.

IV. A játékot felépítő osztályok listája és a feladat felosztása

A játék létrehozásánál cél volt, hogy a program által használt osztályok jól elkülöníthetően megtalálhatók lehessenek, arra az esetre, ha a későbbiekben fejleszteni vagy módosítani szeretnék a játékot. Ezért a működés szempontjából elhatárolhatóan kezelhető elemek mind-mind külön osztályban, elválasztva kaptak helyet. Munkaszervezés szempontjából is könnyebb volt így dolgozni. A program leírásában már hivatkoztunk a létrehozott osztályokról, és azon szerepéről, funkciójáról, így az alábbi táblázatban csak felsorolás szerűn szedtük össze, feltüntetve, hogy melyikhez ki tartozik.

| | | | |
|-----------------|--------|------------------------|--------|
| BoltGUI.java | Tamara | Pont.java | Tamara |
| GUI.java | Tamara | Quiz.java | Edina |
| Haz.java | Edina | QuizGUI.java | Edina |
| Jatekos.java | Tamara | SportszeletGUI.java | Tamara |
| KezdoGUI.java | Edina | Sugo.java | Edina |
| LottozoGUI.java | Edina | SzerencsekerekGUI.java | Tamara |
| MentesGUI.java | Tamara | Tabla.java | Tamara |
| Mezo.java | Tamara | Tanacs.java | Edina |
| UjAllasGUI.java | Edina | Vege.java | Edina |

V . Fejlesztési lehetőségek

- Játékelmény fokozása szempontjából érdemes a boltban található tárgyak, különös képen a berendezési tárgyak szétválasztása. Több megszerezhető tárgy beépítése játékba, főleg a tudatos fogyasztás témakörében.
- Játékmenet, játékszabály átdolgozása: kevesebb szerep szánása a pénz számára, elmozdulás a környezeti pontok felhasználhatóságának irányába.
- A táblán szerepet kapott Ház grafika funkcióval való ellátása. Képes legyen külön-külön kezelni azokat az elemeket, amiket már megszereztünk és azokat amiket még nem. Választásunknak megfelelően azt a tárgyat jelenítse meg amit már beszereztünk.
- Bábú és dobókocka animáció készítése.
- Program zöldítése, játékos formában való tudatosságra nevelés.
- Gyerekek számára való optimalizálás. Könnyed tanulás, érdekes feladatok beépítése.
- Több játékos, játékosok által választható színek legyenek elérhetőek.