



EÖTVÖS LORÁND TUDOMÁNYEGYETEM  
INFORMATIKAI KAR  
PROGRAMOZÁSELMÉLET ÉS SZOFTVERTECHNOLÓGIAI  
TANSZÉK

Cóatl társasjáték fejlesztése webes  
platformra

*Témavezető:*

Abonyi-Tóth Andor  
Egyetemi docens, PhD

*Szerző:*

Tóth Ábel Tibor  
programtervező informatikus BSc

*Budapest, 2025*

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>4</b>
1.1. Előszó . . . . .	4
1.2. A Board Game Arena platform . . . . .	4
1.2.1. Történet . . . . .	4
1.2.2. Szerkezet . . . . .	5
1.2.3. Fejlesztés . . . . .	7
1.3. A Cójáték társasjáték . . . . .	8
1.3.1. Általános információk . . . . .	8
1.3.2. Rövid szabályzat . . . . .	9
1.3.3. Teljes szabályzat . . . . .	10
1.3.4. Megjegyzés . . . . .	10
<b>2. Felhasználói dokumentáció</b>	<b>11</b>
2.1. A program futtatása . . . . .	11
2.1.1. Regisztráció . . . . .	11
2.1.2. Játék indítása . . . . .	11
2.2. Szerkezet . . . . .	13
2.3. Használat . . . . .	14
2.4. Hibák jelentése . . . . .	21
<b>3. Fejlesztői dokumentáció</b>	<b>22</b>
3.1. Technológiák . . . . .	22
3.2. Felhasználói esetek . . . . .	22
3.2.1. Játék indítása . . . . .	22
3.2.2. Játék kezdete . . . . .	23
3.2.3. Egy forduló . . . . .	23
3.2.4. Egy játékos köre . . . . .	24
3.2.5. Végső forduló . . . . .	25

3.3.	Szerkezeti áttekintés . . . . .	25
3.4.	Adatbázis . . . . .	26
3.4.1.	A player tábla . . . . .	27
3.4.2.	A stats tábla . . . . .	27
3.4.3.	A gamelog tábla . . . . .	28
3.4.4.	Az action_log tábla . . . . .	28
3.4.5.	Az pieces tábla . . . . .	28
3.4.6.	Az prophecies tábla . . . . .	29
3.4.7.	A temples tábla . . . . .	29
3.4.8.	A coatls tábla . . . . .	30
3.4.9.	A bga_global és globals táblák . . . . .	30
3.5.	A játékmodell . . . . .	30
3.5.1.	coatl.game.php . . . . .	31
3.5.2.	states.inc.php, Args.trait.php, States.trait.php . . . . .	32
3.5.3.	actions.inc.php, Actions.trait.php . . . . .	33
3.5.4.	ActionLog.trait.php . . . . .	33
3.5.5.	Pieces.trait.php . . . . .	33
3.5.6.	Prophecies.trait.php . . . . .	33
3.5.7.	Temples.trait.php . . . . .	33
3.5.8.	Cooatls.trait.php . . . . .	33
3.6.	Cóatlelemek . . . . .	34
3.7.	Cóatlök . . . . .	34
3.8.	Próbéciakártyák . . . . .	36
3.9.	Próbéciakártya pontozók . . . . .	36
3.9.1.	Szerkezeti áttekintés . . . . .	37
3.9.2.	A kiértékelő Turing gépek működési elvei . . . . .	37
3.9.3.	Piros Próbéciakártyák . . . . .	38
3.9.4.	Fekete X Próbéciakártyák . . . . .	39
3.9.5.	Fekete trió Próbéciakártyák . . . . .	42
3.9.6.	Kék páros Próbéciakártyák . . . . .	45
3.9.7.	Kék négyes Próbéciakártyák . . . . .	47
3.9.8.	Sárga kettő-kettő Próbéciakártyák . . . . .	49
3.9.9.	Sárga szivárvány Próbéciakártyák . . . . .	53
3.9.10.	Zöld színes végzetlen Próbéciakártyák . . . . .	56

## TARTALOMJEGYZÉK

---

3.9.11. Zöld üres végtelen Prófériakártyák . . . . .	59
3.10. Templomkártyák . . . . .	62
3.11. Templomkártya pontozók . . . . .	63
3.11.1. Szerkezeti áttekintés . . . . .	63
3.11.2. Megvalósítás . . . . .	64
3.12. Felhasználói felület . . . . .	64
3.12.1. Drótvázterv . . . . .	65
3.13. Tesztelés . . . . .	66
3.14. Verzió . . . . .	67
<b>4. Összegzés</b>	<b>68</b>
<b>Köszönetnyilvánítás</b>	<b>69</b>
<b>Forrásjegyzék</b>	<b>69</b>
<b>Ábrajegyzék</b>	<b>71</b>
<b>Forráskódjegyzék</b>	<b>74</b>

# 1. fejezet

## Bevezetés

### 1.1. Előszó

A társasjáték ipar az utóbbi évtizedek során újra fénykorát éli, az egyre népesebb közönségnek és a digitális megoldásoknak köszönhetően. A Board Game Arena platform 2010 óta szolgáltatja széles körben társasjátékok játszását böngészőből. Robbanásszerű növekedésüköt önkéntes programozóknak is köszönhetik, akiknek lehetőséget nyújtanak játékok digitalizálására. A szakdolgozatom célja a Cíatl című társasjáték teljes digitális változatának megvalósítása a Board Game Arena platformon.

### 1.2. A Board Game Arena platform

#### 1.2.1. Történet

A Board Game Arena<sup>1</sup> (a továbbiakban BGA) platformot 2010-ben alapította Grégory Isabelli és Emmanuel Colin, két francia társasjáték rajongó. Az alapötlet az volt, hogy egy olyan felületet hozzanak létre, ahol az emberek online is átélhetik a társasjátékok közösségi élményét, akkor is, amikor nem tudnak személyesen találkozni. A platform teljesen böngészőalapú, ami kifejezetten nagy előnynek számít a mai napig, hiszen nem kell semmit letölteni vagy telepíteni a játékosoknak. Emellett számítógépen, táblagépen és okostelefonokon egyaránt elérhető.

Az első 10 év során a BGA játék katalógusában egyre több népszerű játék vált elérhetővé, például a Carcassonne, a 7 Csoda, az Aranyásók, a BANG!, a Hanabi

---

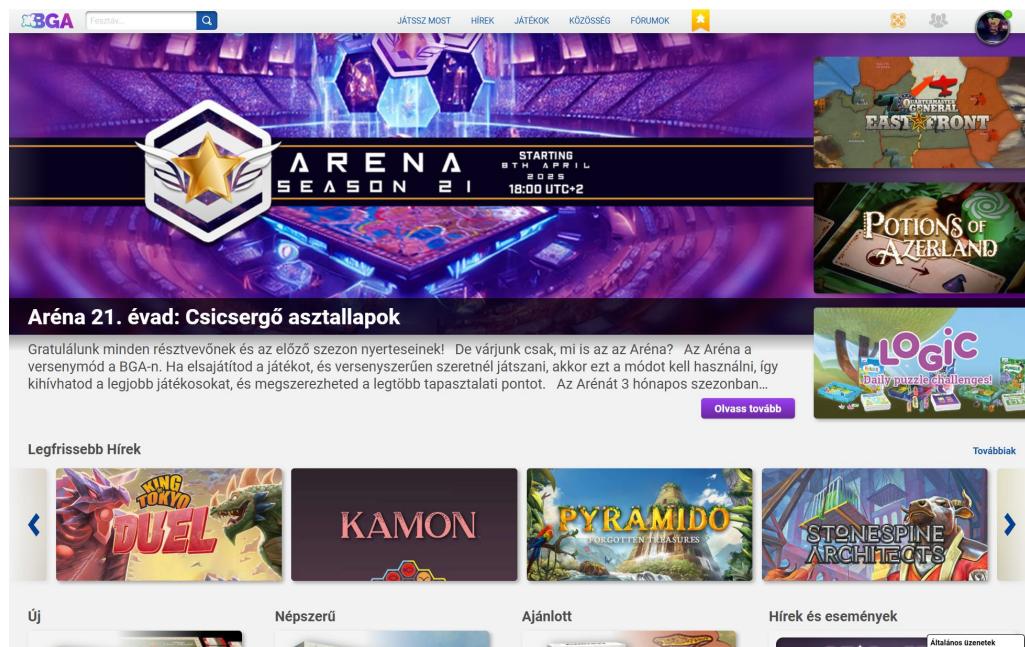
<sup>1</sup><https://boardgamearena.com> (elérés dátuma 2025. 04. 24.)

és még sorolhatnánk. A sok klasszikus cím a platform tekintélyét is emelte és a közösség is nőtt: több százezer felhasználó regisztrált. Lehetőség nyílt az oldal és a játékok szövegeinek közösségi alapú fordítására is, így több nyelven - köztük magyarul is - elérhetővé vált a felület. A 2020-as koronavírus járvány idején a társasjátékos közösségek online térbe kényszerültek. Ez hatalmas löketet adott a platformnak: Drasztikusan megnőtt a felhasználószám. Több játékkiadó is nyitottabbá vált a digitális verziók engedélyezése felé. 2021 januárjában a BGA-t felvásárolta az Asmodee Group, a világ egyik legnagyobb társasjáték kiadója. Bár a lépés aggodalmat keltett a közösségen, a fejlesztők kiálltak amellett, hogy a platform nyitott és közösségi központú maradjon. Ennek a felvásárlásnak hála az oldal fenntarthatósága biztosításra került.

A BGA mára a világ vezető online társasjáték platformjává vált, és továbbra is fejlődik, új játékokat és közösségi eszközöket kínálva a játékosoknak.

### 1.2.2. Szerkezet

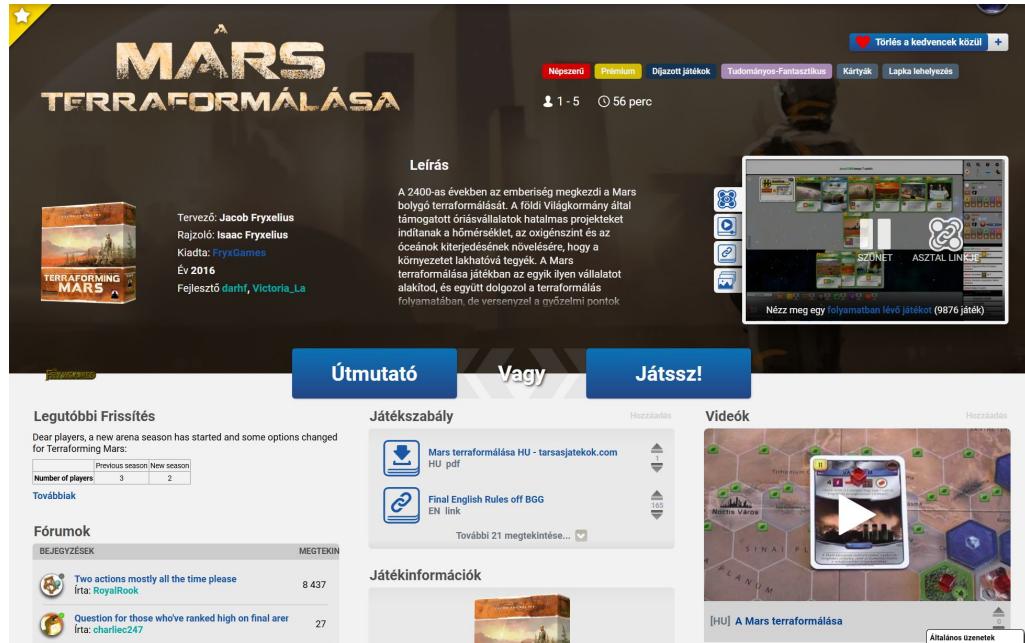
Bejelentkezett felhasználóként a platform főoldalán lehetőségünk van böngészni a legfrissebb játékmegjelenések és hírek között, rákereshetünk egy játékra, megtekinthetjük profilunkat vagy kezdeményezhetünk beszélgetést a többi online felhasználóval.



1.1. ábra. BGA főoldal [1]

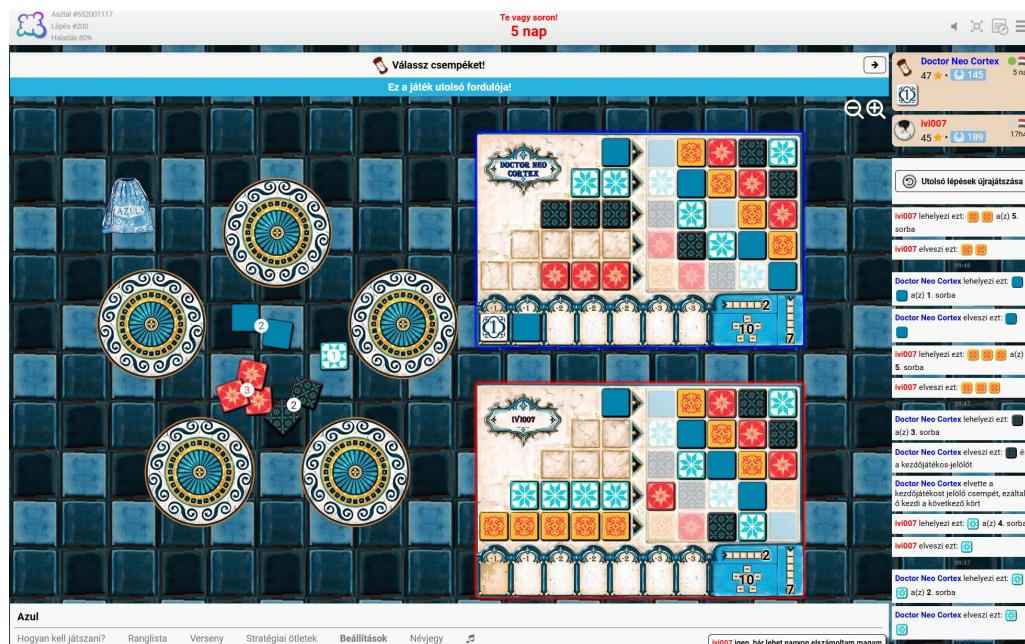
## 1. Bevezetés

Minden elérhető játéknak van saját oldala. Sok egyéb funkció mellett, innen tudunk új asztalt nyitni a játékkal, elolvashatjuk a szabályait vagy megtekinthetjük az aktuális ranglistát.



1.2. ábra. BGA - A Mars Terraformálása: Játék oldal [2]

Miután csatlakoztunk egy asztalhoz és elindult a játék, egy játékképet láthatunk.



1.3. ábra. BGA - Azul: Játék folyamatban [3]



1.4. ábra. BGA - Santorini: Játék folyamatban [4]

### 1.2.3. Fejlesztés

A BGA nyílt lehetőséget biztosít társasjátékok digitalizálására. Elsősorban önkéntes alapon történik a fejlesztés, de a sikeres projekteket a platform és a játékok kiadói juttatásokkal honorálják.

Egy játék csak akkor jelenhet meg az oldalon, ha a jogtulajdonos (kiadó, szerző) hozzájárul. Amikor egy digitalizált játék működőképes, a fejlesztő igényelhet Alfa, majd Béta státuszt a projektjének, ahol felhasználók korlátozott létszámú csoportja kipróbálhatja és visszajelzést adhat a program működéséről. A visszajelzések alapján csiszolható a játékélmény.

Sikerességtől függetlenül a játék elérhetővé válik a nagyközönség számára is a BGA platformon, és bekerül a nyilvános katalógusba.

## 1.3. A Cóatl társasjáték

### 1.3.1. Általános információk

A Cóatl a Synapses Games francia kiadó egyik 2020-ban megjelent társasjátéka. A játékot Pascale Brassard és Etienne Dubois-Roy tervezte, SillyJellie illusztrálta.

A Cóatl egy látványos, könnyen tanulható, mégis taktikus társasjáték 2–4 játékos számára. A játékosok célja, hogy a lehető legszínesebb Cóatlöket (magyarul tollaskígyókat) építsék meg különböző színű elemekből. A játék az azték mitológiából merít inspirációt, mind tematikájában, mind látványvilágában.



1.5. ábra. Cóatl társasjáték doboza [5]



1.6. ábra. Cóatl társasjáték asztalképe [5]

### 1.3.2. Rövid szabályzat

A játék előkészítése során egy közös készletbe kerülnek színes Cójátélemek, illetve felfedésre kerül egy Prófécia- és Templomkártya készlet is. Ezek közös készletek, melyeket minden játékos használhat.

A játékosok körben, egymás után fognak akciókat végrehajtani addig, amíg valamelyikük ki nem váltja a játék végét.

A játék kezdetén minden játékos kap egy saját játékostáblát. Kisorsolásra kerül egy kezdőjátékos, aki megkapja a kezdőjátékos jelölőt és kap 3 Prófériakártyát a pakliból. minden további játékos eggyel több kártyát kap, mint az előtte ülő. Ezen kívül minden játékos kap egy személyes Templomkártyát a kezébe.

Az első kör megkezdése előtt mindenkinél - a kezdőjátékos kivételével - el kell dobni annyi Prófériakártyát a kezéből, hogy három maradjon.

A játékosok a körükben három fő akció közül egyet hajthatnak végre:

- Cójátélem elvétele: A közös készletből választhatnak fej-, test- vagy farok szegmenseket, amiket saját tábláikra helyeznek.
- Prófériakártya elvétele: A közös készletből vagy a pakli tetejéről Prófériakártyát vehetnek fel a kezükbe.
- Cóját építése: Építhetnek egy Cóját a táblájukon lévő Cójátélemek összekapcsolásával és lehelyezhetnek Prófériakártyákat a kezükön, készülő Cójátjeik mellé.

Minden Prófériakártya előir egy színmintát. Egy kártya akkor válik lehelyezhetővé egy készülő Cóját mellé, amikor a színminta megjelenik a Cóját elemeinek színeiben. Egy Cóját befejezetté válik, amikor egy fej és egy farok elem is megjelenik benne. A befejezett Cójátok mellé bizonyos feltételeket elváró Templomkártyák helyezhetők a közös készletből vagy a játékos kezéből.

A játéknak akkor van vége, amikor egy játékos befejezi a harmadik Cójátjét, vagy kifogy a testszelvények készlete. A játék végén a befejezett Cójátok mellé lehelyezett Prófécia- és Templomkártyákért a játékosok pontokat kapnak, majd a legtöbb ponttal rendelkező játékos győz.

### 1.3.3. Teljes szabályzat

A teljes angol nyelvű szabályzat a Synapses Games hivatalos honlapján<sup>2</sup> érhető el. A digitális változat ezen dokumentum alapján készült.

A YouTube videómegosztó oldalon elérhető a d3meeple csatorna "JÁTÉKSZABÁLY: CÓATL" című videója<sup>3</sup>, mely részletes ismertetést ad a játék szabályairól, magyar nyelven.

### 1.3.4. Megjegyzés

A kiadó honlapján fel van tüntetve egy egyjátékos változat szabályzata és egy promó kártyacsomag is, melyet limitált kiadásban lehetett beszerezni a játék megjelenésekor. A szakdolgozatom írása során nem történt egyeztetés a kiadóval ezen változatok implementálásáról, így nem képzik a dolgozatom részét. A program szerkezeti felépítése lehetőséget nyújt az új funkciók, kiegészítések beépítésére.

---

<sup>2</sup>[https://jeuxsynapsesgames.com/wp-content/uploads/2021/06/EN-Coatl\\_rulebook-web\\_new.pdf](https://jeuxsynapsesgames.com/wp-content/uploads/2021/06/EN-Coatl_rulebook-web_new.pdf) (elérés dátuma 2025. 04. 24.)

<sup>3</sup>[https://youtu.be/P7e\\_GZW-up4?si=QlUI0zVEgewuVKkb](https://youtu.be/P7e_GZW-up4?si=QlUI0zVEgewuVKkb) (elérés dátuma 2025. 04. 24.)

## 2. fejezet

# Felhasználói dokumentáció

### 2.1. A program futtatása

Mivel a program a BGA keretrendszerhez készült, így lokális futtatására - a keretrendszer forráskódjának hiányában - nincsen lehetőség. Helyette a BGA főoldalán, korlátozott hozzáférésű Béta tesztfázisban érhető el a játék.

#### 2.1.1. Regisztráció

A program eléréséhez, amennyiben még nem rendelkezünk egy meglévővel, regisztrálnunk kell egy új BGA felhasználói fiókot a platform főoldalán<sup>1</sup>.

#### 2.1.2. Játék indítása

Bejelentkezett BGA felhasználóként a játékkatalógusból érhetjük el a BGA - Cöatl oldalát. Innen tudunk új asztalt nyitni a játékkal, elolvashatjuk a szabályait vagy megtekinthetjük az aktuális ranglistát.

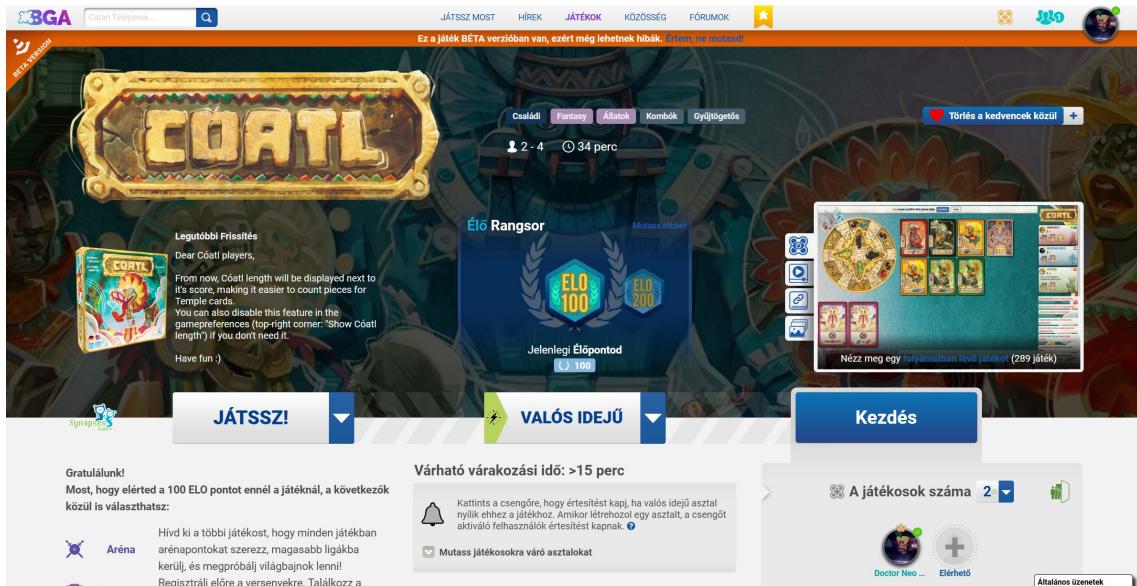
A szakdolgozat írásának időpontjában a Cöatl digitális változatát a Synapses Games kiadó jóváhagyta és engedélyezték a Béta fázisba kerülését. Ez azt jelenti, hogy a főoldalon még nem jelenik meg mindenkinek a játék, de egy link<sup>2</sup> birtokában bárki hozzáférhet.

---

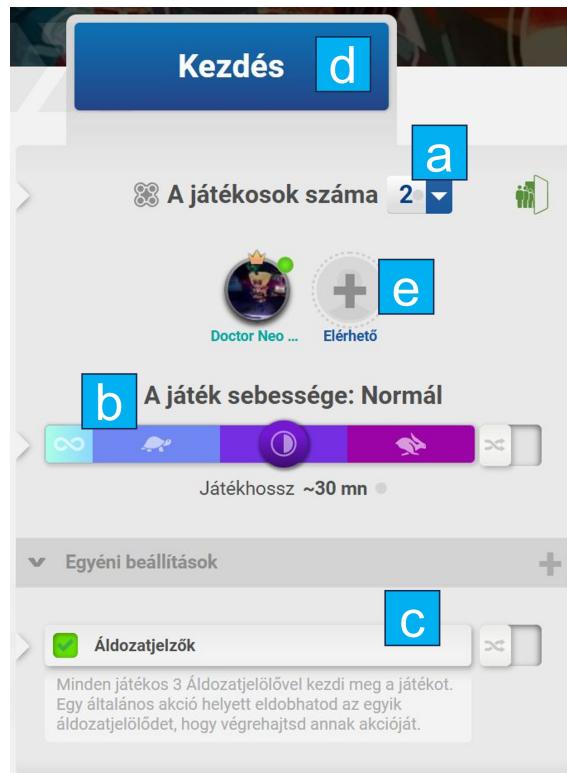
<sup>1</sup><https://boardgamearena.com> (elérés dátuma 2025. 04. 24.)

<sup>2</sup><https://boardgamearena.com/gamepanel?game=coatl> (elérés dátuma 2025. 04. 24.)

## 2. Felhasználói dokumentáció



2.1. ábra. BGA - Cóatl: Játék oldal [6]

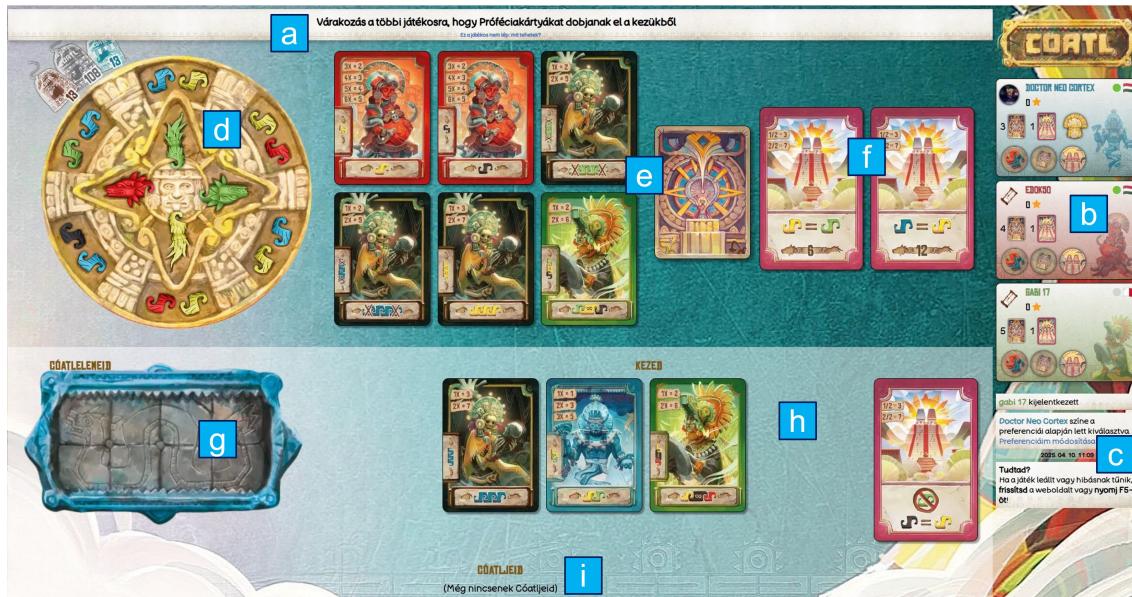


2.2. ábra. BGA - Cóatl: Asztal beállítások [6]

Új asztal indítása előtt beállíthatjuk a játékosszámot [a], a játék sebességét [b], ki és bekapcsolhatjuk az Áldozatjelölők játékvariánst [c]. A kék "Kezdés" gombra [d] kattintva megnyithatjuk az asztalt a többi játékos számára. Lehetőségünk van az elérhető helyekre barátainkat is megírni a "+" gombra [e] kattintva. Amikor a megfelelő számú játékos összegyűlik, a játék elindul.

## 2.2. Szerkezet

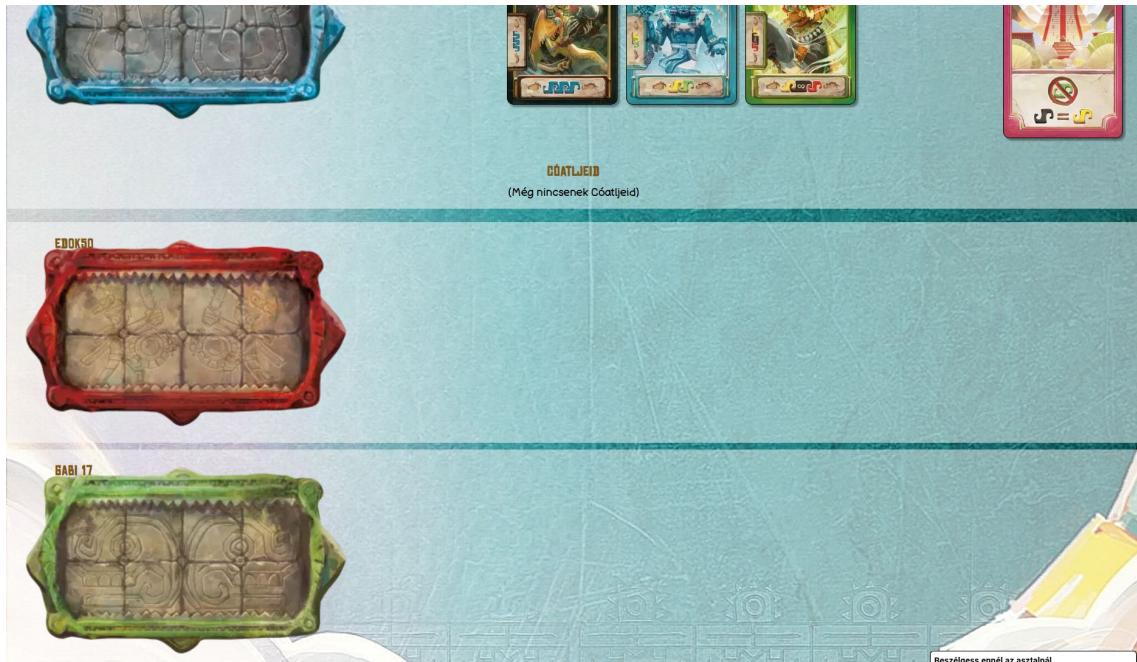
A játék indításakor az előkészítést követő játékállás fogad minket.



2.3. ábra. BGA - Cöatl: Játék kezdőállása [6]

A Cöatl követi a BGA szabványos nézetét. Fent látható az állapotság [a], mely minden játékszakaszban információt nyújt a játékosok lehetséges akcióiról. A jobb oldalon a játékosok listája [b] jelenik meg, tartalmazva a személyes és játék specifikus információkat, esetünkben a játékos pontszámát, színét, a kezdőjátékos jelölőt, a kézben tartott kártyák számát és az Áldozatjelölőket. A játékosok listája alatt a játék naplója [c] látható.

Középen a játékfelület található, melynek elemeire kattintva akciókat hajthatunk végre. A játékfelület három részre bontható. Felül a közös készlet látható, ahol véletlenszerűen felfedett Cöatlelemek [d], Próbáciakártyák [e] és Templomkártyák [f] képezik a minden játékos számára elérhető kínálatot. A közös készlet alatt a játékosnak a saját játékelemei jelennek meg. A játéktábla [g], a kézben tartott Próbácia- és Templomkártyák [h]. Ezen felület alatt fognak megjelenni a játékos által épített Cöatlök [i] is, értelemszerűen a játék kezdetén itt még nincsen semmi. Saját felületünk alatt a többi játékos játéktábláját és Cöatljeit tekinthetjük majd meg.



2.4. ábra. BGA - Cóatl: Többi játékos területe [6]

A többi játékos nézete eltér a mintától: minden játékosnak a saját táblája jelenik meg legfelül, a saját kártyáival. A többi játékos kezében lévő kártyák nem jelennek meg, azok privát információt képeznek.

### 2.3. Használat

A felhasználói felület használatának bemutatása során feltételezzük a játék szabályainak ismeretét. A játékfelület használatához csak a bal egérgomb kattintására, vagy érintőképernyő esetén rövid érintésekre van szükség.

A játék kezdetén az első játékoson kívül mindenkinek el kell dobnia annyi Próbéciakártyát, hogy 3 maradjon a kezében. A kezdőjátékos ilyenkor inaktív a játékban, semmilyen akció nem lehetséges a számára.

**Várakozás a többi játékosra, hogy Próbéciakártyákat dobjanak el a kezükön**

Ez a játékos nem lép: mit tehetek?

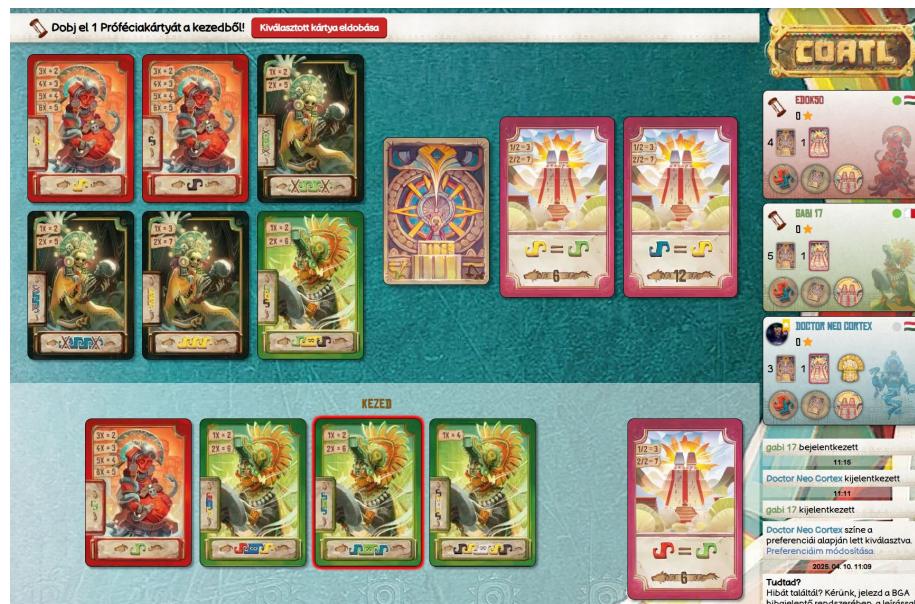
2.5. ábra. BGA - Cóatl: Inaktív játékos állapotság [6]

Amennyiben átváltunk egy aktív játékos nézetére, láthatjuk, hogy számára az állapotság azt az utasítást adja, hogy dobjon el kártyát a kezéből. Ilyenkor ő egy aktív játékos, ezt jelzi az állapotság szövege előtt látható homokóra is.



2.6. ábra. BGA - Cóatl: Aktív játékos állapotsávja [6]

Ebben az esetben a felhasználónak választania kell annyi kártyát a kezéből, amennyi az állapotsávon megjelenik. A kiválasztott kártyák piros körvonalat kapnak. Amikor pontosan az előírt számú kártya kerül kiválasztásra, az állapotsávon lévő gomb pirosra vált. A piros "Kiválasztott kártyák eldobása" gombra történő kattintás után a kézben lévő kártyák eltűnnek és az aktív játékos inaktívvá válik. Ha van még játékos akinek háromnál több Próbéciakártya van a kezében, őt meg kell várniuk a játékosoknak.



2.7. ábra. BGA - Cóatl: Próbéciakártya eldobása [6]

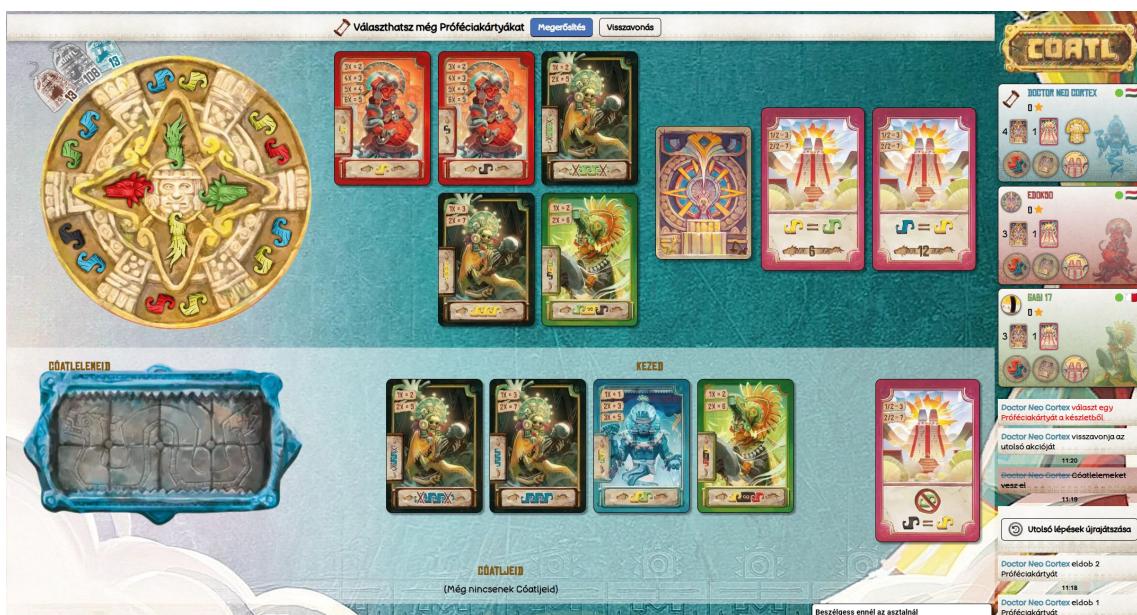
Amikor minden játékosnak 3 Próbéciakártya van a kezében, véget ér a játék kezdő fázisa. Innentől a játékosok szokásos módon, egymás után hajtják végre a köreiket. Egy játékos a körében 3 akció közül választhat: Cójátélemek elvétele, Próbéciakártyák elvétele, Cójáték összerakása.

Cójátélemek elvételéhez kattintsunk a készlettábla egyik szegletére.



2.8. ábra. BGA - Cōatl: Cōatlelemek elvétele [6]

Prófériakártya elvételéhez kattintsunk a készlet egyik kártyájára.



2.9. ábra. BGA - Cōatl: Prófériakártya elvétele a közös készletből [6]

A Prófériakártya készlet melletti paklira kattintva lehetőségünk van felhúzni a pakli legfelső kártyáját, ilyenkor egy párbeszédablak kér meg minket az akció meg-erősítésére.



2.10. ábra. BGA - Cóatl: Prófériakártya húzása a pakliból [6]

Amikor elegendő Cóatlelement és Prófériakártyát gyűjtöttünk, az állapotsávon szereplő "Összerakás" gombra kattintva indíthatjuk a Cóatlök építését. Először egy "Új Cóatl" gomb fog megjelenni. A gombra és egy táblánkon szereplő elemre kattintva új Cóatlt kezdhetünk meg a kiválasztott elem lehelyezésével.



2.11. ábra. BGA - Cóatl: Új Cóatl elkezdése [6]

Ezután az új "+" akciógombok segítségével bővíthetjük az elkezdett Cóatlt. Ezen gombok működéséhez is ki kell választanunk egy Cóatlelement a táblánkról. Amikor sikerül kialakítanunk az egyik Prófériakártyánk követelményében szereplő színmin-tát, lehetőségünk nyílik kijátszani azt a kezünkbelől. Ehhez kattintsunk a "Kártya hozzáadása" gombra majd a megfelelő kártyára. A kártyák lehelyezése után a Cóatl alatt látható pontszám jelölő automatikusan frissül, a Cóathöz kijátszott összértéket mutatva. A lehelyezett kártyák a pontszámjelölő mellett jelennek meg.



2.12. ábra. BGA - Cöatl: Cöatlök építése [6]

Bármikor megtekinthetjük a többi játékos Cöatljeit is a játékterületükön.



2.13. ábra. BGA - Cöatl: Játékostársaink Cöatljei [6]

Amikor egy Cöatl már fejjel és farokkal is rendelkezik, a Cöatl késznek számít. Ilyenkor még utoljára lehetőségünk van lejátszani mellé Prófériakártyákat és egy Templomkártyát is. Az így összegyűjtött pontszám jóváírásra is került a játékospánelünk pontszám részénél.



2.14. ábra. BGA - Cöatl: Egy befejezett Cöatl [6]



2.15. ábra. BGA - Coatl: Játékospantelek az új pontszámmal [6]

A játékospanteleken, a kézben tartott kártyák száma alatt jelennek meg a játékosok Áldozatjelölői. Azokra kattintva a játékosok felhasználhatják őket és aktiválhatják az általuk biztosított alternatív akciókat.

Amikor egy játékos a köre során befejezi a harmadik Cöatljét, kezdetét veszi a végső forduló. Erre egy üzenet is felhívja a figyelmet a státuszsáv alsó részén. A végső forduló során a felület egyértelműen jelzi az aktív játékos számára, hogy még mennyi kört hajthat végre a játék vége előtt.



2.16. ábra. BGA - Cóatl: Végső forduló figyelmeztetés [6]

A játék végén a BGA általános eredménytáblája jelenik meg a képernyőn, a játékosok helyezéseivel és a pontszámaikkal.

The screenshot shows the final score table for the BGA - Cíatlöke game. At the top, it says "Játék vége: gabi 17 nyert" and has buttons for "Visszatérés a főoldalra", "Játszz még egyet", and "Visszavágó ajánlása". Below this, there are tabs for "Játék eredménye", "Végső állás", and "A játék megvásárlása". The main section is titled "JÁTÉK EREDMÉNYE : GABI 17 NYERT" and lists the top three players:

Rang	Játékos	Pontszám
1.	GABI 17	26 ★
2.	DOCTOR NEO CORTEX	20 ★
3.	EDOK50	9 ★

Below the table, there are buttons for "Ici pici szerelmem" and "írj személyes feljegyzést róla: edok50".

2.17. ábra. BGA - Cíatlöke: A játék végeredménye [6]

A végeredmény alatt megtekinthetjük a játékunk statisztikáit.

STATISZTIKÁK			
Játékhossz	53 perc		
Átlagos játékos szint			
Átlagos pontszám	18.3333		
Körök száma	74		
	gabi 17	Doctor Neo Cortex	ed0k50
Játék eredménye	1. (26★)	2. (20★)	3. (9★)
Gondolkodási idő	14:08	24:24	22:32
Körök száma	25	24	25
Elvett Fejszelvények száma	2	3	1
Elvett Testszelvények száma	26	16	30
Elvett Farokszelvények száma	2	3	2
Elvett Prófériakártyák száma	4	6	5
Befejezett Cójatiók száma	2	3	1
Pontszám az 1. befejezett Cójatiért	17	10	9
Pontszám a 2. befejezett Cójatiért	9	6	0
Pontszám a 3. befejezett Cójatiért	0	4	0
Átlagos pontszám egy befejezett Cójatiért	13	6.66667	9
Legmagasabb pontszám egy befejezett Cójatiért	17	10	9
Befejezett Cójatiók átlagos hossza	10.5	7	10
Teljesített Prófériakártyák száma befejezett Cójatióknél	5	6	3
Teljesített Templomkártyák száma befejezett Cójatióknál	2	2	0
"Tökéletes választás" (szelvény) Áldozatjelző felhasználva	igen	igen	nem
"Bepillantás a jövőbe" (prófécia) Áldozatjelző felhasználva	nem	igen	nem
"Papi elhivatottság" (templom) Áldozatjelző felhasználva	nem	igen	nem
 Minden statisztika	 gabi 17 statisztikája ebben a játékban	 Doctor Neo Cortex statisztikája ebben a játékban	 ed0k50 statisztikája ebben a játékban

2.18. ábra. BGA - Cójatl: A játék statisztikái [6]

## 2.4. Hibák jelentése

A BGA lehetőséget nyújt a játékosok számára, hogy visszajelzést küldjenek a fejlesztőknek a hibákról és javaslataikról. A BGA - Cójatl is rendelkezik erre a célra kialakított felülettel<sup>3</sup>.

<sup>3</sup><https://boardgamearena.com/bugs?game=2330> (elérés dátuma 2025. 04. 24.)

## 3. fejezet

# Fejlesztői dokumentáció

### 3.1. Technológiák

A program szerver oldali logikája PHP 8.3, kliens oldali felhasználói felülete JavaScript (Dojo-JS) programnyelveken íródott. A kapcsolatot AJAX-al lett megvalósítva. Adatbázisként a BGA mySQL szervereit használja. A program a BGA saját keretrendszerét<sup>1</sup> használja. A felhasználói felület megjelenítése HTML5 és CSS3 leírónyelkekkel készült.

### 3.2. Felhasználói esetek

#### 3.2.1. Játék indítása

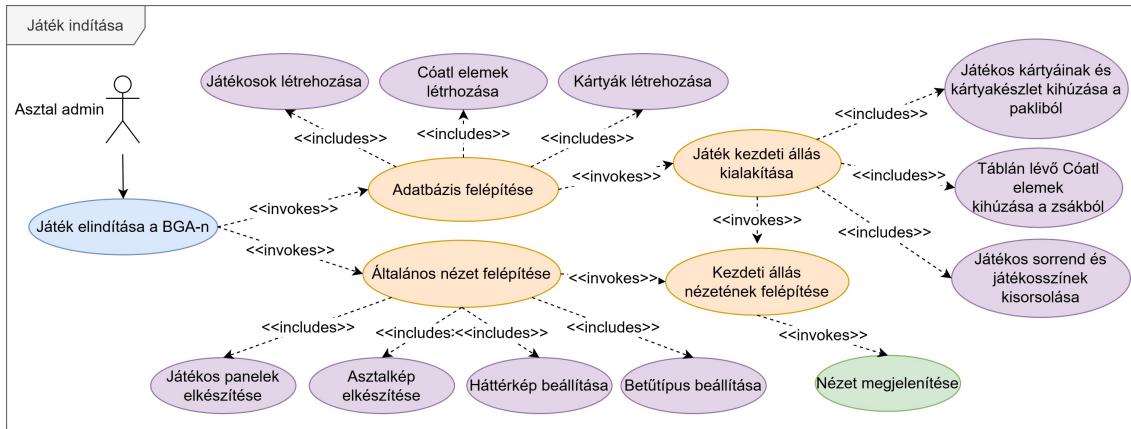
A játék indítása a BGA - Cöatl oldalán<sup>2</sup> történik, a virtuális asztal házigazdája által, miután megfelelő számú játékos csatlakozott az asztalhoz.

Indításkor létrejön az asztal adatbázisa, itt fogjuk tárolni a pillanatnyi játékállás minden információját. Indításkor létrejönnek a játékosok, a Cöatilemek, a Prófécia- és Templomkártyák, melyekből azonnal közös készletet is alakítunk, majd megfelelő számú kártyákat osztunk a játékosok kezébe. Az általános felhasználói nézet kialakítása után megjelenítjük a pillanatnyi játékállást az adatbázisból kinyert adatok alapján. Hasonló folyamat játszódik le a weboldal újratöltésekor is, az adatbázis elemek létrehozásának kivételével.

---

<sup>1</sup><https://en.doc.boardgamearena.com/Studio> (elérés dátuma 2025. 04. 24.)

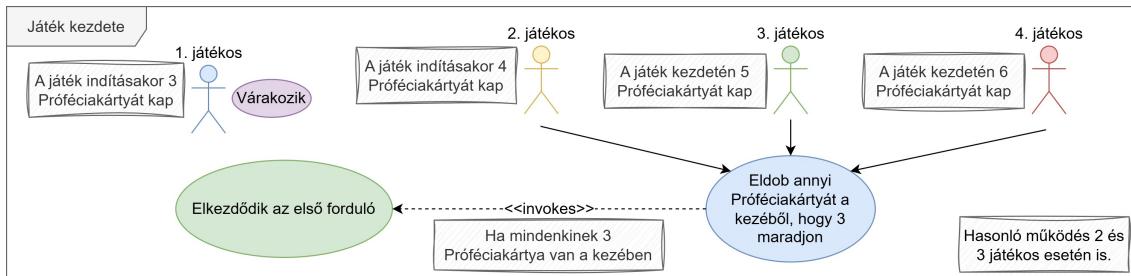
<sup>2</sup><https://boardgamearena.com/gamepanel?game=coatl> (elérés dátuma 2025. 04. 24.)



3.1. ábra. A játék indítása felhasználói eset diagrammal

### 3.2.2. Játék kezdete

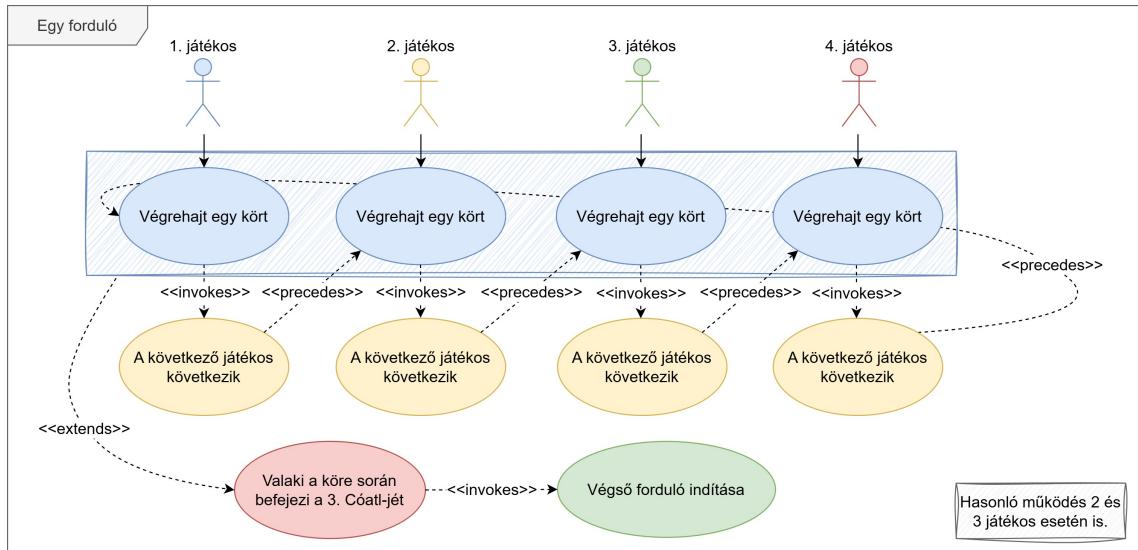
A játék indításakor a kezdőjátékos három Próbáciakártyát kap a kezébe, minden utána következő játékosok pedig egyel többet mint az előző. A játék kezdetén a kezdőjátékoson kívül mindenkinél el kell dobnia annyi Próbáciakártyát a kezéből, hogy három maradjon. Ezt a játékosok szimultán teszik meg. Miután ezzel mindenki végzett, a kezdőjátékos következik és elkezdődik az első forduló.



3.2. ábra. A játék kezdete felhasználói eset diagrammal

### 3.2.3. Egy forduló

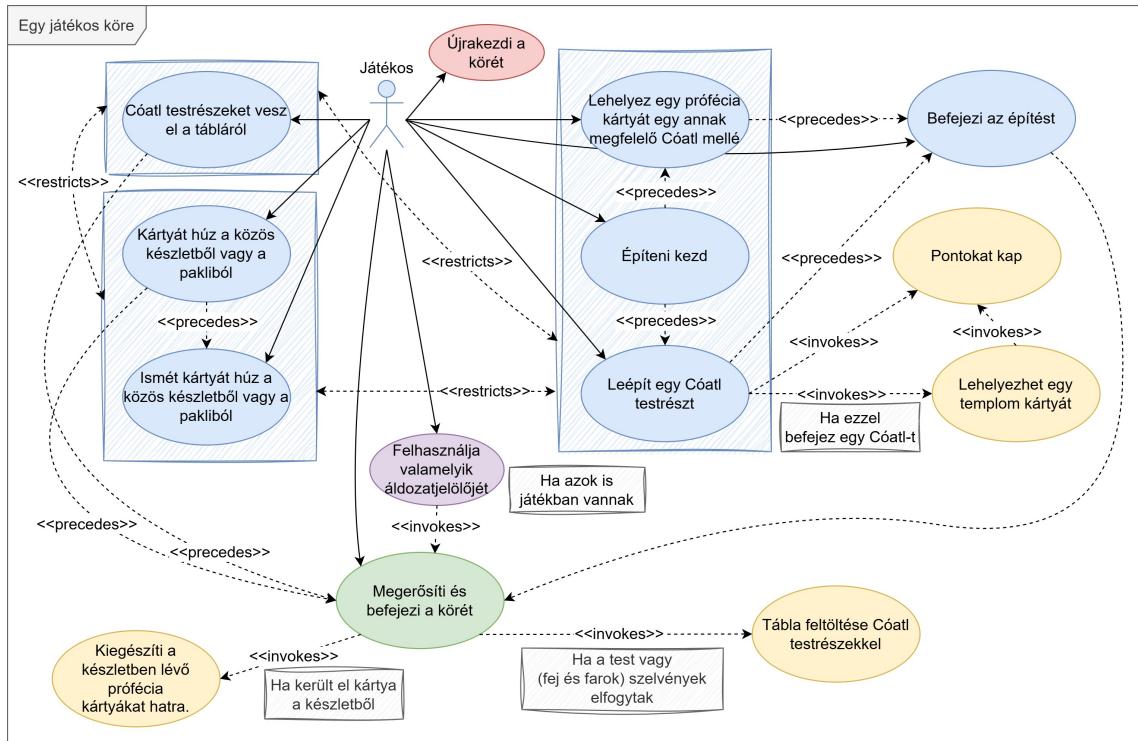
Egy forduló során a játékosok egymás után végrehajtanak egy-egy kört. A fordulóban utolsóként játszó játékos után újból az kezdőjátékos következik és új forduló kezdődik. Amikor egy játékos a köre során befejezi a 3. Cooatljét, elkezdődik a végső forduló.



3.3. ábra. Egy forduló felhasználói eset diagrammal

### 3.2.4. Egy játékos köre

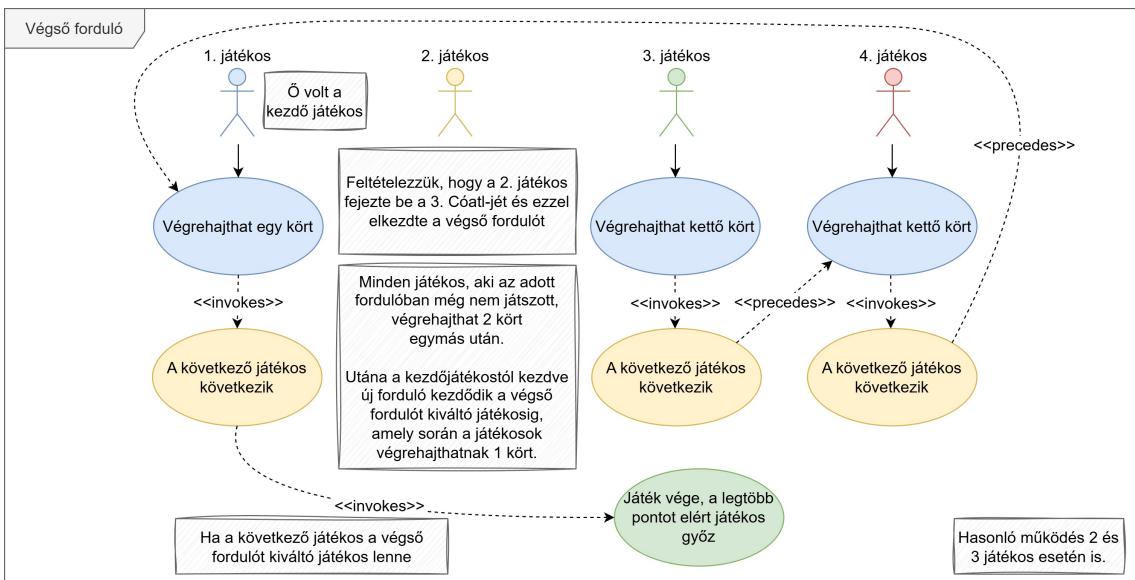
A játékos a köre során végrehajtja a három alap akciót egyikét, vagy használhatja az egyik Áldozatjelölőjét, amennyiben azok is játékban vannak és még nem használta fel őket.



3.4. ábra. Egy játékos köre felhasználói eset diagrammal

### 3.2.5. Végső forduló

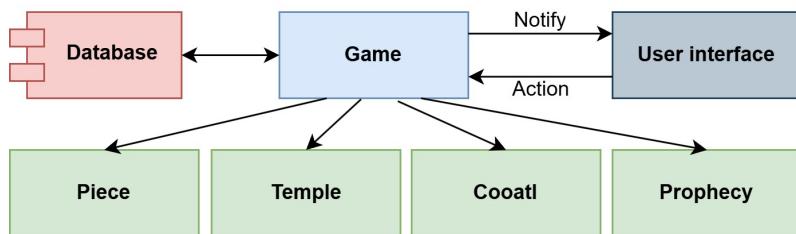
A végső forduló az azt kiváltó játékos után következő játékossal kezdődik. A kezdőjátékos jelölő segítségével megállapítjuk, hogy a játékos hajtott-e végre akciót az utolsó általános fordulóban. Ha nem, a végső fordulóban kettő kört hajt végre egymás után. Amennyiben már hajtott végre akciót, az utolsó fordulóban egy kört hajthat végre. Amikor a végső fordulót kiváltó játékos következne, a játék véget ér. A legtöbb ponttal rendelkező játékos győz.



3.5. ábra. Végső forduló felhasználói eset diagrammal

### 3.3. Szerkezeti áttekintés

A program adatbázis-modell-nézet felépítéssel készült.

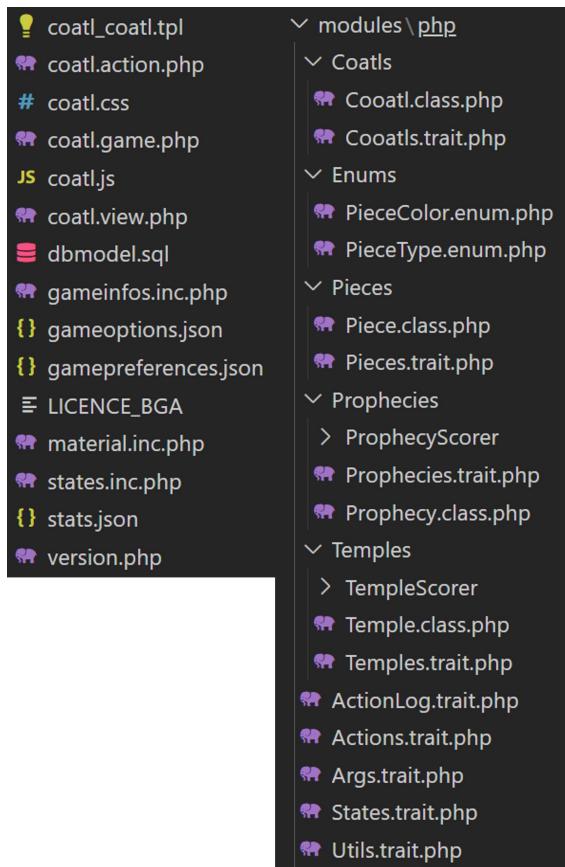


3.6. ábra. A program szerkeze UML diagrammal

Az adatbázis tárol minden információt, ami szükséges egy pillanatnyi állapot leírásához.

A modell (játék) lekérdezi az adatbázisból a pillanatnyi állapotot és játék közben felülírja azokat a játékosok akciói alapján, megállapítja egy játékos lehetséges akciót és számításokat végez a pontszámok meghatározásához. A változásokról értesítéseket küld a felhasználói felületnek. A felhasználói felülettől kapott akciókat kiértékeli, megállapítja, hogy az akció lehetséges-e és végrehajtja az akció hatásait. Külön osztályokkal írjuk le a Cójátélemeket, Prófériákártyákat, Templomkártyákat és Cójáték objektumait, melyeket a modell használ.

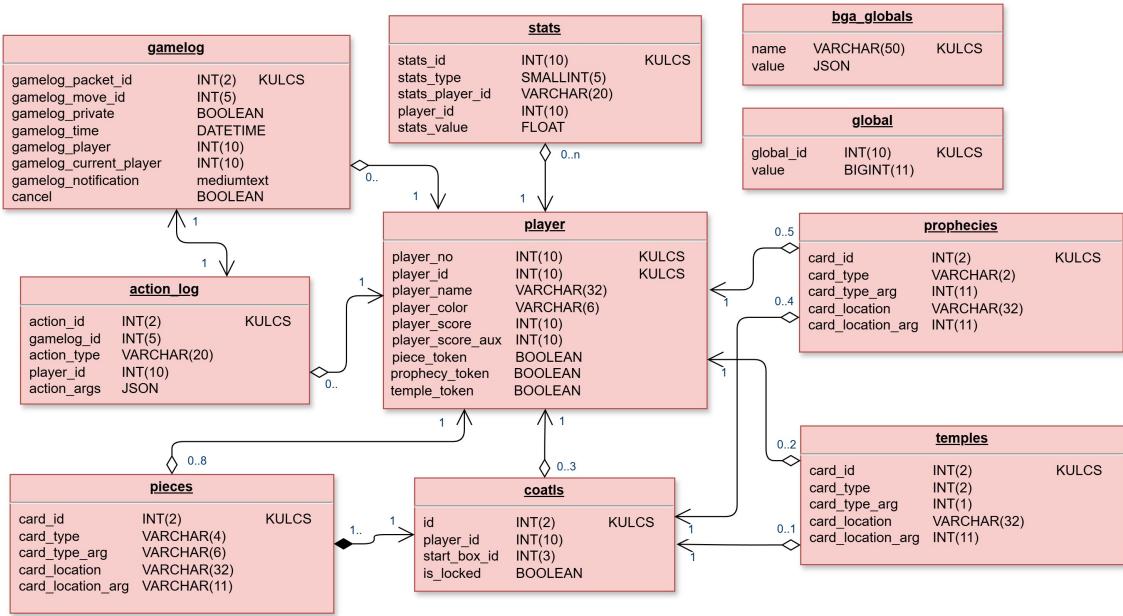
A felhasználói felület megjeleníti az asztalképet a modelltől kapott információk alapján és felépíti a csatlakozási pontokat (gombokat, kattintható elemeket). Egy akció végrehajtásakor jelet küld a modellnek az akció paramétereivel.



3.7. ábra. Az program könyvtárának szerkezete

## 3.4. Adatbázis

Az adatbázis tárol minden információt, ami szükséges egy pillanatnyi állapot leírásához. Ennek eléréshez 10 darab táblát használunk.



3.8. ábra. Az adatbázis szerkezete UML diagrammal

### 3.4.1. A player tábla

A player táblát a BGA keretrendszer automatikusan generálja a játék indításakor, a játékosok felhasználói fiókjai alapján. Általunk használt mezők:

- player\_no: a játékos sorszámát írja le a körsorrend alapján
- player\_id: a játékos egyedi azonosítója
- player\_name: a játékos felhasználóneve
- player\_color: a játékos színe
- player\_score: a játékos pontszáma
- player\_score\_aux: a játékos másodlagos pontszáma, döntetlen feloldásához

Az előre elkészített mezők mellé készítünk további hármat (piece\_token, prophecy\_token, temple\_token) melyek játékspecifikusak. Ezekben a mezőkben tároljuk, hogy a játékos használta-e már az egyes Áldozatjelölőit.

### 3.4.2. A stats tábla

A stats táblát a BGA keretrendszer automatikusan generálja a játék indításakor, a stats.json fájl alapján. A tábla lekérdezéséhez és értékeinek módosításához a

keretrendszer által definiált metódusokat használjuk, közvetlenül nem hajtunk végre rajta műveleteket.

### 3.4.3. A gamelog tábla

A gamelog táblát a BGA keretrendszer automatikusan generálja a játék indításakor, a játék naplózásához. Az itt tárolt bejegyzések jelentek meg játék során a játékospanelek alatt látható naplóban. Az automatikusan generáltak közül egyedül a gamelog\_move\_id rekordot használjuk mely a bejegyzett mozgás azonosítóját tárolja. Ezen kívül készítünk egy cancel mezőt, melyet igazra állítunk, ha az adott bejegyzés által mentett lépés visszavonásra került a felhasználó által, ezáltal ennek megfelelő megjelenést tudunk majd adni ennek a bejegyzésnek a naplóban.

### 3.4.4. Az action\_log tábla

A action\_log táblát a BGA keretrendszer generálja a játék indításakor, a dbmodel.sql fájlban leírtak alapján. Ebben a táblában tároljuk a játékos akcióit, az akció paramétereivel. Az akciók visszavonásához használjuk. Rekordjai:

- action\_id: az akció egyedi azonosítója
- gamelog\_id: az akcióhoz tartozó gamelog bejegyzés gamelog\_move\_id értéke
- action\_type: az akció típusa
- player\_id: az akciót végrehajtó játékos azonosítója
- action\_args: az akció paraméterei

### 3.4.5. Az pieces tábla

A pieces táblát a BGA keretrendszer generálja a játék indításakor, a dbmodel.sql fájlban leírtak alapján. Ebben a táblában tároljuk a játékban lévő Cíatlelemek információit.

- card\_id: a Cíatleme egyedi azonosítója
- card\_type: a Cíatleme típusa (fej, test, farok)
- card\_type\_arg: a Cíatlelem színe (sárga, piros, fekete, zöld, kék)

- card\_location: a Cójáték helye (pl. zsák, készlet, játékos tábla, Cójáték)
- card\_location\_arg: a Cójáték helyének paramétere (pl. pozíció a zsákban)

A pieces tábla a keretrendszer Deck<sup>3</sup> modulja alapján készült.

#### 3.4.6. Az prophecies tábla

A prophecies táblát a BGA keretrendszer generálja a játék indításakor, a db-model.sql fájlban leírtak alapján. Ebben a táblában tároljuk a játékban lévő Próbákkártyák információit.

- card\_id: a Próbákkártya egyedi azonosítója
- card\_type: a Próbákkártya színe (sárga, piros, fekete, zöld, kék)
- card\_type\_arg: a Próbákkártya típusa (misc/card\_types/prophecy.jpg alapján)
- card\_location: a Próbákkártya helye (pl. pakli, készlet, játékos keze, Cójáték)
- card\_location\_arg: a Próbákkártya helyének paramétere (pl. pozíció a pakliban)

A prophecies tábla a keretrendszer Deck<sup>4</sup> modulja alapján készült.

#### 3.4.7. A temples tábla

A temples táblát a BGA keretrendszer generálja a játék indításakor, a db-model.sql fájlban leírtak alapján. Ebben a táblában tároljuk a játékban lévő Templomkártyák információit.

- card\_id: a Templomkártya egyedi azonosítója
- card\_type: a Templomkártya típusa (misc/card\_types/temples.jpg alapján)
- card\_type\_arg: nincs használatban, a keretrendszer viszont kéri a létezését
- card\_location: a Templomkártya helye (pl. pakli, készlet, játékos keze, Cójáték)

---

<sup>3</sup><https://en.doc.boardgamearena.com/Deck> (elérés dátuma 2025. 04. 24.)

<sup>4</sup>3. lábjegyzék

- card\_location\_arg: a Templomkártya helyének paramétere (pl. pozíció a pakliban)

A temples tábla a keretrendszer Deck<sup>5</sup> modulja alapján készült.

### 3.4.8. A coatls tábla

A coatls táblát a BGA keretrendszer generálja a játék indításakor, a dbmodel.sql fájlban leírtak alapján. Ebben a táblában tároljuk a játékban lévő Cöatlök információit.

- id: a Cöatl egyedi azonosítója
- player\_id: a Cöatlt birtokló játékos azonosítója
- start\_box\_id: a Cöatl első elemét tároló doboz azonosítója
- is\_locked: megadja, hogy a Cöatl módosítható-e még, vagy sem

### 3.4.9. A bga\_global és globals táblák

A bga\_global és globals táblákat a BGA keretrendszer automatikusan generálja a játék indításakor. Egyéb játékkal kapcsolatos globális változók tárolására alkalmazhatóak, például a játékvariánsok beállításai innen érhetőek el. A Cöatl programjában a játék végét jelző változókat tároljuk itt. A tábla lekérdezéséhez és értékeinek módosításához a keretrendszer által definiált metódusokat használjuk, közvetlenül nem hajtunk végre rajta műveleteket.

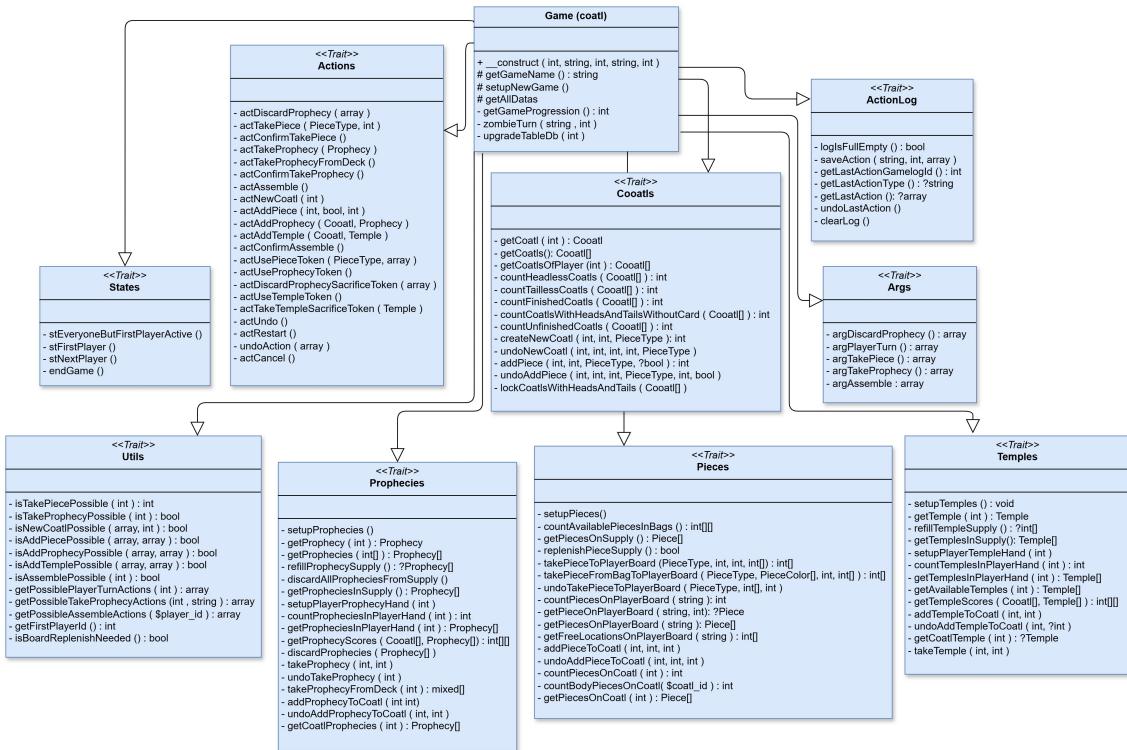
## 3.5. A játékmodell

A játékmodell végzi az adatok lekérdezését az adatbázisból, feldolgozza őket, és továbbküldi őket a felhasználói felületet megjelenítő egység felé. Ennek a fordított irányú folyamatát is kezeli, amikor értesítést kap a felhasználói felülettől feldolgozza azt és elmenti a módosításokat az adatbázisban.

A kód rendszerezésének érdekében a fő játékmodell (coatl) osztály függvényeit és metódusait trait fájlokba rendeztem, ellátott funkcióik alapján.

---

<sup>5</sup><https://en.doc.boardgamearena.com/Deck> (elérés dátuma 2025. 04. 24.)



3.9. ábra. A játékmodell szerkezete UML diagrammal

### 3.5.1. coatl.game.php

A BGA keretrendszer által automatikusan létrehozott játékmodell fő fájl<sup>6</sup>. Itt kerültek definiálásra a keretrendszer által előírt alap függvények és metódusok.

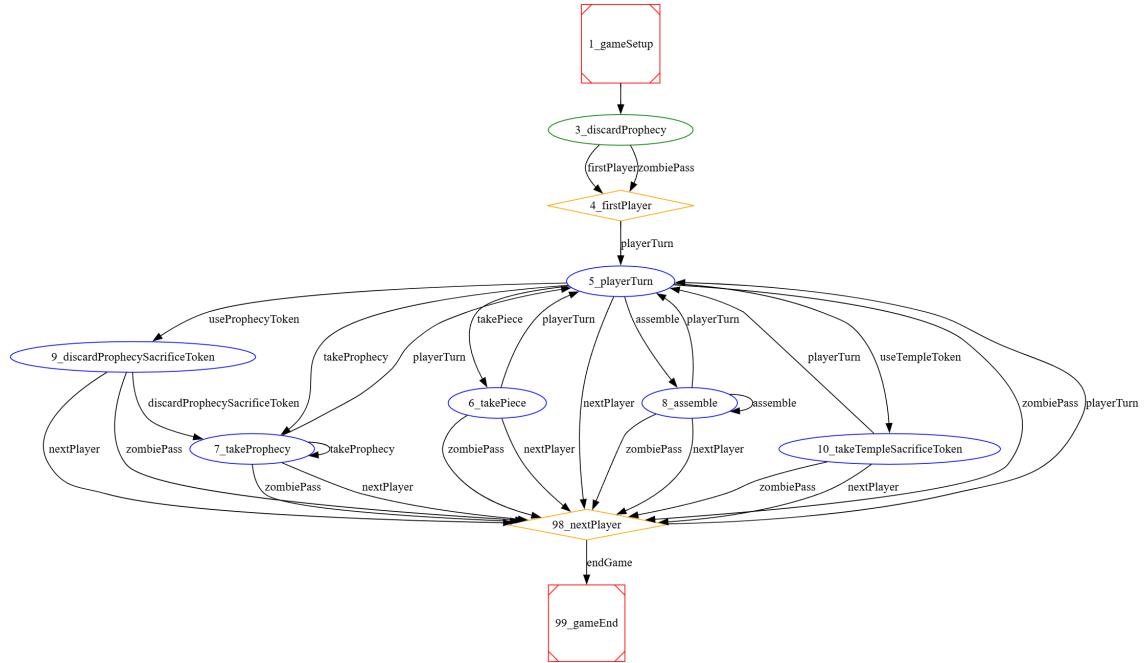
- `getGameName`: keretrendszer része, automatikusan generált.
- `setupNewGame`: a játék indításakor lefutó metódus, a játékelemek előkészítését végzi.
- `getAllDatas`: az oldal betöltésekor lefutó függvény, visszatér a bejelentkezett játékos számára látható összes információval.
- `getGameProgression`: visszatér a játék haladási arányával (0-100 közötti egész szám).
  - 0% az éppen elkezdett játék, 100% a hamarosan véget érő játék.
  - Implementációja során a játékostáblán szereplő Cójátélemeket, a befejezett Cójáték számát és a nem befejezett Cójáték hosszát vettem alapul.

<sup>6</sup>[https://en.doc.boardgamearena.com/Main\\_game\\_logic:\\_Game.php](https://en.doc.boardgamearena.com/Main_game_logic:_Game.php) (elérés dátuma 2025. 04. 24.)

- zombieTurn: a játék asztalát elhagyó játékosok körét leíró metódus.

### 3.5.2. states.inc.php, Args.trait.php, States.trait.php

A BGA keretrendszer által automatikusan létrehozott states.inc.php<sup>7</sup> fájlban definiáljuk a különböző játékállapotokat, a közöttük lévő átmeneteket, a lehetséges akciókat és a státuszszávon megjelenő szöveget is.



3.10. ábra. A játék állapotai és átmenetei

Az állapotok definiálásakor van lehetőség megadni egy függvényt, amely az állapot paramétereivel tér vissza és az állapotba történő átváltáskor megkapja őket a felhasználói felületet megjelenítő egység. Az ehhez definiált függvényeket az Args.trait.php fájlban helyeztem el.

Az állapotok definiálásakor van lehetőség megadni egy metódust, amely lefut az állapotba történő átváltáskor. Az ehhez definiált függvényeket a States.trait.php fájlban helyeztem el.

<sup>7</sup>[https://en.doc.boardgamearena.com/Your\\_game\\_state\\_machine:\\_states.inc.php](https://en.doc.boardgamearena.com/Your_game_state_machine:_states.inc.php) (elérés dátuma 2025. 04. 24.)

### 3.5.3. actions.inc.php, Actions.trait.php

A BGA keretrendszer által automatikusan létrehozott actions.inc.php<sup>8</sup> fájlban definiáljuk a felhasználó felülettől beérkezhető akciókat. Az itt definiálásra került metódusok átalakítják a beérkezett akció paramétereit és továbbítják őket modell felé. A továbbításkor meghívott modell akció metódusokat az Actions.trait.php fájlban helyeztem el.

### 3.5.4. ActionLog.trait.php

Az action\_log adatbázis tábla lekérdezéséhez és módosításához definiált függvényeket és metódusokat az ActionLog.trait.php fájlban helyeztem el.

### 3.5.5. Pieces.trait.php

A pieces adatbázis tábla lekérdezéséhez és módosításához illetve a Piece objektumok létrehozásához és velük való számításokhoz definiált függvényeket és metódusokat a Pieces.trait.php fájlban helyeztem el.

### 3.5.6. Prophecies.trait.php

A prophecies adatbázis tábla lekérdezéséhez és módosításához illetve a Prophecy objektumok létrehozásához és velük való számításokhoz definiált függvényeket és metódusokat a Prophecies.trait.php fájlban helyeztem el.

### 3.5.7. Temples.trait.php

A temples adatbázis tábla lekérdezéséhez és módosításához illetve a Temple objektumok létrehozásához és velük való számításokhoz definiált függvényeket és metódusokat a Temples.trait.php fájlban helyeztem el.

### 3.5.8. Cooatls.trait.php

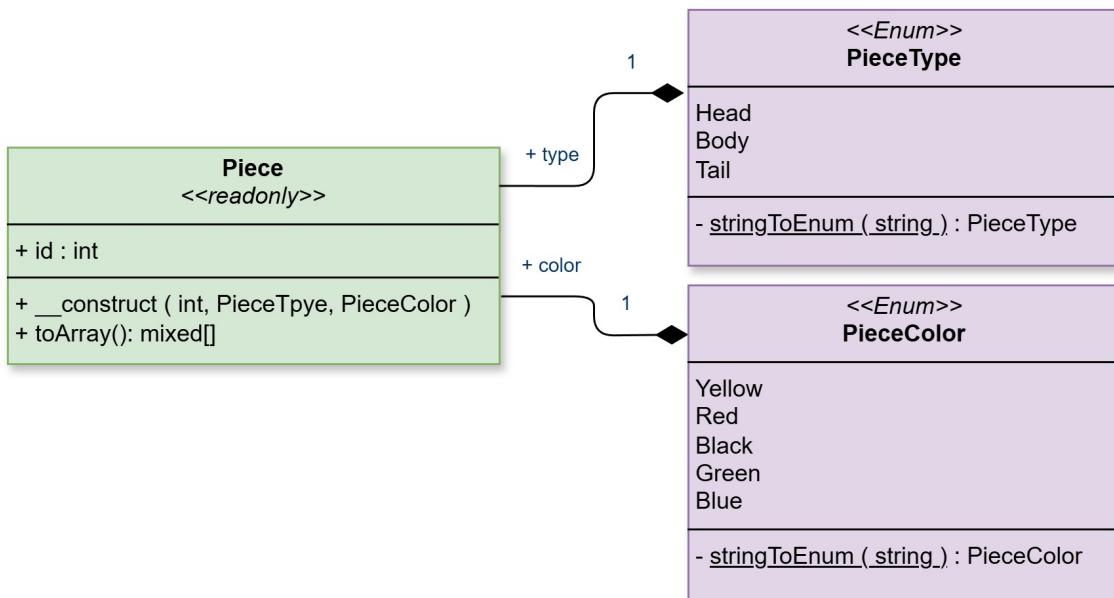
Az coatls adatbázis tábla lekérdezéséhez és módosításához illetve a Cooatl objektumok létrehozásához és velük való számításokhoz definiált függvényeket és metódusokat a Cooatls.trait.php fájlban helyeztem el.

---

<sup>8</sup>[https://en.doc.boardgamearena.com/Players\\_actions:\\_yourgamenname.action.php](https://en.doc.boardgamearena.com/Players_actions:_yourgamenname.action.php) (elérés dátuma 2025. 04. 24.)

### 3.6. Cоатлемек

A csak olvasható Cоатлемек objektumok rendelkeznek egy azonosítóval, egy típus és egy szín tulajdonsággal. A Cоатлемек típusaihoz és színeihez felsoroló osztályt definiáltam.

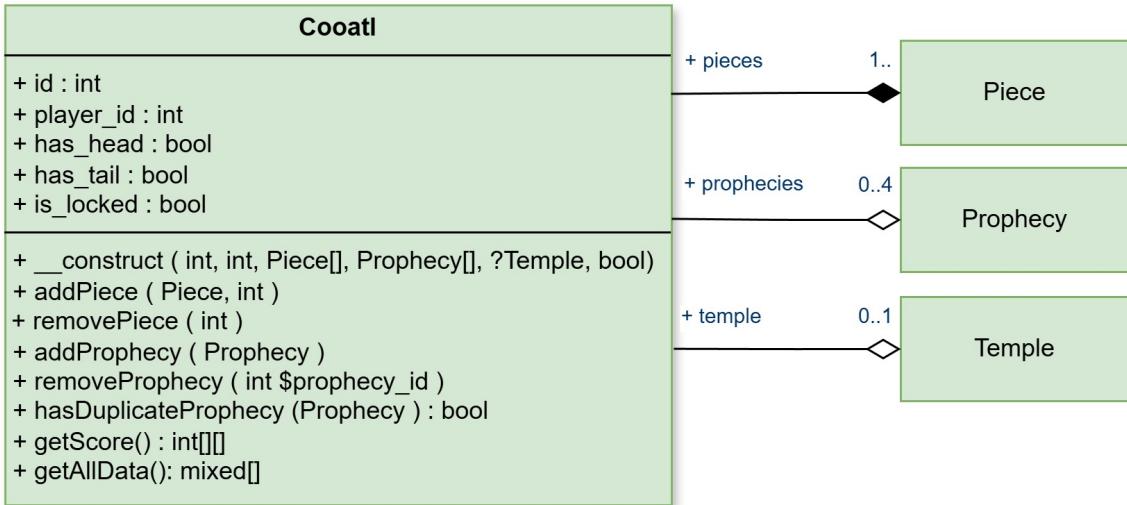


3.11. ábra. Cоатлемек szerkezete UML diagrammal

A Piece objektumok rendelkeznek egy `toArray()` metódussal, amely visszatér az objektum adatainak tömbös reprezentációjával. A felsoroló típusok átalakításra kerülnek szövegekké.

### 3.7. Cоатлök

A Cоатл objektumok rendelkeznek egy azonosítóval, egy játékosazonosítóval ami a Cоатл birtokló játékost jelöli és "van\_feje", "van\_farka", "lezárt" igaz/hamis tulajdonságokkal. A Cоатл rendelkezhet 0-4 Prофéciákártyával és 0-1 Templomkártyával, illetve rendelkeznie kell 1 vagy több Cоатлемemmel melyeket egy asszociatív tömbben felsorolva tárolunk úgy, hogy az elemek kulcsa növekvő sorrendben megmutatja a Cоатлben elfoglalt helyüket.



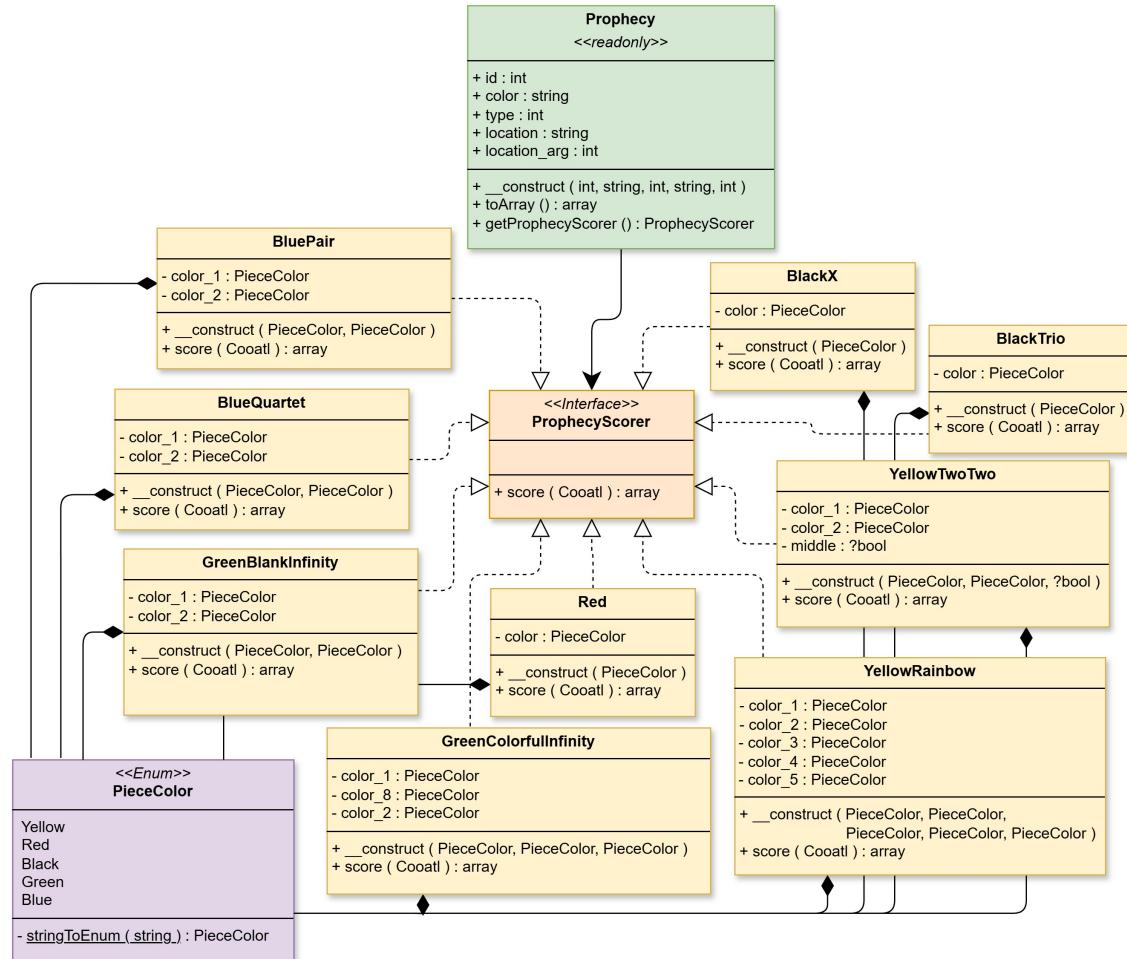
3.12. ábra. Cooatlök szerkezete UML diagrammal

Függvények és metódusaik:

- addPiece, removePiece, addProphecy, removeProphecy: Cooatlelemek és Próféciakártyák hozzáadása és eltávolítása a Cooatl megfelelő adattagjaiból.
  - Alapvetően az adatbázis módosításával adunk hozzá Cooatlelemeket a Cooatlkhöz, majd ezket kérdezzük le és hozunk létre az adatok alapján egy új Cooatl objektumot. Viszont előfordul, hogy nem szeretnénk a teljes Cooatlt lekérdezni egy módosítás után. Ilyenkor egy korábban lekérdezett objektumot módosítunk ezekkel a metódusokkal.
- hasDuplicateProphecy: megvizsgálja az argumentumként megkapott Próféciakártyáról, hogy van-e már azonos típusú kártya lehelyezve.
  - Erre azért van szükség, mert vannak egyforma Próféciakártyák a játékban és egy Cooatl mellé nem helyezhető le két egyforma kártya.
- getScore: visszatér a Cooatl mellé lehelyezett kártyák összesített pontértékével és a kártyák szintjeivel.
- getAllData: visszatér az objektum adatainak tömbös reprezentációjával. A Cooatlelemek felsoroló típusai átalakításra kerülnek szövegekké.

## 3.8. Próbéciakártyák

A csak olvasható Próbéciakártya objektumok rendelkeznek egy azonosítóval, egy szín, egy típus, egy hely és egy hely\_arg paraméter tulajdonsággal.



3.13. ábra. Próbéciakártyák szerkezete UML diagrammal

A Próbéciakártya objektumok rendelkeznek egy `toArray()` metódussal, amely visszatér az objektum adatainak tömbös reprezentációjával. Ezen kívül rendelkeznek egy `getProphecyScorer` függvénytel, amely egy Próbéciakártya pontozó interfészt magvalósító osztály objektumával tér vissza, a Próbéciakártya típusától függően.

## 3.9. Próbéciakártya pontozók

A játék során a játékosok Próbéciakártyákat helyezhetnek le a kezükön a készülő Cooatljeik mellett, hogy győzelmi pontokat szerezzenek. minden Próbéciakártya alján látható egy minta. Vannak amik egy előfordulást pontoznak és olyanok is amik annál

több pontot érnek, minél többször látható a minta egy Cooatlben. A különböző szintek pontszámainak meghatározásához a kártya bal felső sarkában szereplő táblázat nyújt segítséget.

### 3.9.1. Szerkezeti áttekintés

A programban a Próbáciakártyák pontozását erre kialakított objektumok végzik. A Próbáciakártyákat a Prophecy osztály ábrázolja, ezen osztály rendelkezik egy getProphecyScorer() függvénytel, amely visszatér a kártya típusának megfelelő pontozó objektummal. Kilenc pontozó osztály implementálja a ProphecyScorer interfészét. A ProphecyScorer interfésznek egyetlen score függvénye egy Cooatl típusú objektumot vár bemenetként - ez az a Cooatl amit ki szeretnénk értékelni a kártyával - és egy ["pontszám" => egész szám, "szint" => egész szám] szerkezetű tömbbel tér vissza. A "pontszám" kulcsa tartalmazza a kártya kijátszásáért járó pontszámot, a "szint" kulcsa pedig a teljesített szintet.

A kilenc pontozó osztály kiterjeszti a ProphecyScorer interfészét, saját változókat vesznek fel aszerint, hogy mennyi Cooatlelem szín jelenik meg a kártyákon. A kiértékelő algoritmusaiak Turing gépekkel kerültek ábrázolásra és implementálásra. Egy adott kártyához tartozó Turing gép a teljesített szintek számát állapítja meg, vagyis azt, hogy a kártyán szereplő minta mennyiszer szerepel a Cooatlben. A pontszámot ezen érték alapján határozzuk meg.

### 3.9.2. A kiértékelő Turing gépek működési elvei

- Determinisztikus, kettő szalagos Turing gépeket használunk.
  - Az első szalagon jobbról-balra, sorrendben szerepelnek a vizsgált Cooatlben szereplő elemek színei (ezek lehetséges értékei: piros, sárga, fekete, kék, zöld).
  - A második szalagra (munkaszalagra) annyi darab 1-est fog írni a gép, amennyi szintet teljesít a Cooatl.
- Offline Turing gépeket használunk, mivel az első szalagot nem írjuk felül, a kiértékelés során a Cooatl szerkezete nem módosul.
- Amennyiben végére érünk az első szalagnak, a gép leáll.

- minden elemet egyszer vizsgálnak meg, így gépek időkorlátosak. Futási idejük a vizsgált Cíatl hosszával egyenlő.

### 3.9.3. Piros Prófériakártyák

A piros Prófériakártyák egyetlen szín előfordulását várják el, így ezt a színt fogjuk tekinteni a kártya által meghatározott színnek. Amennyiszer szerepel az adott szín a Cíatlban, annyi szint teljesül. A táblázat a 3. szinttel kezdődik és a 6. szintig tart, így az 1. és 2. szint teljesítéséért nem jár pont. Ha több mint 6 szint teljesül, akkor is a 6. szintért járó pontszám kerül elszámolásra.



3.14. ábra. Példa piros Prófériakártyára [5]

A Turing gép három állapottal dolgozik: 0 (kezdő állapot), igen, nem.

- Az első szalag felett lévő fej minden lépés után jobbra mozog.
- A második szalagot színtek teljesítésének a könyvelésre használjuk. Egy szint könyveléskor egy 1-es számot ír és jobbra mozog. Egyébként nem mozog és nem ír.
- A 0-ás állapotban: Ha az első szalagon a Prófériakártya által elvárt szín jelenik meg, elkönyvelődik egy szint teljesítése és a gép marad a 0-ás állapotban. Egyébként marad a 0-ás állapotban.
- Amennyiben az első szalagon üresre érkezünk, elérünk a Cíatl végét, a gép leáll.

$M = \langle \{0, \text{igen}, \text{nem}\}, \{\text{red circle}, \text{yellow circle}, \text{black circle}, \text{green circle}, \text{blue circle}\}, \{\text{red circle}, \text{yellow circle}, \text{black circle}, \text{green circle}, \text{blue circle}, 1, \text{ü}\}, \delta, 0, \text{igen}, \text{nem} \rangle$				
$\delta$	$\text{yellow circle}, \text{ü}$	$\text{purple square} (\text{red circle} / \text{black circle} / \text{green circle} / \text{blue circle}), \text{ü}$	$\text{ü}, \text{ü}$	továbbiak
0	$0 \xrightarrow{\text{yellow circle}}$ $1 \xrightarrow{\text{ }}$	$0 \xrightarrow{\text{purple square}}$ $\text{ü} \xrightarrow{\text{ }}$	igen	nem



3.15. ábra. Piros Próféciakártyák működésének ábrázolása Turing géppel

Az implementációban minden szint teljesítése után megvizsgáljuk, hogy elérteük-e az utolsó (6.) szintet. Amennyiben igen, visszatérünk a legmagasabb pontszámmal. A többi Cíatlelem már nem befolyásolhatja ezt.

```

1 $level = 0;
2 foreach ($coatl->pieces as $piece) {
3     if ($piece->color === $this->color) {
4         $level++;
5         if ($level === 6) return array("score" => 5, "level" => 4);
6     }
7 }
8 if ($level < 3) return array("score" => 0, "level" => 0);
9 else if ($level == 3) return array("score" => 2, "level" => 1);
10 else if ($level == 4) return array("score" => 3, "level" => 2);
11 else return array("score" => 4, "level" => 3);
    
```

3.1. forráskód. Red.class.php részlet

### 3.9.4. Fekete X Próféciakártyák

A fekete X Próféciakártyák 2-4 elem hosszúságú mintát várnak el. Ezek az elemek mindenkor azonos színűek, így ezt a színt fogjuk tekinteni a kártya által meghatározott színnek. Az elemeknek egymás mellett folytatónak kell szerepelniük a Cíatlben. Az 1. és 4. helyeken nem szerepelhetnek az áthúzott elemek. Ez jelentheti azt, hogy az ott lévő elemek a kártya által meghatározotttól eltérő színűek kell lennie, vagy azt, hogy egyáltalán nem lehet ott elem. A 2. és a 3. helyeken a kártya által meghatározott színű Cíatlelemeknek kell szerepelniük. Amennyiszer szerepel ezen minta egy Cíatlben, annyi szint teljesül. A táblázat kettő szintet különböztet meg. Ha egy szint sem teljesül akkor nem jár pont. Ha több mint kettő szint teljesül, akkor is a 2. szintért járó pontszám kerül elszámolásra.



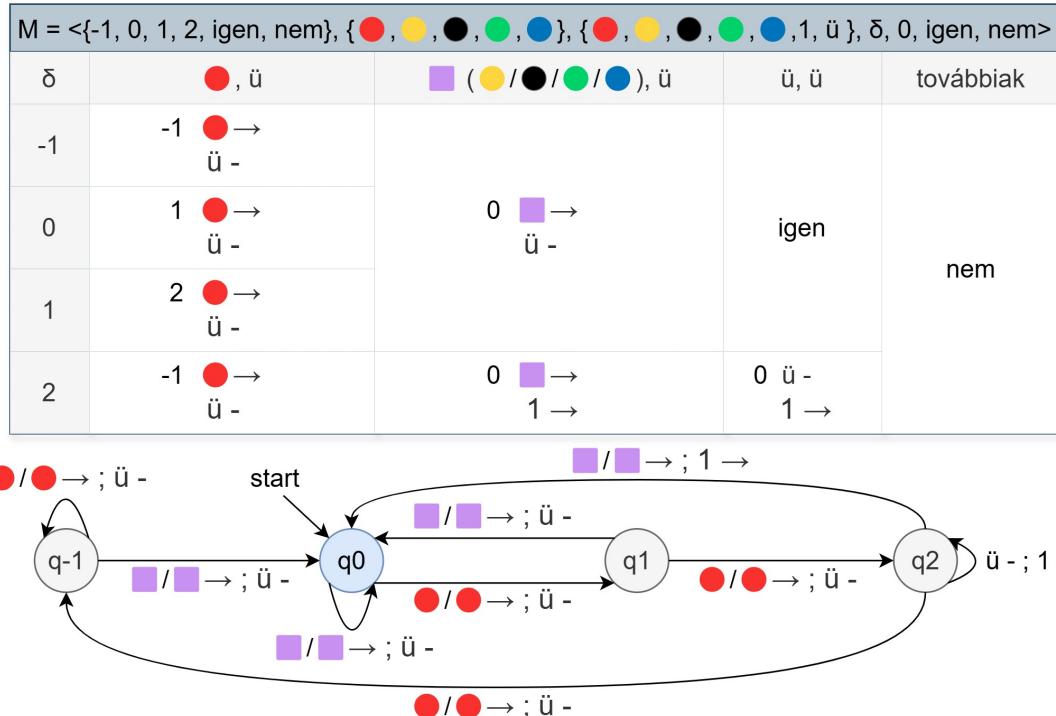
3.16. ábra. Példa fekete X Prófériakártyára [5]

A Turing gép hat állapottal dolgozik: -1, 0 (kezdő állapot), 1, 2, igen, nem. Szándékosan nem a -1-es állapot a kezdő, hiszen az is megfelelő, amennyiben az első áthúzott elem helyén egyáltalán nem szerepel elem a Cíatlban.

- Az első szalag felett lévő fej minden lépés után jobbra mozog.
- A második szalagot színtek teljesítésének a könyvelésre használjuk. Egy szint könyveléskor egy 1-es számot ír és jobbra mozog. Egyébként nem mozog és nem ír.
- A 0-ás állapotban: Ha az első szalagon a kártya által meghatározott szín jelenik meg, a gép átvált az 1-es állapotba. Egyébként marad a 0-ás állapotban.
- Az 1-es állapotban: Ha az első szalagon a kártya által meghatározott szín jelenik meg, a gép átvált a 2-es állapotba. Egyébként visszavált a 0-ás állapotba.
- A 2-es állapotban: Ha az első szalagon a kártya által meghatározott szín jelenik meg, a gép átvált a -1-es állapotba: ezen a helyen nem szabadott volna ilyen elemek szerepelnie. Egyébként elkönyvelődik egy szint teljesítése és a gép visszavált a 0-ás állapotba.
- A -1-es állapotban: Ha az első szalagon a kártya által meghatározott szín jelenik meg, a gép marad a -1-es állapotban: itt egy háromnál hosszabb sorozat alakult ki a kártya által meghatározott színből, biztosan nem lehet egy helyes minta része. Egyébként a gép átvált a 0-ás állapotba.
- Amennyiben az első szalagon üresre érkezünk, elértük a Cíatl végét, a gép leáll. Kivéve, ha a 2-es állapotban voltunk a Cíatl vége elérésekor. Ilyenkor

előbb elkönyvelődik egy szint teljesítése és ezután áll le a gép. Erre azért van szükség, mert a minta akkor is helyes, amennyiben az X-el jelzett Cöatlelem nincs jelen a Cöatlben.

- Ezen Turing gép időkorlátja kivételesen Cöatl\_hossz + 1 lesz.



3.17. ábra. Fekete X Prófériakártya működésének ábrázolása Turing géppel

Az implementációban minden szint teljesítése után megvizsgáljuk, hogy elérünk-e az utolsó (2.) szintet. Amennyiben igen, visszatérünk a legmagasabb pontszámmal. A többi Cöatlelem már nem befolyásolhatja ezt.

A Cöatl végére érve megvizsgáljuk, hogy a 2-es állapotban voltunk-e, amikor végére értünk a Cöatlnak. Ha igen, elkönyvelődik egy szint teljesítése.

```

1 $state = 0;
2 $level = 0;
3 foreach ($coatl->pieces as $piece) {
4     switch ($state) {
5         case -1:
6             if ($piece->color !== $this->color) $state = 0;
7             break;
8         case 0:
9             if ($piece->color === $this->color) $state = 1;
10            break;

```

```

11         case 1:
12             if ($piece->color === $this->color) $state = 2;
13             else $state = 0;
14             break;
15         case 2:
16             if ($piece->color !== $this->color) {
17                 $level++;
18                 if ($level === 2) return array("score" => 5, "level" => 2);
19                 $state = 0;
20             }
21             else $state = -1;
22             break;
23     }
24 }
25 if ($state == 2) {
26     $level++;
27     if ($level === 2) return array("score" => 5, "level" => 2);
28 }
29 if ($level == 0) return array("score" => 0, "level" => 0);
30 else return array("score" => 2, "level" => 1);

```

3.2. forráskód. BlackX.class.php részlet

### 3.9.5. Fekete trió Próbéciakártyák

A fekete trió Próbéciakártyák 3 elem hosszúságú mintát várnak el. Ezek az elemek mindenkor azonos színűek, így ezt a színt fogjuk tekinteni a kártya által meghatározott színnek. Az elemeknek egymás mellett folytatónak kell szerepelniük a Cíatlban. A táblázat kettő szintet különböztet meg. Ha egy szint sem teljesül akkor nem jár pont. Ha több mint kettő szint teljesül, akkor is a 2. szintért járó pontszám kerül elszámolásra.

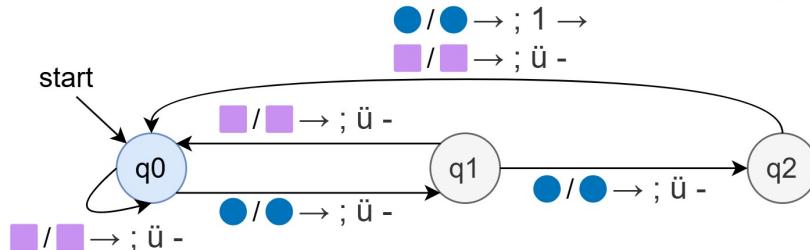


3.18. ábra. Példa fekete trió Prófeciákártyára [5]

A Turing gép öt állapottal dolgozik: 0 (kezdő állapot), 1, 2, igen, nem.

- Az első szalag felett lévő fej minden lépés után jobbra mozog.
- A második szalagot színtek teljesítésének a könyvelésre használjuk. Egy szint könyveléskor egy 1-es számot ír és jobbra mozog. Egyébként nem mozog és nem ír.
- A 0-ás állapotban: Ha az első szalagon a kártya által meghatározott szín jelenik meg, a gép átvált az 1-es állapotba. Egyébként marad a 0-ás állapotban.
- Az 1-es állapotban: Ha az első szalagon a kártya által meghatározott szín jelenik meg, a gép átvált a 2-es állapotba. Egyébként visszavált a 0-ás állapotba.
- A 2-es állapotban: Ha az első szalagon a kártya által meghatározott szín jelenik meg, elkönyvelődik egy szint teljesítése és a gép visszavált a 0-ás állapotba. Egyébként visszavált a 0-ás állapotba.
- Amennyiben az első szalagon üresre érkezünk, elértük a Cíatl végét, a gép leáll.

$M = \langle \{0, 1, 2, \text{igen}, \text{nem}\}, \{\text{red}, \text{yellow}, \text{black}, \text{green}, \text{blue}\}, \{\text{red}, \text{yellow}, \text{black}, \text{green}, \text{blue}, 1, \text{ü}\}, \delta, 0, \text{igen}, \text{nem} \rangle$				
$\delta$	$\bullet, \text{ü}$	$\square (\text{red}/\text{yellow}/\text{black}/\text{green}), \text{ü}$	$\text{ü}, \text{ü}$	továbbiak
0	1 $\bullet \rightarrow$ $\text{ü} -$			
1	2 $\bullet \rightarrow$ $\text{ü} -$	0 $\square \rightarrow$ $\text{ü} -$	igen	nem
2	0 $\bullet \rightarrow$ 1 $\rightarrow$			



3.19. ábra. Fekete trió Prófeciákártya működésének ábrázolása Turing géppel

Az implementációban minden szint teljesítése után megvizsgáljuk, hogy elérteük-e az utolsó (2.) szintet. Amennyiben igen, visszatérünk a legmagasabb pontszámmal. A többi Cíattlelem már nem befolyásolhatja ezt.

```

1 $state = 0;
2 $level = 0;
3 foreach ($coatl->pieces as $piece) {
4     switch ($state) {
5         case 0:
6             if ($piece->color === $this->color) $state = 1;
7             break;
8         case 1:
9             if ($piece->color === $this->color) $state = 2;
10            else $state = 0;
11            break;
12        case 2:
13            if ($piece->color === $this->color) {
14                $level++;
15                if ($level === 2) return array("score" => 7, "level" => 2);
16            }
17            $state = 0;
18            break;
19        }
20    }

```

```
21 if ($level == 0) return array("score" => 0, "level" => 0);
22 else return array("score" => 3, "level" => 1);
```

### 3.3. forráskód. BlackTrio.class.php részlet

#### 3.9.6. Kék páros Próféciakártyák

A kék páros Próféciakártyák 2 elem hosszúságú mintát várnak el. Ezek az elemek minden különböző színűk, így megkülönböztetünk egy kártya által meghatározott első és egy második színt (előfordulásuk sorrendjében). Az elemeknek egymás mellett folytatónak kell szerepelniük a Cíatlban. A táblázat három szintet különböztet meg. Ha egy szint sem teljesül akkor nem jár pont. Ha több mint három szint teljesül, akkor is a 3. szintért járó pontszám kerül elszámolásra.

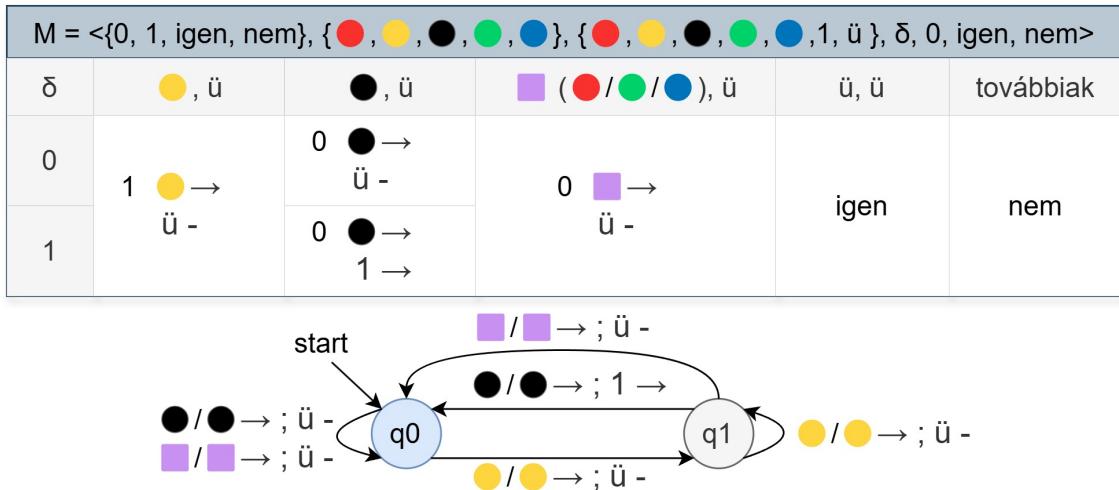


3.20. ábra. Példa kék páros Próféciakártyára [5]

A Turing gép négy állapottal dolgozik: 0 (kezdő állapot), 1, igen, nem.

- Az első szalag felett lévő fej minden lépés után jobbra mozog.
- A második szalagot szintek teljesítésének a könyvelésre használjuk. Egy szint könyveléskor egy 1-es számot ír és jobbra mozog. Egyébként nem mozog és nem ír.
- A 0-ás állapotban: Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép átvált az 1-es állapotba. Egyébként marad a 0-ás állapotban.
- Az 1-es állapotban: Ha az első szalagon a kártya által meghatározott második szín jelenik meg, elkönyvelődik egy szint teljesítése és a gép visszavált a 0-ás állapotba. Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép az 1-es állapotban marad. Egyébként visszavált a 0-ás állapotba.

- Amennyiben az első szalagon üresre érkezünk, elérünk a Cöatl végét, a gép leáll.



3.21. ábra. Kék páros Prófériakártya működésének ábrázolása Turing géppel

Az implementációban minden szint teljesítése után megvizsgáljuk, hogy elérünk-e az utolsó (3.) szintet. Amennyiben igen, visszatérünk a legmagasabb pontszámmal. A többi Cöattlelém már nem befolyásolhatja ezt.

```

1 $state = 0;
2 $level = 0;
3 foreach ($coatl->pieces as $piece) {
4     switch ($state) {
5         case 0:
6             if ($piece->color === $this->color_1) $state = 1;
7             break;
8         case 1:
9             if ($piece->color === $this->color_2) {
10                 $level++;
11                 if ($level === 3) return array("score" => 5, "level" => 3);
12                 $state = 0;
13             }
14             else if ($piece->color !== $this->color_1) $state = 0;
15             break;
16     }
17 }
18 if ($level == 0) return array("score" => 0, "level" => 0);
19 else if ($level == 1) return array("score" => 1, "level" => 1);
20 else if ($level == 2) return array("score" => 3, "level" => 2);
    
```

```
21 else return array("score" => 5, "level" => 3);
```

### 3.4. forráskód. BluePair.class.php részlet

#### 3.9.7. Kék négyes Prófériakártyák

A kék négyes Prófériakártyák 4 elem hosszúságú mintát várnak el. Ez a minta minden két színből áll, melyek sormintát alkotnak, így megkülönböztetünk egy a kártya által meghatározott első és egy második színt (előfordulásuk sorrendjében). Az elemeknek egymás mellett folytatónak kell szerepelniük a Cíatlban. A táblázat egy szintet határoz meg. Ha nem teljesül, akkor nem jár pont. Ha több mint egyszer teljesül, akkor is az 1. szintért járó pontszám kerül elszámolásra.



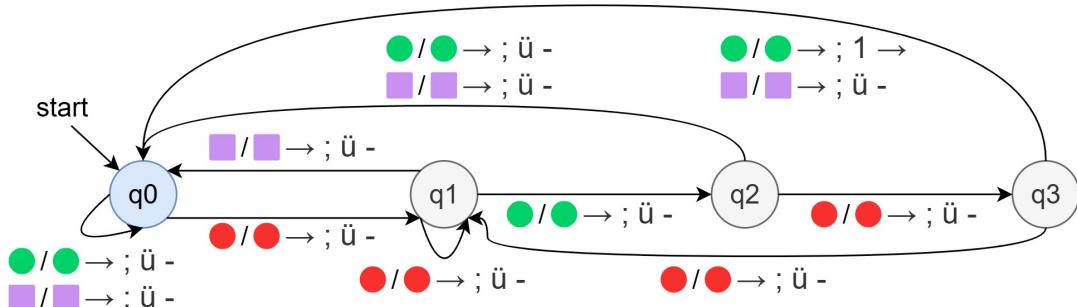
3.22. ábra. Példa kék négyes Prófériakártyára [5]

A Turing gép hat állapottal dolgozik: 0 (kezdő állapot), 1, 2, 3, igen, nem.

- Az első szalag felett lévő fej minden lépés után jobbra mozog.
- A második szalagot szintek teljesítésének a könyvelésre használjuk. Egy szint könyveléskor egy 1-es számot ír és jobbra mozog. Egyébként nem mozog és nem ír.
- A 0-ás állapotban: Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép átvált az 1-es állapotba. Egyébként marad a 0-ás állapotban.
- Az 1-es állapotban: Ha az első szalagon a kártya által meghatározott második szín jelenik meg, a gép átvált a 2-es állapotba. Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép az 1-es állapotban marad. Egyébként visszavált a 0-ás állapotba.

- Az 2-es állapotban: Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép átvált a 3-as állapotba. Egyébként visszavált a 0-ás állapotba.
- A 3-as állapotban: Ha az első szalagon a kártya által meghatározott második szín jelenik meg, elkönyvelődik egy szint teljesítése és a gép átvált a 0-ás állapotba. Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép átvált az 1-es állapotba. Egyébként visszavált a 0-ás állapotba.
- Amennyiben az első szalagon üresre érkezünk, elérünk a Cöatl végét, a gép leáll.

M = <{0, 1, 2, 3, igen, nem}, {●, ○, ●, ○, ●, ○, ●, ○}, {●, ○, ●, ○, ●, ○, 1, ü}, δ, 0, igen, nem>						
δ	●, ü	○, ü	■ (○/●/○), ü	ü, ü	továbbiak	
0	1 ● →	0 ○ → ü -				
1	ü -	2 ○ → ü -				
2	3 ● → ü -	0 ○ → ü -	0 ■ → ü -		igen	nem
3	1 ● → ü -	0 ○ → 1 →				



3.23. ábra. Kék négyes Prófériakártya működésének ábrázolása Turing géppel

Az implementációban amint teljesül az 1. szint, visszatérünk a hozzárendelt pontszámmal. A többi Cöatilelem már nem befolyásolhatja ezt.

```

1 $state = 0;
2 foreach ($coatl->pieces as $piece) {
3     switch ($state) {
4         case 0:
5             if ($piece->color === $this->color_1) $state = 1;

```

```

6      break;
7
8      case 1:
9          if ($piece->color === $this->color_2) $state = 2;
10         else if ($piece->color !== $this->color_1) $state = 0;
11         break;
12
13     case 2:
14         if ($piece->color === $this->color_1) $state = 3;
15         else $state = 0;
16         break;
17
18     case 3:
19         if ($piece->color === $this->color_2) return array("score" =>
20             5, "level" => 1);
21         else if ($piece->color === $this->color_1) $state = 1;
22         else $state = 0;
23         break;
24     }
25 }
26
27 return array("score" => 0, "level" => 0);

```

3.5. forráskód. BlueQuartet.class.php részlet

### 3.9.8. Sárga kettő-kettő Próbéciakártyák

A sárga kettő-kettő Próbéciakártyák 4 vagy 5 elem hosszúságú mintát várnak el. Ez a minta mindenkor két színből áll, melyek három-kettő, kettő-három vagy kettő-kettő arányban oszlanak el, így megkülönböztetünk egy a kártya által meghatározott első és egy második színt (előfordulásuk sorrendjében). Az elemeknek egymás mellett folytatónak kell szerepelniük a Cíatlban. A táblázat egy szintet határoz meg. Ha nem teljesül, akkor nem jár pont. Ha több mint egyszer teljesül, akkor is az 1. szintért járó pontszám kerül elszámolásra.



3.24. ábra. Példa sárga három-kettő Prófeciákártyára [5]

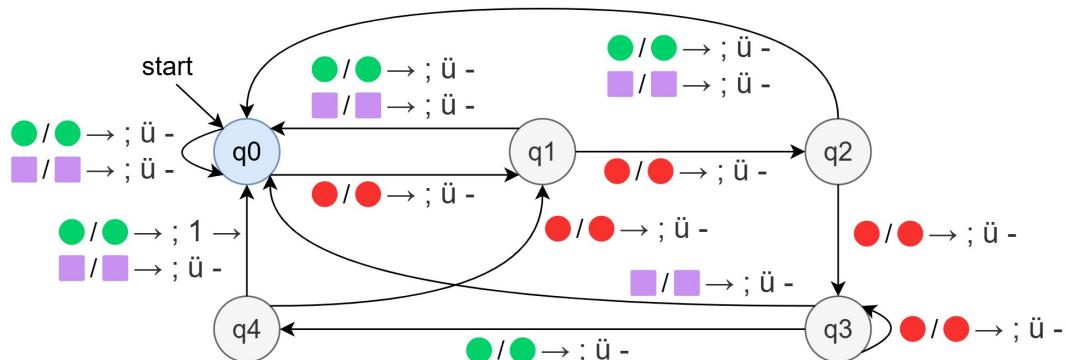
Az alábbi Turing gép a három-kettő arányú mintát képes felismerni.

A Turing gép hét állapottal dolgozik: 0 (kezdő állapot), 1, 2, 3, 4, igen, nem.

- Az első szalag felett lévő fej minden lépés után jobbra mozog.
- A második szalagot színtek teljesítésének a könyvelésre használjuk. Egy szint könyveléskor egy 1-es számot ír és jobbra mozog. Egyébként nem mozog és nem ír.
- A 0-ás állapotban: Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép átvált az 1-es állapotba. Egyébként marad a 0-ás állapotban.
- Az 1-es állapotban: Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép átvált a 2-es állapotba. Egyébként visszavált a 0-ás állapotba.
- A 2-es állapotban: Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép átvált a 3-as állapotba. Egyébként visszavált a 0-ás állapotba.
- A 3-as állapotban: Ha az első szalagon a kártya által meghatározott második szín jelenik meg, a gép átvált a 4-es állapotba. Ha az első szalagon a kártya által meghatártozott első szín jelenik meg, a gép a 3-as állapotban marad. Egyébként visszavált a 0-ás állapotba.
- A 4-as állapotban: Ha az első szalagon a kártya által meghatározott második szín jelenik meg, elkönyvelődik egy szint teljesítése és a gép átvált a 0-ás állapotba. Ha az első szalagon a kártya által meghatártozott első szín jelenik meg, a gép átvált az 1-es állapotba. Egyébként visszavált a 0-ás állapotba.

- Amennyiben az első szalagon üresre érkezünk, elérünk a Cíatl végét, a gép leáll.

$M = \langle \{0, 1, 2, 3, 4, \text{igen}, \text{nem}\}, \{\text{red circle}, \text{yellow circle}, \text{black circle}, \text{green circle}, \text{blue circle}\}, \{\text{red circle}, \text{yellow circle}, \text{black circle}, \text{green circle}, \text{blue circle}, 1, \text{ü}\}, \delta, 0, \text{igen}, \text{nem} \rangle$					
$\delta$	$\text{red circle}, \text{ü}$	$\text{green circle}, \text{ü}$	$\text{purple square} (\text{yellow circle} / \text{black circle} / \text{blue circle}), \text{ü}$	$\text{ü}, \text{ü}$	továbbiak
0	1 $\text{red circle} \rightarrow \text{ü} -$				
1	2 $\text{red circle} \rightarrow \text{ü} -$	0 $\text{green circle} \rightarrow \text{ü} -$			
2	3 $\text{red circle} \rightarrow \text{ü} -$		0 $\text{purple square} \rightarrow \text{ü} -$	igen	nem
3		4 $\text{green circle} \rightarrow \text{ü} -$			
4	1 $\text{red circle} \rightarrow \text{ü} -$	0 $\text{green circle} \rightarrow \text{ü} -$	1 $\rightarrow$		



3.25. ábra. Sárga három-kettő Próbéciakártya működésének ábrázolása Turing géppel

A kettő-kettő és a kettő-három arányú mintákat felismerő Turing gépek működése hasonló, apró módosítások szükségesek hozzájuk.



(a) Sárga kettő-kettő      (b) Sárga kettő-három

3.26. ábra. Példák sárga kettő-kettő és kettő-három Próbéciakártyára [5]

$M = \langle \{0, 1, 3, 4, \text{igen}, \text{nem}\}, \{\text{red}, \text{yellow}, \text{black}, \text{green}, \text{blue}\}, \{\text{red}, \text{yellow}, \text{black}, \text{green}, \text{blue}, 1, \text{ü}\}, \delta, 0, \text{igen}, \text{nem} \rangle$					
$\delta$	$\text{blue}, \text{ü}$	$\text{yellow}, \text{ü}$	$\text{purple} (\text{red}/\text{black}/\text{green}), \text{ü}$	$\text{ü}, \text{ü}$	továbbiak
0	1 $\text{blue} \rightarrow \text{ü} -$	0 $\text{yellow} \rightarrow \text{ü} -$			
1	3 $\text{blue} \rightarrow \text{ü} -$	4 $\text{yellow} \rightarrow \text{ü} -$	0 $\text{purple} \rightarrow \text{ü} -$	igen	nem
3					
4	1 $\text{blue} \rightarrow \text{ü} -$	0 $\text{yellow} \rightarrow 1 \rightarrow$			

$M = \langle \{0, 1, 2, 3, 4, \text{igen}, \text{nem}\}, \{\text{red}, \text{yellow}, \text{black}, \text{green}, \text{blue}\}, \{\text{red}, \text{yellow}, \text{black}, \text{green}, \text{blue}, 1, \text{ü}\}, \delta, 0, \text{igen}, \text{nem} \rangle$					
$\delta$	$\text{black}, \text{ü}$	$\text{green}, \text{ü}$	$\text{purple} (\text{red}/\text{yellow}/\text{blue}), \text{ü}$	$\text{ü}, \text{ü}$	továbbiak
0	1 $\text{black} \rightarrow \text{ü} -$	0 $\text{green} \rightarrow \text{ü} -$			
1	2 $\text{black} \rightarrow \text{ü} -$	3 $\text{green} \rightarrow \text{ü} -$	0 $\text{purple} \rightarrow \text{ü} -$	igen	nem
2					
3	1 $\text{black} \rightarrow \text{ü} -$	4 $\text{green} \rightarrow \text{ü} -$			
4		0 $\text{green} \rightarrow 1 \rightarrow$			

3.27. ábra. Sárga kettő-kettő és kettő-három Prófériakártyák működésének ábrázolása Turing géppel

Az implementációban amint teljesül az 1. szint, visszatérünk a hozzárendelt pontszámmal. A többi Códát le nem befolyásolhatja ezt.

A program 3 egymáshoz hasonló Turing gép leírása helyett 1 modulárisat használ. A YellowTwoTwo osztály rendelkezik egy ?bool típusú közép változóval. Ha a közép null értékű akkor 2-2 arányú, ha igaz értékű akkor 3-2 arányú, ha pedig hamis értékű akkor 2-3 arányú a kártya.

```

1 $state = 0;
2 foreach ($coat1->pieces as $piece) {
3     switch ($state) {
4         case 0:
5             if ($piece->color === $this->color_1) $state = 1;
6             break;
7         case 1:
8             if ($piece->color === $this->color_1) {

```

```

9      if ($this->middle !== null) $state = 2;
10     else $state = 3;
11   }
12   else $state = 0;
13   break;
14 case 2:
15   if (($this->middle && $piece->color === $this->color_1) || (!
16       $this->middle && $piece->color === $this->color_2)) $state =
17       3;
18   else if ($piece->color !== $this->color_1) $state = 0;
19   break;
20 case 3:
21   if ($piece->color === $this->color_2) $state = 4;
22   else if ($piece->color === $this->color_1) {
23     if ($this->middle !== null && !$this->middle) $state = 1;
24   }
25   else $state = 0;
26   break;
27 case 4:
28   if ($piece->color === $this->color_2) {
29     if ($this->middle !== null) return array("score" => 6, "level"
30         " => 1);
31     else return array("score" => 4, "level" => 1);
32   }
33   else if ($piece->color === $this->color_1) $state = 1;
34   else $state = 0;
35   break;
36 }
37 return array("score" => 0, "level" => 0);

```

### 3.6. forráskód. YellowTwoTwo.class.php részlet

#### 3.9.9. Sárga szivárvány Próbéciakártyák

A sárga szivárvány Próbéciakártyák 5 elem hosszúságú mintát várnak el. Ez a minta minden öt különböző színből áll, így megkülönböztetünk egy a kártya által meghatározott első, egy második, egy harmadik, egy negyedik és egy ötödik színt (előfordulásuk sorrendjében). Az elemeknek egymás mellett folytatónak kell szerepelniük a Cíatlban. A táblázat egy szintet határoz meg. Ha nem teljesül, akkor

nem jár pont. Ha több mint egyszer teljesül, akkor is az 1. szintért járó pontszám kerül elszámolásra.

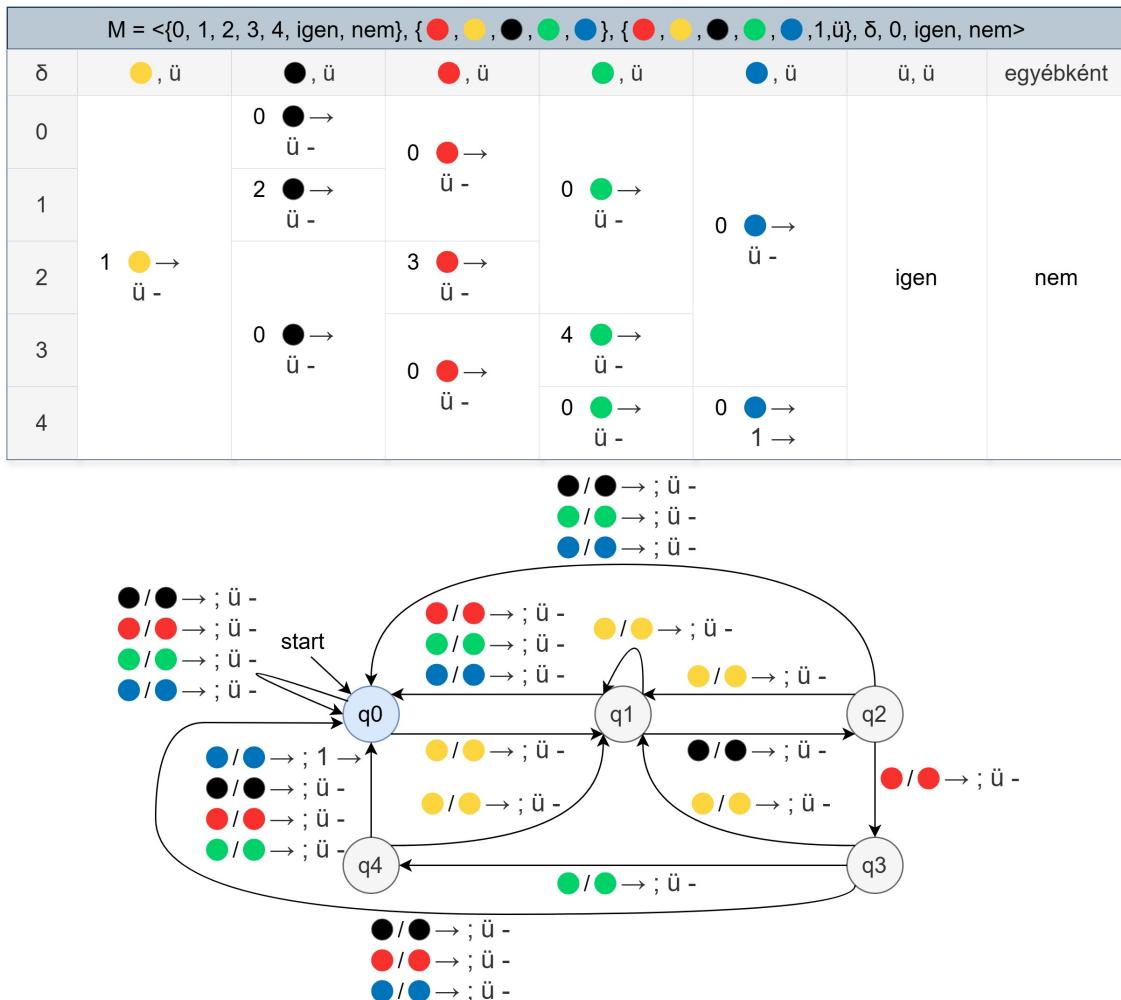


3.28. ábra. Példa sárga szivárvány Prófeciákártyára [5]

A Turing gép hét állapottal dolgozik: 0 (kezdő állapot), 1, 2, 3, 4, igen, nem.

- Az első szalag felett lévő fej minden lépés után jobbra mozog.
- A második szalagot színtek teljesítésének a könyvelésre használjuk. Egy szint könyveléskor egy 1-es számot ír és jobbra mozog. Egyébként nem mozog és nem ír.
- A 0-ás állapotban: Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép átvált az 1-es állapotba. Egyébként marad a 0-ás állapotban.
- Az 1-es állapotban: Ha az első szalagon a kártya által meghatározott második szín jelenik meg, a gép átvált a 2-es állapotba. Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép marad az 1-es állapotba. Egyébként visszavált a 0-ás állapotba.
- A 2-es állapotban: Ha az első szalagon a kártya által meghatározott harmadik szín jelenik meg, a gép átvált a 3-as állapotba. Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép visszavált az 1-es állapotba. Egyébként visszavált a 0-ás állapotba.
- A 3-as állapotban: Ha az első szalagon a kártya által meghatározott negyedik szín jelenik meg, a gép átvált a 4-es állapotba. Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép visszavált az 1-es állapotba. Egyébként visszavált a 0-ás állapotba.

- A 4-as állapotban: Ha az első szalagon a kártya által meghatározott ötödik szín jelenik meg, elkönyvelődik egy szint teljesítése és a gép átvált a 0-ás állapotba. Ha az első szalagon a kártya által meghatártozott első szín jelenik meg, a gép visszavált az 1-es állapotba. Egyébként visszavált a 0-ás állapotba.
  - Amennyiben az első szalagon üresre érkezünk, elérünk a Cíatl végét, a gép leáll.



3.29. ábra. Sárga szivárvány Prófériakártya működésének ábrázolása Turing géppel

Az implementációban amint teljesül az 1. szint, visszatérünk a hozzárendelt pontszámmal. A többi Cíatlelem már nem befolyásolhatja ezt.

```
1 $state = 0;
2 foreach ($coatl->pieces as $piece) {
3     switch ($state) {
4         case 0:
5             if ($piece->color === $this->color_1) $state = 1;
```

```

6         break;
7
8     case 1:
9         if ($piece->color === $this->color_2) $state = 2;
10        else if ($piece->color !== $this->color_1) $state = 1;
11        break;
12
13    case 2:
14        if ($piece->color === $this->color_3) $state = 3;
15        else if ($piece->color === $this->color_1) $state = 1;
16        else $state = 0;
17        break;
18
19    case 3:
20        if ($piece->color === $this->color_4) $state = 4;
21        else if ($piece->color === $this->color_1) $state = 1;
22        else $state = 0;
23        break;
24
25    case 4:
26        if ($piece->color === $this->color_5) return array("score"
27            " => 7, "level" => 1);
28        else if ($piece->color === $this->color_1) $state = 1;
29        else $state = 0;
30        break;
31
32    }
33
34 }
35
36 return array("score" => 0, "level" => 0);

```

### 3.7. forráskód. YellowRainbow.class.php részlet

#### 3.9.10. Zöld színes végtelen Próféciakártyák

A zöld színes végtelen Próféciakártyák 3+ elem hosszúságú mintát várnak el. Ez a minta minden három színből áll, két szín közrefog egy harmadik színt, viszont a harmadik szín tetszőleges (de legalább 1) elemszámmal szerepelhet, így megkülönböztetünk egy a kártya által meghatározott első, egy végtelen és egy második színt (előfordulásuk sorrendjében). Az elemeknek egymás mellett folytatónak kell szerepelniük a Cíatlban. A táblázat kettő szintet különböztet meg. Ha egy szint sem teljesül akkor nem jár pont. Ha több mint kettő szint teljesül, akkor is a 2. szintért járó pontszám kerül elszámolásra.

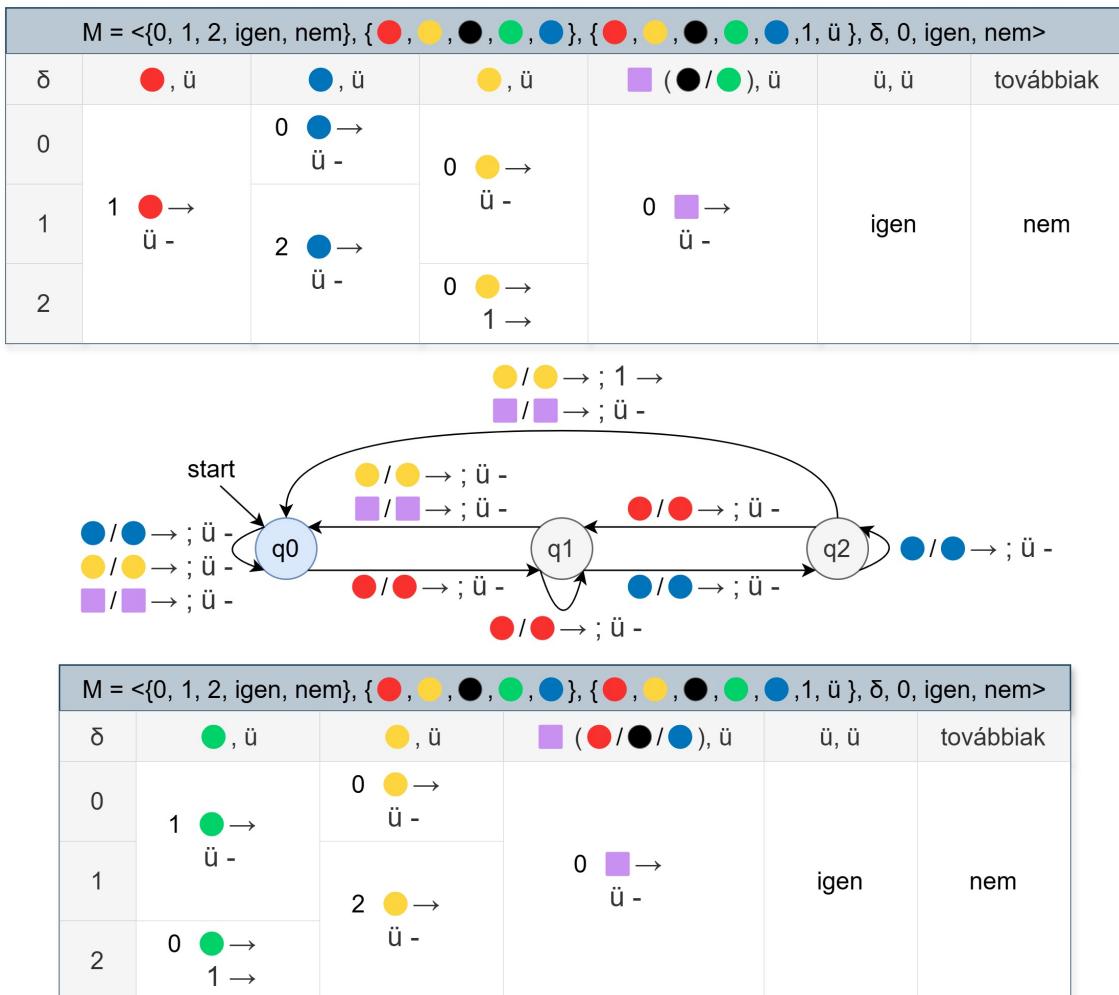


3.30. ábra. Példa zöld színes végtelen Próféciakártyára [5]

A Turing gép öt állapottal dolgozik: 0 (kezdő állapot), 1, 2, igen, nem.

- Az első szalag felett lévő fej minden lépés után jobbra mozog.
- A második szalagot színtek teljesítésének a könyvelésre használjuk. Egy szint könyveléskor egy 1-es számot ír és jobbra mozog. Egyébként nem mozog és nem ír.
- A 0-ás állapotban: Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép átvált az 1-es állapotba. Egyébként marad a 0-ás állapotban.
- Az 1-es állapotban: Ha az első szalagon a kártya által meghatározott végtelen szín jelenik meg, a gép átvált a 2-es állapotba. Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép az 1-es állapotban marad. Egyébként visszavált a 0-ás állapotba.
- A 2-es állapotban: Ha az első szalagon a kártya által meghatározott második szín jelenik meg, elkönyvelődik egy szint teljesítése és a gép átvált a 0-ás állapotba. Ha az első szalagon a kártya által meghatározott végtelen szín jelenik meg, a gép a 2-es állapotban marad. Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép visszavált az 1-es állapotba. Egyébként visszavált a 0-ás állapotba.
- Amennyiben az első szalagon üresre érkezünk, elérünk a Cíatl végét, a gép leáll.

A játékban előfordulnak olyan zöld színes végtelen Próféciakártyák melyeken a kezdő és befejező szín megegyezik. Ezen kártyák Turing gépének működése hasonló.



3.31. ábra. Zöld színes végtelen Próféciakártya működésének ábrázolása Turing géppel

Az implementációban minden szint teljesítése után megvizsgáljuk, hogy elérteük-e az utolsó (2.) szintet. Amennyiben igen, visszatérünk a legmagasabb pontszámmal. A többi Cójáték már nem befolyásolhatja ezt. Az azonos kezdő és befejező színű kártyák nem kaptak külön leírást, ilyenkor a szín\_1 és szín\_2 változók értéke azonos, a kritikus 2-es állapotban korábban vizsgáljuk meg a szín\_2 értékét, így nem merül fel probléma.

```

1 $state = 0;
2 $level = 0;
3 foreach ($coatl->pieces as $piece) {
4     switch ($state) {
5         case 0:
6             if ($piece->color === $this->color_1) $state = 1;
7             break;
8         case 1:

```

```

9   if ($piece->color === $this->color_8) $state = 2;
10  else if ($piece->color !== $this->color_1) $state = 0;
11  break;
12 case 2:
13  if ($piece->color === $this->color_2) {
14    $level++;
15    if ($level === 2) return array("score" => 6, "level" => 2);
16    $state = 0;
17  }
18  else if ($piece->color === $this->color_1) $state = 1;
19  else if ($piece->color !== $this->color_8) $state = 0;
20  break;
21 }
22 }
23 // in the end return the score and level, depending on the level
24 if ($level == 0) return array("score" => 0, "level" => 0);
25 else return array("score" => 2, "level" => 1);

```

3.8. forráskód. GreenColorfulInfinity.class.php részlet

### 3.9.11. Zöld üres végtelen Próféciakártyák

A zöld üres végtelen Próféciakártyák 5+ elem hosszúságú mintát várnak el. Ez a minta minden legalább 2 színból áll, két színes pár közrefog egy bármilyen színekből álló szakaszt, és ezen szakasz tetszőleges (de legalább 1) elemszámmal szerepelhet, így megkülönböztetünk egy a kártya által meghatározott első és egy második színt (előfordulásuk sorrendjében). Végtelen színt nem határozunk meg, mivel az bármilyen lehet. Az elemeknek egymás mellett folytatónak kell szerepelniük a Cíatlban. A táblázat egy szintet határoz meg. Ha nem teljesül, akkor nem jár pont. Ha több mint egyszer teljesül, akkor is az 1. szintért járó pontszám kerül elszámolásra.



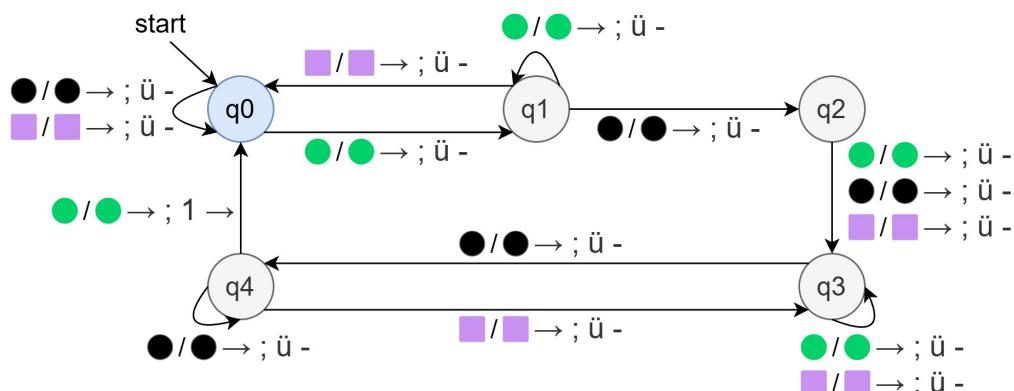
3.32. ábra. Példa zöld üres végtelen Prófériákártyára [5]

A Turing gép hét állapottal dolgozik: 0 (kezdő állapot), 1, 2, 3, 4, igen, nem.

- Az első szalag felett lévő fej minden lépés után jobbra mozog.
- A második szalagot színtek teljesítésének a könyvelésre használjuk. Egy szint könyveléskor egy 1-es számot ír és jobbra mozog. Egyébként nem mozog és nem ír.
- A 0-ás állapotban: Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép átvált az 1-es állapotba. Egyébként marad a 0-ás állapotban.
- Az 1-es állapotban: Ha az első szalagon a kártya által meghatározott második szín jelenik meg, a gép átvált a 2-es állapotba. Ha az első szalagon a kártya által meghatározott első szín jelenik meg, a gép az 1-es állapotban marad. Egyébként visszavált a 0-ás állapotba.
- A 2-es állapotban: Az első szalagon megjelenő bármilyen szín (de nem üres szimbólum) esetén a gép átvált a 3-as állapotba: ezzel biztosítjuk, hogy a végtelen szakasz legalább egy elemet tartalmazzon.
- A 3-as állapotban: Ha az első szalagon a kártya által meghatározott második szín jelenik meg, a gép átvált a 4-es állapotba. Egyébként a 3-as állapotban marad.
- A 4-as állapotban: Ha az első szalagon a kártya által meghatározott első szín jelenik meg, elkönyvelődik egy szint teljesítése és a gép átvált a 0-ás állapotba. Ha az első szalagon a kártya által meghatározott második szín jelenik meg, a gép a 4-es állapotban marad. Egyébként visszatvált a 3-as állapotba.

- Amennyiben az első szalagon üresre érkezünk, elérünk a Cöatl végét, a gép leáll.

$M = \langle \{0, 1, 2, 3, 4, \text{igen}, \text{nem}\}, \{\text{red circle}, \text{yellow circle}, \text{black circle}, \text{green circle}, \text{blue circle}\}, \{\text{red circle}, \text{yellow circle}, \text{black circle}, \text{green circle}, \text{blue circle}, 1, \text{ü}\}, \delta, 0, \text{igen}, \text{nem} \rangle$					
$\delta$	$\text{green circle}, \text{ü}$	$\text{black circle}, \text{ü}$	$\text{purple square} (\text{red circle} / \text{yellow circle} / \text{blue circle}), \text{ü}$	$\text{ü}, \text{ü}$	továbbiak
0	1 $\text{green circle} \rightarrow$ $\text{ü} -$	0 $\text{black circle} \rightarrow$ $\text{ü} -$			
1	1 $\text{green circle} \rightarrow$ $\text{ü} -$	2 $\text{black circle} \rightarrow$ $\text{ü} -$		0 $\text{purple square} \rightarrow$ $\text{ü} -$	
2	3 $\text{green circle} \rightarrow$ $\text{ü} -$	3 $\text{black circle} \rightarrow$ $\text{ü} -$			igen
3		4 $\text{black circle} \rightarrow$ $\text{ü} -$	3 $\text{purple square} \rightarrow$ $\text{ü} -$		nem
4	0 $\text{green circle} \rightarrow$ 1 $\rightarrow$				



3.33. ábra. Zöld üres végtelen Prófériakártya működésének ábrázolása Turing géppel

Az implementációban amint teljesül az 1. szint, visszatérünk a hozzárendelt pontszámmal. A többi Cöatilelem már nem befolyásolhatja ezt.

```

1 $state = 0;
2 foreach ($coatl->pieces as $piece) {
3     switch ($state) {
4         case 0:
5             if ($piece->color === $this->color_1) $state = 1;
6             break;
7         case 1:
8             if ($piece->color === $this->color_2) $state = 2;
9             else if ($piece->color !== $this->color_1) $state = 0;
10            break;

```

```

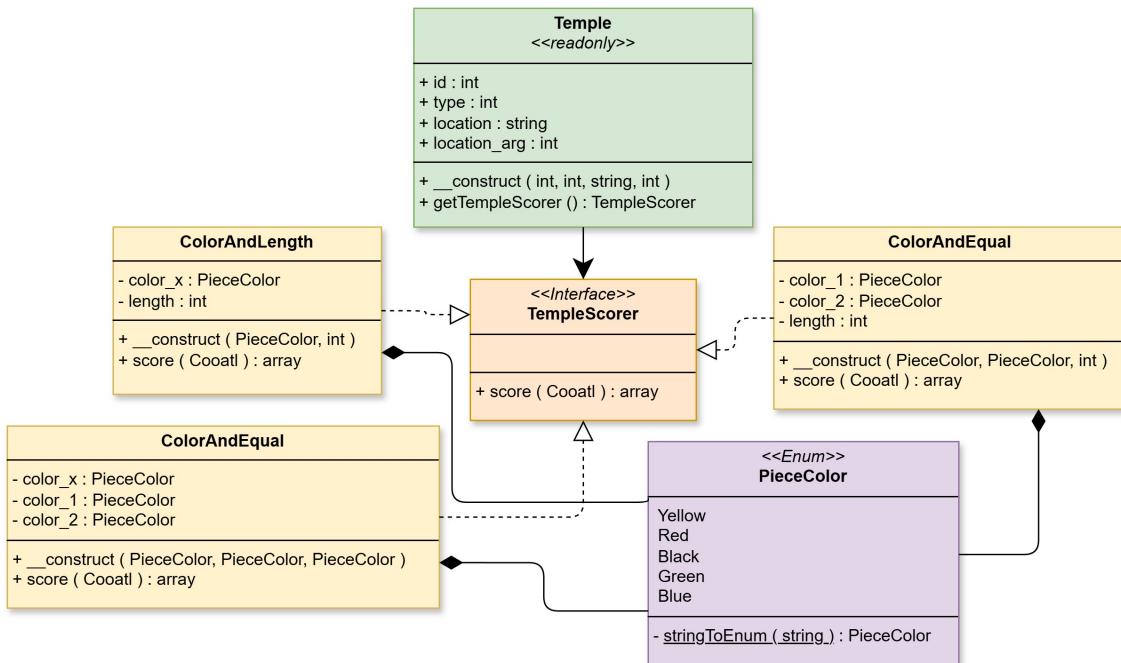
11   case 2:
12     $state = 3;
13     break;
14   case 3:
15     if ($piece->color === $this->color_2) $state = 4;
16     break;
17   case 4:
18     if ($piece->color === $this->color_1) return array("score" =>
19       4, "level" => 1);
20     else if ($piece->color !== $this->color_2) $state = 3;
21     break;
22   }
23 }
24 return array("score" => 0, "level" => 0);

```

3.9. forráskód. GreenBlankInfinity.class.php részlet

## 3.10. Templomkártyák

A csak olvasható Templomkártya objektumok rendelkeznek egy azonosítóval, egy típus, egy hely és egy hely\_paraméter tulajdonsággal.



3.34. ábra. Templomkártyák szerkezete UML diagrammal

A Templomkártya objektumok rendelkeznek egy getTempleScorer függvényel, mely egy Templomkártya pontozó interfész magvalósító osztály objektumával tér vissza, a Templomkártya típusától függően.

## 3.11. Templomkártya pontozók

A játék során a játékosok egy Templomkártyát helyezhetnek el a közös készletből vagy a kezüköből a befejezett (fejjel és farokkal rendelkező) Cójatljéik mellé, hogy győzelmi pontokat szerezzenek. minden Templomkártyán látható kettő követelmény, ezek közül legalább az egyiket teljesítenie kell a Cójatlnak, hogy lehelyezhető legyen mellé a Templomkártya. A követelmények fajtái:

- Hossz követelmény: A Cójatl pontosan annyi elemből álljon, mint a kártya által meghatározott szám.
- Szín tiltó követelmény: A Cójatlból ne szerepeljen a kártya által meghatározott szín.
- Szín egyenlőség követelmény: A Cójatlból pontosan annyi meghatározott színű elem legyen egy másik meghatározott színű.

### 3.11.1. Szerkezeti áttekintés

A programban a Templomkártyák pontozását erre kialakított objektumok végezik. A Templomkártyákat a Temple osztály ábrázolja, ezen osztály rendelkezik egy getTempleScorer() függvényel, amely visszatér a kártya típusának megfelelő pontozó objektummal. A Templomkártyákat három típusba soroljuk, aszerint, hogy a három követelmény szerint melyik kettő szerepel rajtuk, így három pontozó osztály implementálja a TempleScorer interfészét. A TempleScorer interfésznek egyetlen score függvénye egy Cooatl típusú objektumot vár bemenetként - ez az a Cójatl amit ki szeretnénk értékelni a kártyával - és egy ["pontszám" => egész szám, "szint" => egész szám] szerkezetű tömbbel tér vissza. A "pontszám" kulcsa tartalmazza a kártya kijátszásáért járó pontszámot, a "szint" kulcs pedig a telesített követelmények számát. A három pontozó osztály kiterjeszti a TempleScorer interfészét, saját változókat vesznek fel aszerint, hogy mennyi Cójatlelem szín, vagy szám jelenik meg a kártyákon.

### 3.11.2. Megvalósítás

Az implementálás során használtam a PHP tömbök megszámlálás függvényét, mivel ezzel  $O(1)$ -es időben lehet lekérdezni, esetünkben a Cöatl objektum Cöatilemek tömbjének hosszát. A tiltást és egyenlőséget vizsgáló kereső és megszámláló algoritmusok úgy lettek megírva, hogy egy Templomkártya kiértékeléséhez elegendő legyen egyszer megvizsgálni minden Cöatilemet egy Templomkártya kiértékeléséhez.

## 3.12. Felhasználói felület

A BGA keretrendszer által automatikusan létrehozott coatl\_coatl.tpl<sup>9</sup> fájlban került leírásra a játékfelület HTML váza, a Cöatilem készlettáblára mutatásakor lefutó eseménykezelő metódusok és javascript sémák.

A BGA keretrendszer által automatikusan létrehozott coatl.view.php<sup>10</sup> fájlban került leírásra a coatl\_coatl.tpl fájlban definiált blokkok generálása és a változók meghatározása, például a játékosok neveinek megjelenítése a játékokstából felett.

A BGA keretrendszer által automatikusan létrehozott coatl.js<sup>11</sup> fájlban kerültek leírásra a felhasználói felület játékelemekkel feltöltött függvények, az animációkért felelős metódusok, a játékmodell számára akcióüzenetet küldő metódusok és a játékmodelltől érkező értesítésekre reagáló metódusok.

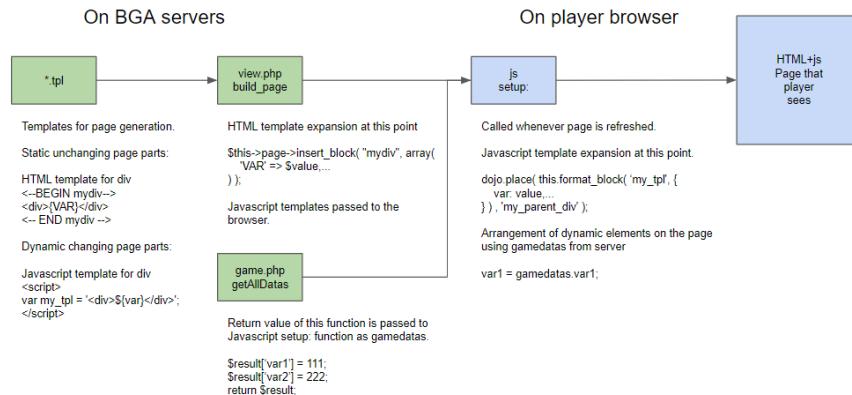
---

<sup>9</sup>[https://en.doc.boardgamearena.com/Game\\_layout:\\_view\\_and\\_template:\\_yourgamename\\_view.php\\_and\\_yourgamename\\_yourgamename.tpl](https://en.doc.boardgamearena.com/Game_layout:_view_and_template:_yourgamename_view.php_and_yourgamename_yourgamename.tpl) (elérés dátuma 2025. 04. 24.)

<sup>10</sup>9. lábjegyzék

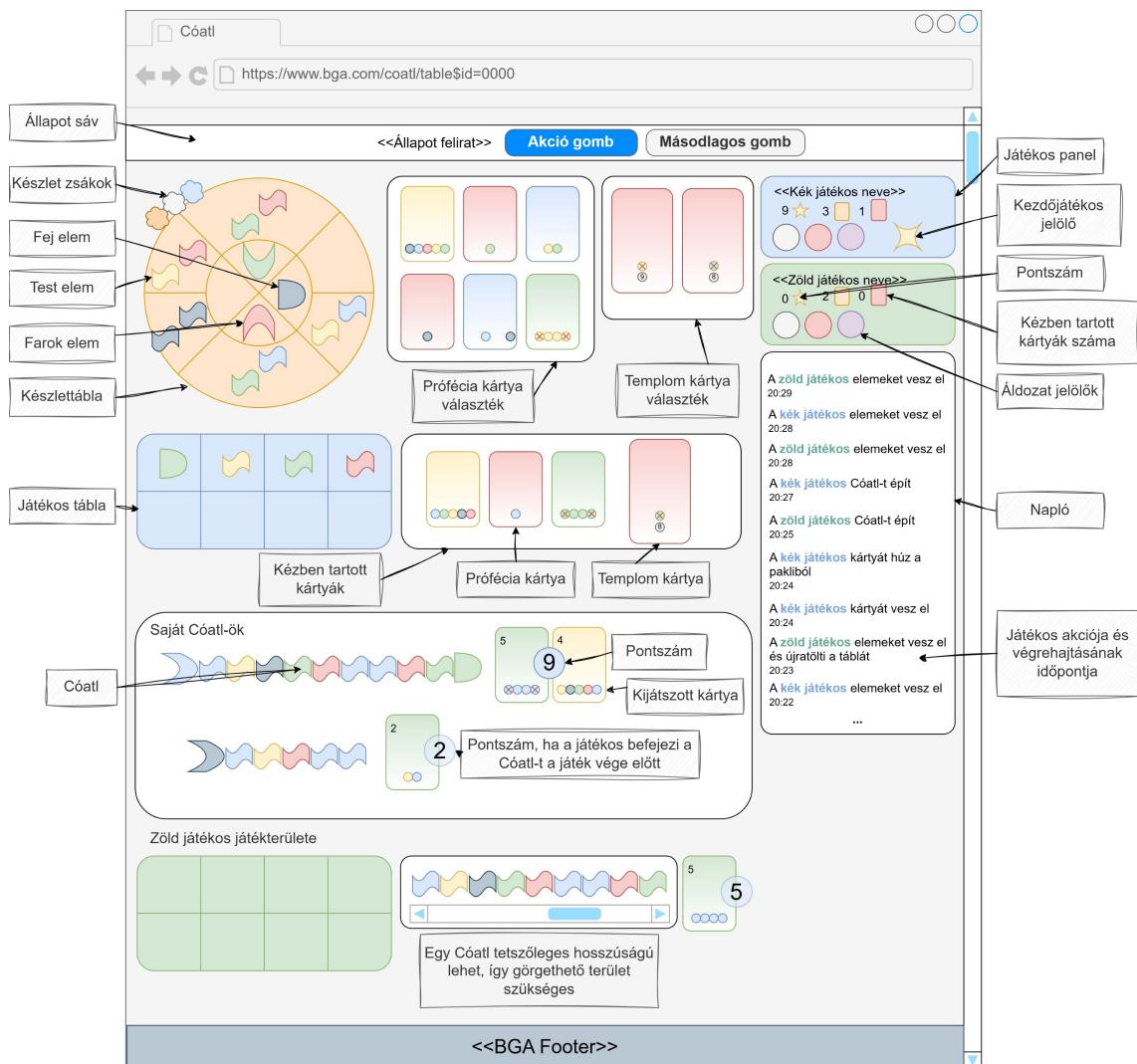
<sup>11</sup>[https://en.doc.boardgamearena.com/Game\\_interface\\_logic:\\_yourgamename.js](https://en.doc.boardgamearena.com/Game_interface_logic:_yourgamename.js) (elérés dátuma 2025. 04. 24.)

Generating BGA web pages from Templates



3.35. ábra. Felhasználói felület felépítésének folyamatábrája [7]

### 3.12.1. Drótvázterv



3.36. ábra. A felhasználói felület drótvázterve

### 3.13. Tesztelés

A program állandó felhasználói tesztelés alá van vetve a BGA játékosok által. A visszajelzései alapján javítások és fejlesztések kerülnek implementálásra. Új visszajelzés küldésére és a korábbiak megtekintésére a BGA külön felületet<sup>12</sup> biztosít. A Próbáciakártya pontozók teszteléséhez PHPUnit tesztosztályok kerültek definiálásra a misc/ mappában. Itt 1007 darab különböző bemenettel kerül tesztelésre a 9 eltérő módon működő Próbáciakártya pontozó osztály.



3.37. ábra. BGA - Cántor: Különlegesebb teszesetek gyakorlatban is megvalósítva [6]

<sup>12</sup><https://boardgamearena.com/bugs?game=2330> (elérés dátuma 2025. 04. 24.)

### 3.14. Verzió

A BGA-n minden játék oldalán elolvasható az adott játék programjának a verziószáma a játék információi között. A verziószám az összeállítás időpontja alapján kerül kialakításra ÉvHónapNap-ÓraPerc elrendezéssel.

Játékinformációk		
		
Tervező	Pascale Brassard, Etienne Dubois-Roy	
Illusztrátor	SillyJellie	
Kiadó	<a href="#">Synapses Games</a>	
Év	2020	
Fejlesztő	<a href="#">Doctor Neo Cortex</a>	
Lejátszott játékok száma	<b>737</b>	
Játékosok száma	<b>2 - 4</b>	
Játékhossz	<b>33 perc</b>	
Összetettség	 	2
Stratégia		3
Szerencse		1
Kapcsolat		1
BGA megjelenés	2025. április 11.	
Verzió	<u>250430-2025</u>	

3.38. ábra. BGA - Cóatl: Játékinformációk [6]

A szakdolgozat leadásakor a BGA - Cóatl legfrissebb verziója egy 2025. április 30-án 20 óra 25 perckor összeállított 250430-2025 számú szoftver. A forráskódja megegyezik az src/ mappában mellékelt kóddal, a grafikai fájlok kivételével.

## 4. fejezet

### Összegzés

A szakdolgozatom célja a Synapses Games által kiadott Cóatl című társasjáték teljes digitális változatának megvalósítása volt a Board Game Arena keretrendszerében. A dolgozat készítése során új tervezési mintákat, szoftvereket és technológiákat ismertem meg.

A program tervezéséhez hasznosítottam az eseményvezérelt alkalmazásokról, szoftver tervezésről és az adatbázis kezelésről szerzett szaktudásomat. A játékban szereplő Próféciakártyák pontozását Turing gépekkel modelleztem. A program megvalósításához egyszerre volt szükség kliensoldali és szerveroldali webprogramozásra, illetve termék- és webdízájn ismeretekre.

Eredményként sikerült elkészítenem eddigi legnagyobb projektemet, egy társasjáték digitális verzióját. A játék Béta verziója elérhető a Board Game Arena oldalán és a játékosok folyamatosan játszhatnak vele.

Az adaptáció pozitív fogadtatást kapott a játékosok felől, napról napra nő az aktív játékosok száma. A felhasználói igényeknek eleget téve igyekszem fejleszteni és karbantartani a kész szoftvert. A továbbiakban a Board Game Arena adminisztrátorai és a Synapses Games munkatársai döntenek arról, hogy mikor válik elérhetővé a játék minden játékos számára.

# Köszönetnyilvánítás

Szeretnék köszönetet mondani Grégory Isabelli, Emmanuel Colin, Pascale Brassard, Etienne Dubois-Roy, SillyJellie és Carl Briere számára, valamint a Board Game Arena és a Synapses Games minden munkatársának, akik nélkül ez a projekt elképzelhetetlen lett volna.

Köszönet dr. Abonyi-Tóth Andor egyetemi docensnek és témavezetőmnek, valamint valamennyi egyetemi oktatónak, akik segítettek megszerezni a projekt elkezítéséhez szükséges szaktudást.

Köszönet a szüleimnek Tóthné Juhász Andreának és Tóth Zoltán Zénónak, a testvéremnek Tóth Zalán Zénónak, a barátnőmnek Bakos Gabriellának, legjobb barátunknak Oláh Zsófiának és a nagyszüleimnek Banos Editnek, Juhászné Vida Margónak és Juhász Józsefnak, akik végig támogattak az egyetemi tanulmányaim és diplomamunkám elkészülése során.

Végül szeretném megemlíteni barátnőm kiscicáját, Bubit. Figyelemmel követte a játék elkészülését, kedves dorombolásával igyekezett a feszült pillanatokban is segítséget nyújtani. Bubi 2025. április 14-én nyugovóra tért.



# Forrásjegyzék

- [1] Board Game Arena. URL: <https://boardgamearena.com/> (elérés dátuma 2025. 04. 24.).
- [2] Board Game Arena és FryxGames. *BGA - A Mars Terraformálása*. URL: <https://boardgamearena.com/gamepanel?game=terraformingmars> (elérés dátuma 2025. 04. 24.).
- [3] Board Game Arena és Next Move. *BGA - Azul*. URL: <https://boardgamearena.com/gamepanel?game=azul> (elérés dátuma 2025. 04. 24.).
- [4] Board Game Arena és Roxley Games. *BGA - Santorini*. URL: <https://boardgamearena.com/gamepanel?game=santorini> (elérés dátuma 2025. 04. 24.).
- [5] Synapses Games. *Cóatl*. URL: <https://jeuxsynapsesgames.com/en/game/coatl/> (elérés dátuma 2025. 04. 24.).
- [6] Board Game Arena és Synapses Games. *BGA - Cóatl*. URL: <https://boardgamearena.com/gamepanel?game=coatl> (elérés dátuma 2025. 04. 24.).
- [7] Board Game Arena. *Board Game Arena Studio Doc - File:Bga-pages-from-templates.PNG*. URL: <https://en.doc.boardgamearena.com/File:Bga-pages-from-templates.PNG> (elérés dátuma 2025. 04. 24.).

# Ábrák jegyzéke

1.1.	BGA főoldal [1]	5
1.2.	BGA - A Mars Terraformálása: Játék oldal [2]	6
1.3.	BGA - Azul: Játék folyamatban [3]	6
1.4.	BGA - Santorini: Játék folyamatban [4]	7
1.5.	Cóatl társasjáték doboza [5]	8
1.6.	Cóatl társasjáték asztalképe [5]	8
2.1.	BGA - Cóatl: Játék oldal [6]	12
2.2.	BGA - Cóatl: Asztal beállítások [6]	12
2.3.	BGA - Cóatl: Játék kezdőállása [6]	13
2.4.	BGA - Cóatl: Többi játékos területe [6]	14
2.5.	BGA - Cóatl: Inaktív játékos állapotsávja [6]	14
2.6.	BGA - Cóatl: Aktív játékos állapotsávja [6]	15
2.7.	BGA - Cóatl: Próbéciakártya eldobása [6]	15
2.8.	BGA - Cóatl: Cójátélemek elvétele [6]	16
2.9.	BGA - Cóatl: Próbéciakártya elvétele a közös készletből [6]	16
2.10.	BGA - Cóatl: Próbéciakártya húzása a pakliból [6]	17
2.11.	BGA - Cóatl: Új Cójáték elkezdése [6]	17
2.12.	BGA - Cóatl: Cójátékok építése [6]	18
2.13.	BGA - Cóatl: Játékostársaink Cójátékok [6]	18
2.14.	BGA - Cóatl: Egy befejezett Cójáték [6]	18
2.15.	BGA - Cóatl: Játékospánelek az új pontszámmal [6]	19
2.16.	BGA - Cóatl: Végső forduló figyelmezhetősége [6]	20
2.17.	BGA - Cóatl: A játék végeredménye [6]	20
2.18.	BGA - Cóatl: A játék statisztikái [6]	21
3.1.	A játék indítása felhasználói eset diagrammal	23
3.2.	A játék kezdete felhasználói eset diagrammal	23

3.3.	Egy forduló felhasználói eset diagrammal . . . . .	24
3.4.	Egy játékos köre felhasználói eset diagrammal . . . . .	24
3.5.	Végső forduló felhasználói eset diagrammal . . . . .	25
3.6.	A program szerkezete UML diagrammal . . . . .	25
3.7.	Az program könyvtárának szerkezete . . . . .	26
3.8.	Az adatbázis szerkezete UML diagrammal . . . . .	27
3.9.	A játékmodell szerkezete UML diagrammal . . . . .	31
3.10.	A játék állapotai és átmenetei . . . . .	32
3.11.	Cóatlelemek szerkezete UML diagrammal . . . . .	34
3.12.	Cóatlök szerkezete UML diagrammal . . . . .	35
3.13.	Próbéciakártyák szerkezete UML diagrammal . . . . .	36
3.14.	Példa piros Próbéciakártyára [5] . . . . .	38
3.15.	Piros Próbéciakártyák működésének ábrázolása Turing géppel . . . . .	39
3.16.	Példa fekete X Próbéciakártyára [5] . . . . .	40
3.17.	Fekete X Próbéciakártya működésének ábrázolása Turing géppel . . . . .	41
3.18.	Példa fekete trió Próbéciakártyára [5] . . . . .	43
3.19.	Fekete trió Próbéciakártya működésének ábrázolása Turing géppel . . . . .	44
3.20.	Példa kék páros Próbéciakártyára [5] . . . . .	45
3.21.	Kék páros Próbéciakártya működésének ábrázolása Turing géppel . . . . .	46
3.22.	Példa kék négyes Próbéciakártyára [5] . . . . .	47
3.23.	Kék négyes Próbéciakártya működésének ábrázolása Turing géppel . . . . .	48
3.24.	Példa sárga három-kettő Próbéciakártyára [5] . . . . .	50
3.25.	Sárga három-kettő Próbéciakártya működésének ábrázolása Turing géppel . . . . .	51
3.26.	Példák sárga kettő-kettő és kettő-három Próbéciakártyára [5] . . . . .	51
3.27.	Sárga kettő-kettő és kettő-három Próbéciakártyák működésének ábrázolása Turing géppel . . . . .	52
3.28.	Példa sárga szivárvány Próbéciakártyára [5] . . . . .	54
3.29.	Sárga szivárvány Próbéciakártya működésének ábrázolása Turing géppel . . . . .	55
3.30.	Példa zöld színes végtelen Próbéciakártyára [5] . . . . .	57
3.31.	Zöld színes végtelen Próbéciakártya működésének ábrázolása Turing géppel . . . . .	58
3.32.	Példa zöld üres végtelen Próbéciakártyára [5] . . . . .	60

3.33. Zöld üres végtelen Prófériakártya működésének ábrázolása Turing géppel . . . . .	61
3.34. Templomkártyák szerkezete UML diagrammal . . . . .	62
3.35. Felhasználói felület felépítésének folyamatábrája [7] . . . . .	65
3.36. A felhasználói felület drótvázterve . . . . .	65
3.37. BGA - Cójáték: Különlegesebb tesztesetek gyakorlatban is megvalósítva [6] . . . . .	66
3.38. BGA - Cójáték: Játékinformációk [6] . . . . .	67

# Forráskódjegyzék

3.1.	Red.class.php részlet . . . . .	39
3.2.	BlackX.class.php részlet . . . . .	41
3.3.	BlackTrio.class.php részlet . . . . .	44
3.4.	BluePair.class.php részlet . . . . .	46
3.5.	BlueQuartet.class.php részlet . . . . .	48
3.6.	YellowTwoTwo.class.php részlet . . . . .	52
3.7.	YellowRainbow.class.php részlet . . . . .	55
3.8.	GreenColorfulInfinity.class.php részlet . . . . .	58
3.9.	GreenBlankInfinity.class.php részlet . . . . .	61