Ötletek a Nonogram alkalmazáshoz:

A képen áramkör, művészet, tervezés látható

Description automatically generated

avatar: rákattintva jön ki a menü

ahol a kurzor van, ott sötétebb az x és y tengely

segítségek

színek

Pistike

rajz

számok

pontszám/max

Kép címe a fejlécben

Valami ilyesmi lenne a játékablak.

és ilyenek a nonogramok (Wikimedia Commons):

A képen keresztrejtvény, minta, tér, öltés látható

Description automatically generated

Cím lehetne valami ilyesmi:

NoNo: Paint by Numbers

itt egy logo terv is:

A képen áramkör, művészet, tervezés látható

Description automatically generated

vagy egy másik:

A képen rajz, vázlat, Grafika, clipart látható

Description automatically generated

A képeket lehetne adatbázisban tárolni, ahogyan a felhasználói adatokat is. Mivel lehetnének színes nonogramok is, több információt is el kell tárolni bennük. Az adatbázisban kell legalább 3 tábla:

Az egyik a felhasználóknak (USER), ebben rögzíteni kellene a személyes és regisztrációs adatokon egy összpontszámot és azt is, hogy mely képeket oldotta már meg, esetleg melyik képet csinálja most. A USER táblában kellene rögzíteni azokat a képeket is, amelyeket már megoldott, vagy éppen megold. Talán úgy lenne a legegyszerűbb, ha a mentéskor, vagy képbezáráskor a kép pillanatnyi állását egy stringben betöltené a kép azonosítójával együtt. Esetleg egy külön táblában (SOLVEDIMAGE) tárolni ezeket a felhasználónévhez kötve (idegen kulcs). Ezen kívül ide kellene becsatolni a rendelkezésre álló segítségeket (HELPS) is, amiknek lehetne az idegen kulcsa a felhasználónév, azaz a USER kulcsa.

A segítségeken azt a kell érteni, hogy ha elakad a játékos egy kitöltésben, akkor kérhet segítséget. A segítségek különböző fajtájúak lehetnek: a pillanatnyi kitöltés ellenőrzése és max 2 hibás négyzet megmutatása, egy egész sor vagy oszlop megmutatása, meghatározott négyzet értékének felfedése, véletlenszerűen fed fel 3 mezőt, egy 3\*3-as területet fed fel, egy 5\*5-ös területet fed fel, stb. Ezeket egyrészt lehetne nyerni egy kép megoldásával, valamilyen súlyozott véletlenszerű módszerrel, másrészt a képmegoldásokért és a belépésekért járna valamennyi token, amiért „vásárolni” is lehetne ezeket. Ez még nincs teljesen kidolgozva, de az alapötlet valami ilyesmi lenne.

A képekről is kellene egy külön tábla (IMAGE), ez tartalmazná a képazonosítót, a kép címét, a méreteit a képnek, hogy színes vagy BW; ha tematizálni vagy kategorizálni akarjuk a képeket, akkor a kategóriát és egy rajzot vagy thumbnailt is, valamint egy meghatározott pontértéket is. Valamint itt kell tárolnunk a kép tartalmát is.

Talán a legegyszerűbb, ha kép egyes négyzeteihez karaktereket rendelünk, így a kép teljes tartalma rögzíthető egyetlen stringben. Az „N” jelölné a null értéket, vagyis az üres mezőt, a „T” a teli mezőt (ha egy stringben csak N és T van, akkor az B/W kép, ha más karakterek is vannak, akkor színes, pl „R” pirosat, „B” kéket, „Z” zöldet, „G” szürkét stb. Speciális karakterek nem kellenek bele, mert a szintén tárolt képméret alapján fel lehet építeni sorról-sorra, illetve ki lehet olvasni sorról-sorra.

Magához a képhez generálni kell meghatározásokat vagy clue-kat is, mert csak ez alapján oldható meg. Külön a sorokhoz, külön az oszlopokhoz. Itt van jelentősége igazán a színes vagy B/W képek különbségének, ugyanis a clue-k meghatározása más szabályok szerint történik a kettőben (két egymás melletti fekete mező folytonos, így két kitöltött mező között üresnek kell lennie; két egymás melletti színes mező, ha más színűek, akkor külön mezőnek számítanak, nem kell közéjük üres mező). Talán emiatt érdemes előbb csak B/W-re megcsinálni, és azután kiterjeszteni, ha már működik a dolog.

Csatolok ide egy ER diagrammot is:

A képen szöveg, rajz, diagram, vázlat látható

Description automatically generated

Azon még el lehet gondolkodni, hogy legyen-e üzenetküldési vagy chatelési lehetőség az applikcióban – de ez is egy további lépés lehetne inkább.