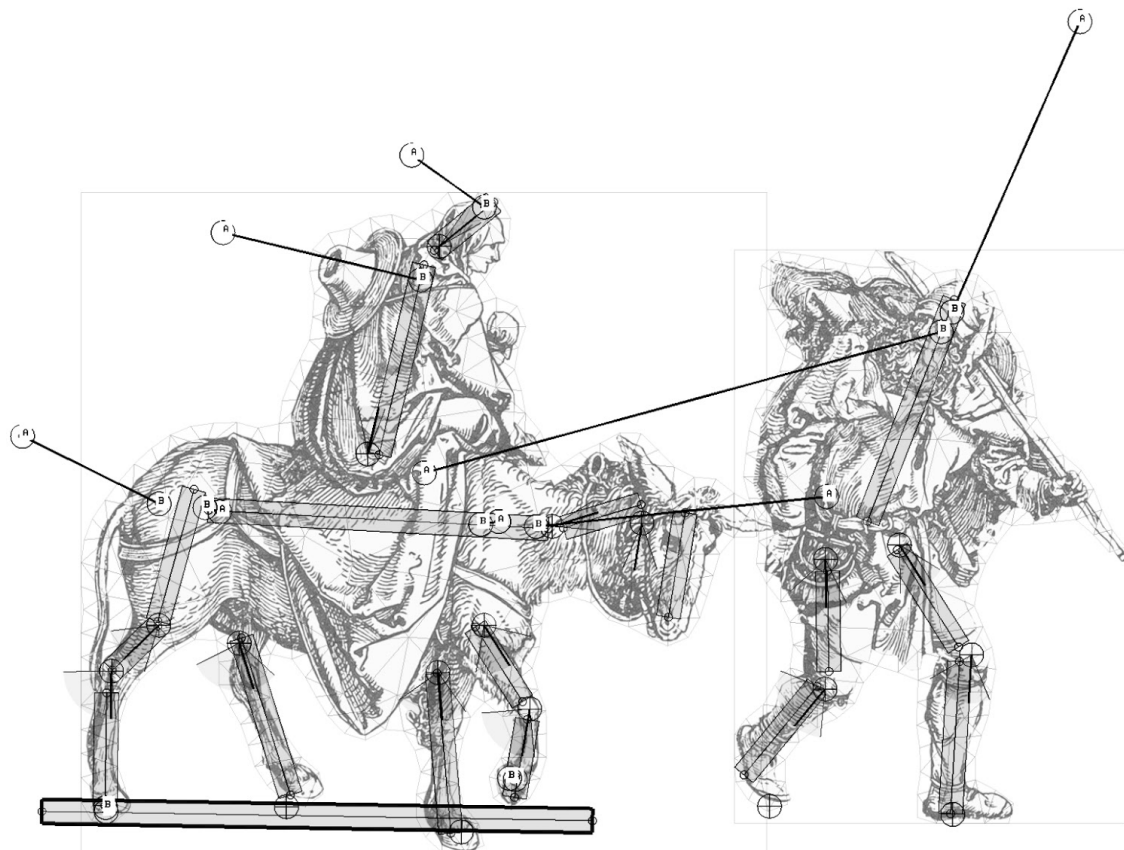


Átlátszó felületek

Kreatív digitális eszközök tervezése



Doktori értekezés / Moholy-Nagy Művészeti Egyetem Doktori Iskola

Szerző: Samu Bence

Témavezető: Szirtes János DLA habil. egyetemi tanár

Budapest, 2016

Tartalom

Bevezetés	3
Információs társadalom	5
Az információ fogyasztói társadalom	5
Digitális írástudás / eszközfüggőség	6
Fekete dobozok / digitális matrjoskák	7
Szabad szoftver	8
Szoftverművészet / kreatív kódolás / meta-eszköz	10
Kreatív digitális eszközök tervezése	12
Digitális szerszámok	13
Kreatív problémamegoldók	13
Belépési küszöb - Metafora	15
Lehetőség-tér	20
Transzparencia	22
Mestermunka	29
Egyszerűség	32
Transzparencia	33
Tág felhasználási terület	37
Tézisek	38
Kivonat	39
Bibliográfia	40
Kiállítások, művészeti megjelenések, installációk	42

Bevezetés

Kutatásom középpontjában a HCI (Human-Computer-Interaction) azaz Ember-Gép interakció áll. Ez a kutatás arra a felületre (interfész) fókuszál ami a felhasználó és a gép között található. Vizsgálja ezt az emberi oldalról és gép oldaláról egyaránt. Ez a terület érinti a számítástechnika tudomány a viselkedés tudomány, tanulás elmélet, design és a média elmélet területeit.

Értekezésem célközönsége, hasonlóan a mestermunkáméhoz, olyan emberek, akik nem rendelkeznek kiterjed ismerettel a digitális technológia területén, viszont igényük van kihasználni az általa felkínált lehetőségeket, megismerni a mögöttük húzódó tervezési módszereket. Érdeklődnek az interakció és az interfésztervezés technológiai és filozófiai háttere iránt. Célom, hogy közérthető formában mutassam be ennek egy igen gyorsan fejlődő, a művészetét jövőjét is meghatározó területnek a problémáit, kérdéseket tegyek fel és példákon keresztül meg is válaszoljam azokat. Mestermunkámmal is hasonló célokat tűztem ki magam elé. Egy olyan szoftver-eszközt hoztam létre, ami kiindulópontként szolgálhat a digitális írástudás elsajátításában. A jelenlegi állapot nem végső, sokkal inkább egy hosszú távú kutatás aktuális állapotának tekinthető. Az egyetemi éveim alatt fogalmazódott meg bennem az a gondolat, hogyha egyedi szoftverek készítésével válaszokat tudok adni saját kreatív problémáimra, ez a megközelítés más alkotó emberek számára is hasznos lehet. Művészeti és alkalmazott munkáim megvalósításához számos meta-eszközt, azaz eszköz készítő eszközt fejlesztettem ki, egy-egy probléma megoldására. Bizonyos esetekben arra volt szükség, hogy az adott feladat megoldását hatékonyabbá tegyem, máskor olyan nyitott rendszerek létrehozására volt szükség, ahol a végeredmény nem determinisztikus, a szoftver eszközeivel kreatív alkotói folyamat révén alakult ki. Ezeket a szoftvereket digitális eszközöknek nevezem, mivel hasonlóan a hagyományos értelemben vett eszközökhöz, általuk is problémákat lehet megoldani (itt első sorban egyszerű eszközökről beszélek, mint a rajz, vagy mintázási, formaalkotó eszközök, pl.: ceruza, kés, kalapács). Egy kreatív eszköz felhasználási területe tágas, nem előre definiált. Megfelelő kezekben teremteni képes, segítségével a gondolatok fizikai formát ölthetnek, médiuma lehet a kreativitásnak. A digitális eszközök célja is hasonló, itt a gondolatok a technológián keresztül válnak mások által láthatóvá, közös kommunikációs tereink objektumává. A technológia, közelmúltban bekövetkezett robbanásszerű fejlődése, lehetővé tette a művészek számára, hogy beleavatkozassanak ebbe a digitális elit által kisajátított területbe. A művészet és a technológia fúziója lehetővé tette, hogy ez az új médium sokkal emberközpontúbbá válhasson. A művészek következő generációjának nem csak felhasználnia kell a technológiát, hanem formálnia is azt. A programozás erre a legmegfelelőbb eszköz. Segítségével szinte bármilyen probléma leírható és meg is oldható. Az igazán innovatív

gondolatok megvalósítására nem létezik eszköz. Ezeket létre kell hozni. Oktatási tapasztalataim azt mutatják, hogy a programozás elsajátítása nem triviális feladat. Magas fokú matematikai, analitikus és absztrakt gondolkodást igényel, valamint nagy türelmet és alázatot egy végtelenül „buta” eszközzel a számítógéppel szemben. Az interdiszciplinaritás és a magas belépési küszöb miatt, sokan inkább csak felhasználói maradnak a technológiának. A megfelelő módon előállított digitális eszközök segíthetnek átlépni ezt az küszöböt, ugródeszkaként szolgálhatnak a mélyebb tudás elsajátításnak útján. Kutatásom ilyen digitális eszközökre és azok tervezésére fókuszál. Mestermunkám egy olyan, általam fejlesztett szoftver, amely a technológiában való kreatív gondolkodást segíti elő, valamint teret biztosít új fajta problémák feltérképezésére, megoldások kipróbálására.

Mivel a kreativitás maga egy igen nehezen mérhető jelenség, így a kreatív digitális eszközök tervezése nem egyszerű feladat. Az ilyen tervezés kiindulópontja a kreatív munkafolyamatok tanulmányozása lehet¹. Ilyenek a “brainstorming”, modellezés, vagy a vázlatkészítés. Az ilyen gyakorlatokra jellemző a szabad asszociálás, az ötletek gyors kipróbálása és iterálása, vagy eldobása. Ezekhez általában a legegyszerűbb eszközök a legmegfelelőbbek, mint a papír és a ceruza, építőkocka, agyag. Ezek olyan eszközök melyek elég általánosak ahhoz, hogy bármely kreatív problémát könnyen modellezni lehessen velük. A kreatív digitális eszközöknek is hasonló szerepet kell betölteniük.

Az információs társadalom polgárai egyre több “virtuális” tartalmat, szellemi terméket állítanak elő, olyan cél-eszközök segítségével, melyek kevés hangsúlyt fektetnek az egyediségre, a személyes problémákra, a tömegek alkotási vágyát uniformizált módon próbálják kielégíteni. Ezáltal a létrejövő szellemi termékek is sablonosak, hasonlóságot mutatnak. Ilyenek például a népszerű fotó alkalmazásokban használható, előre beállított hangulat-effektek, vagy az emoji-k (érzelmeket kifejező sárga fejecskék, ikonok) használata. Ezzel szemben a kreatív eszközök lehetőséget biztosítanak a felfedezésre, teret engednek a szabad asszociálásra, a próbálgatásra, az empirikus módon való tanulásra. Az ilyen szoftvereknek hasonló módon kell funkcionálniuk, mint a fent említett kézbe vehető primitív eszközöknek. Egy hatékony kreatív eszköz könnyen elsajátítható, intuitív módon működik és tágas teret biztosítani használójának. Jellemzői az alacsony belépési küszöb, azaz “kezdőknek” is könnyen használható, valamint a tág lehetőség-tér, hogy összetett problémák megoldására is alkalmas legyen. Értekezésemben ezeket a problémákat, gondolatokat és tervezési módszereket járom körül a digitális tartalomkészítés területén, majd a mestermunkámon keresztül bemutatom gyakorlati alkalmazásukat is.

¹ Mitchel Resnick: Design Principles for Tools to Support Creative Thinking (Tervezési minták, kreatív gondolkodást segítő eszközök készítéséhez) c. írása, MIT Media Lab (2005)

Információs társadalom

Az információ fogyasztói társadalom

Az információs társadalom² egy olyan társadalom, ahol az információ előállítása, terjesztése és alakítása a legfőbb mozgató motorja a gazdaságnak, társadalomnak és a politikának. Legfontosabb eleme az információs és kommunikációs technológia, ami az információ robbanáshoz vezetett melynek kihatása volt az élet minden területére (gazdaság, oktatás, egészségügy, hadiipar és közigazgatás...) Azokat az embereket, akik ilyen fajta társadalomban élnek digitális állampolgároknak nevezik. Ezekre az társadalmakra jellemző, hogy az anyagi értékek mellett egyre több szellemi terméket állítanak elő. Az információ növekedést egy kutatás³ számokkal is alátámasztotta. Míg 1986-ben a világ 2.6 exabyte⁴ adatot állított elő és tárolt, 2007-re ez 296 exabyte-ra emelkedett. Ez nem egy lineáris folyamat, az információ mennyiségének növekedése évente 25%-al emelkedik. Mivel a kommunikációs eszközök árának drámai csökkenése mára már olyan szintet ért el, hogy a fejlett társadalmakban szinte minden ember rendelkezik személyi számítógéppel (általában többel is), az információ előállítása a mindennapok részévé vált. Habár mára bárki lehet digitális szerző, a szellemi érték előállítása, valódi értékesség szempontjából megkérdőjelezhető. Mivel ez az értekezés elsősorban a kulturális azon belül is a művészeti, illetve kreatív értékek előállítására koncentrálnak, érdemes feltenni azt a kérdést, hogy a felhalmozott információ mennyisége arányos-e az általuk hordozott kulturális értékek mennyiségével? Egy amerikai felmérés kimutatta, hogy egy átlagos gyerek 21 éves korára 10.000 óra videó játékon van túl. Ez megközelítőleg az általános iskolában eltöltött teljes időnek a fele. Ha ennek az időnek csak egy részét kreatív módon tölthetnék, globális szinten ez hatalmas változásokat okozhatna⁵. Így jogos az a kérdés, hogy a digitális eszközökkel, milyen módokon lehetne kihasználni az ilyen globális kreatív energiákat?

² Wikipedia: Information society

³ Martin Hilbert: "The World's Technological Capacity to Store, Communicate, and Compute Information" címmel ír erről a Science Express blogon

⁴ 1 Exabyte = 10^{18} byte

⁵ Daphné Bavelier a játékokban rejlő lehetőséget kutatja, globális problémák megoldására: <http://cms.unige.ch/fapse/people/bavelier/>

Digitális írástudás / eszközfüggőség

Évezredekig az analfabetizmus természetes állapot volt, mivel csak a kiváltságosok rendelkeztek az írás-tudás képességével. Az információs társadalomban hasonló tendencia mutatkozik. Míg kezdetekben csak a mérnökök voltak képesek a gépekkel, és az azokon keresztül való kommunikációra, ez napjainkra teljesen megváltozott. Legalább is elsőre úgy tűnik! A számítógép alapvetően egy általános eszköz, eltérően a hagyományos értelemben vett eszközöktől, ennek nincs definiált célja. A felhasználó technológiai tudásán múlik, hogy mire és hogyan használja. A digitális írástudás készsége a szövegszerkesztéstől az álló és mozgókép készítésen át, az ezeket létrehozó szoftverek megalkotásáig terjedő végtelenül széles terület.

A számítógép korunk legkifinomultabb eszköze, habár maga a gép rendkívül bonyolult működését tekintve egyszerű. Bizonyos jelekre meghatározott válaszokat adnak és ezt következetesen mindig ugyanúgy teszik. A gépek alapvetően a digitális jeleket (igen / nem) értelmezik és ezek alapján utasításokat, logikai műveleteket hajtanak végre. A gépi nyelv ember számára alig, vagy egyáltalán nem értelmezhető bináris számsorozatokból áll. Ezeknek az eszközöknek a "lelke" a szoftverekben rejlik. A szoftver bármikor megújulhat, míg az azt hordozó hardver változatlan marad. A szoftver az emberi összetevője a gépeknek. Az első számítógép hardverek megalkotásával egy időben jelentek meg azok a számítógépes nyelvek melyek megpróbálták leküzdeni az ember gép kommunikáció nehézségeit. Ez kulcs fontosságú volt ahhoz, hogy az emberek összetettebb problémákat oldhassanak meg velük. Az első számítógépes nyelvekkel⁶ megalkothatták az első szoftvereket, melyeket mint eszközök használtak magasabb szintű problémák megoldásának építőköveiként. Ez a folyamat napjainkig is nyomon követhető, egyre szofisztikáltabb eszközök jelennek meg, melyek egyre több emberhez el tudnak jutni. A digitális írástudás a szoftverek által mindenkinek elérhetővé vált. A közösségi média, az információ megosztás, a mozgó, vagy állókép készítés állandó jelenléte, jó példa erre a jelenségre. A kreativitás és a művészet szemszögéből érdekes megvizsgálni, hogy mennyire befolyásolja a gondolat vagy a mű kialakulását az eszköz⁷, vagyis az azt létrehozó ember (vagy cég) szándéka, tervezési módszere, nyitottsága, képessége. Észre kell, hogy vegyük, hogy ellentétben a hagyományos írástudás képességével, a digitális polgárok számos tekintetben függenek az általuk használt eszközök készítőitől. Itt a függőség nem fizikai természetű, sokkal inkább a digitális technológia zártságában, kontrollálhatóságában keresendő. Az írástudás valójában a szoftvereket előállító "digitális elit" kezében maradt. Habár a szoftvergyártó cégek érdeke, hogy használóik igényeit

⁶ Assembly a gépi kódhoz legközelebb álló nyelv, ennek segítségével készítik a magasabb szintű nyelveket

⁷ Ezt a gondolatot a médium elmélet is körbejárja, de én elsősorban a technológiai vonatkozásait vizsgálom.

kielégítsék, ezeknek az eszközöknek a tervezésekor sokszor más stratégiák, anyagi érdekek is érvényesülnek. A folyamatos változás és elévülés meghatározó eleme ennek a folyamatnak.

Fekete dobozok / digitális matrjoskák

A fekete doboz műszaki értelemben olyan készüléket, vagy rendszert jelent, aminek a belső működése ismeretlen, rejtve van a kívülálló számára, viszont ki és bemenettel rendelkezik. Működésére csak úgy lehet következtetni, hogy megvizsgáljuk az ingerekre (bemenet) adott válaszreakcióit (kimenet). Ezt a módszert reverse-engineering⁸-nek is nevezik, aminek az a hátránya, hogy egy adott problémára számtalan megoldás létezik, és habár a vélt teória a ki és bemeneti szinten következetességet, egyezést mutat, az még nem biztos, hogy ténylegesen azonos a fekete doboz belső logikájával.

Egy szoftver architektúra is ilyen alapelemekből épül fel. Egy algoritmus tervezője a feladatokat analitikus módon közelíti meg. Először a problémát, kisebb, jobban értelmezhető al-problémára bontja, majd azokat ismét kisebb feladatokra. Ezt a folyamatot addig ismétli, míg egészen egyszerű, matematikailag megoldható, formalizálható problémákra talál. Ezt aztán becsomagolja egy "fekete dobozba", ezeknek a dobozoknak a ki és bemeneteit összeköti egymással, és az így létrejött nagyobb csoportot újabb dobozba teszi. Minden alapszintű, kis doboznak megvan a maga egyszerű feladata és általános problémákat oldanak meg. A nagyobb dobozok komplexebb, specializáltabb feladatokra valók, ezeket összeillesztve áll elő az algoritmus, ami szintén egy ilyen fekete doboznak tekinthető, feladata az adott probléma megoldása.

Ebben a technológiai társadalomban, folyamatosan fekete-dobozokkal vagyunk körülvéve. A fekete-dobozokra szükség is van, hiszen ez segít minket a tájékozódásban, abban, hogy magasabb, absztraktabb szinten kezelhessük a bennünk körülvevő digitális világot. Ha egy számítógép valós működését meg szeretnénk érteni, a szoftver-rendszereken túl, az elektronika, mikroelektronika és a kvantummechanika területein is kiterjedt tudással kéne rendelkezni. A Leonardók kora már elmúlt, nincs olyan ember, aki az egész technológiai rendszerünket egy személyben át tudná látni. Ezért kell bízunk a fekete-dobozainkba. Abba, hogy azok megfelelően működnek és pontosan azt csinálják (és csak azt) mint rájuk bízunk. Digitális társadalmunk ezekre a "digitális bizalmakra" épül. Elhisszük, hogy a felhőben biztonságban vannak a fényképeink, személyes adataink. Ezeket a bizalmakat építik

⁸ visszafejtés módszere, egy olyan mérnöki módszer, ami, már elkészült tárgyak, elektromos berendezések működését próbálja újra terv szintre hozni, hogy ezáltal az reprodukálható, vagy módosítható legyen.

azok a cégek is, akik szeretnék, hogy termékeikben megbízzunk, ragaszkodjunk hozzájuk, szeressük őket. Érthető az egyre fokozódó figyelem a személyiségi jogok és az adatbiztonság területén, hiszen a digitális rendszerek átszövik szellemi és anyagi életünk nagy részét. A fekete doboz információ architektúra szempontjából a hatékony építkezés fontos eszköze, de ha a rendszer nem transzparens, a hozzáférés korlátozott, veszélyes fegyverré is válhat.

Szabad szoftver

A szabad szoftver egy olyan mozgalom, ami alapjaiban meghatározta és mai napig is alakítja az információ és a szoftverek áramlásának, terjesztésének, tágabb értelemben véve a globális tudás alakulásának folyamatát. Mestermunkám tekintetében is meghatározó ez a jelenség, mivel az általam készített szoftverek is ezekre épülnek, részét képezik ennek a gondolkodásmódnak.

Az 70-es évekig, a számítástechnika kialakulásának kezdeténél a szoftverekre teljesen más módon tekintettek mint napjainkban. A kutatók és a hozzáértők ezeket a kódokat megosztották egymással, tanultak egymástól, míg a hardvert gyártó cégek örültek, hogy az emberek használhatóbbá, jobbá teszik az ő rendszereiket. Mivel a szoftverek általában egy általános nyelvben, pl. BASIC⁹-ben készültek, a szoftver tulajdonképpen az azt megalkotó kóddal volt azonos. A szoftver kódot számos módon terjesztették, mint például nyomtatott formában magazinokban, könyvekben (bestseller BASIC Computer Games), vagy akár a rádión keresztül dekódolható hang formájában. A kép a 70-es évek végén változott meg, amikor is ugrásszerűen megnőtt a szoftverfejlesztő cégek száma és erős verseny indult el a hatékonyabb, jobb szoftverek területén. A fejlesztők már nem akarták, hogy riválisaik elolvashassák, "ellophassák" szellemi terméküket. A 80-as évekre a szoftvereket már nem kód formájában terjesztették, hanem lefordított (bináris), vagyis ember által nem értelmezhető gépi formátumban. A szerzői jogokat kiterjesztették a szoftverekre is.

Ekkor, 1983-ban Richard Stallman, az MIT¹⁰ egyik mesterséges intelligencia kutatója bejelentette, hogy el akar készíteni egy teljesen ingyenes operációs rendszert, aminek a forrás kód is a része. Ezzel saját és más kutatók munkáját akarta megkönnyíteni, hogy függetlenedni tudjanak a

⁹ A BASIC általános célú programozási nyelv, melyet Kemény Tamás és Thomas Kurzt fejlesztett oktatási célokra.

¹⁰ Amerikai Egyesült államokban aktívan működő magánegyetem és kutatóintézet. Diákjainak nagy része posztgraduális képzésben vesz részt.

hardver és szoftvergyártó cégektől. A felvetés hamar követőkre talált és hamar a jogi keretrendszert is megalkották, ami a következő szabályokat tartalmazta:

- tetszőleges céllal használható
- működése szabadon tanulmányozható, szükség esetén az egyén céljaira szabható
- szabadon másolható, terjeszthető
- módosítás esetén szintúgy továbbterjeszthető, lehetővé téve, hogy egy egyedi változtatással együtt járó haszonból egy nagyobb közösség is részesedjen

A szoftver utolsó és egyben legfontosabb darabja, a “rendszer mag” Linus Torvalds nevéhez fűződik, amivel a rendszer teljessé válhatott. Ez vezetett a GNU/Linux megjelenéséhez, ami egy olyan operációs rendszer, ami teljes mértékben megfelel a fenti követelményeknek. A gyakorlatban a mai napra ezt az eredményezte, hogy a az internet alapját képező szerver hálózatok számítógépei közel száz százaléka erre a rendszerre épül. Ezeknek a rendszereknek a népszerűsége nem az ingyenességben rejlik, sokkal inkább abban, hogy olyan sok ember használja, fejleszti és javítja folyamatosan, hogy az stabilitás és biztonság szempontjából egyedülálló. Jól példázza a közösségi szoftverfejlesztés hatékonyságát és fejlődésének léptékeit az is, hogy a világhálóra kerülése¹¹ óta 13.500 fejlesztő járult hozzá a rendszerkód írásához, óránkénti 7-8 darab kiegészítéssel, javítással biztosítva, hogy a rendszer minden értelemben naprakész és biztonságos legyen. Ezzel a tempóval semelyik zárt szoftverfejlesztő cég nem vehette fel a versenyt, ezért sokan úgy döntöttek, hogy inkább együttműködve, beleadva saját szellemi értékeiket csatlakoznak a folyamathoz. A legnagyobb ilyen cégek közé tartozik a Google, Samsung, IBM és a Texas instruments is. Ez egy olyan forradalom, ami az információs társadalom jelenét és a jövőjét is meghatározza, teret biztosítva az információ biztonság és a globális szellemi értékek fejlődéséhez.

¹¹ Github rendszer, általában szoftverfejlesztők által használt, közösségi kód-megosztó platform.

Szoftverművészet / kreatív kódolás / meta-eszköz

A szoftverművészet egy olyan művészeti ág, ahol a számítógépes kód az önkifejezés első számú eszköze. Ennek a gyökere a személyi számítógépek elterjedésével egyidejű jelenség, a 70-es évektől, folyamatosan fejlődő műfaj. A 60-as években, a megfizethető videó technika megjelenése hasonlítható ehhez a folyamathoz. Az addig zárt formában létező, kontrollált és monopolizált mozgóképes nyelv egyszerre bárki számára elérhetővé vált. Ezek az eszközök művészek kezében egy egészen új és forradalmi kifejezőeszközzé formálódtak, saját vizuális nyelvet alkotva. A szoftverek területén, ehhez hasonló először a demoscene szubkultúrában jelent meg. Ezeknek a szubkultúráknak a tagjai, olyan fiatalok voltak, akik játék programokat “törtek fel” és terjesztettek. A kódot úgy módosították, hogy a szoftver indításakor az ő saját animációs “névjegyük” jelenjen meg, mint aláírás. Ebből a jelenségből indult ki a számítógépes animáció és maga a szoftverművészet hosszú fejlődési útja. Később a tudományos és művészeti egyetemek berkeibe is bekerült, fuzionálva a tudomány és a művészet területeit. Ennek úttörője az MIT (Massachusetts Institute of Technology), ahonnan számos olyan kezdeményezés indult el, ami a mai napig is meghatározza a szoftverművészet fejlődését. Az ehhez hasonló egyetemek, kutatólaborok, valamint a szabad szoftver mozgalom következménye, hogy a XX. század vége fele számos olyan digitális keretrendszer látott napvilágot, ahol a kód szabadon vizsgálható és közösségileg módosítható formában bárki számára ingyenesen elérhetővé vált. Ilyen például a Processing¹², vagy az OpenFrameworks¹³ is, ami tulajdonképpen egy közösség által létrehozott szoftver-eszköz-készlet, amely folyamatosan fejlődik, a közösség igényeire formálódik, egyfajta párbeszédben áll használóival. Itt nincs éles határvonal a fejlesztők és a fogyasztók között, a felhasználók kreatív fejlesztők, a fejlesztők művészek is egyben. Arra a alapgondolatra építenek, hogy az egyéni szellemi energia közösségi szinten sokkal nagyobb értéket hordoz. Az általuk megalkotott rendszer egy meta-eszköz, az eszköz készítő eszközök új generációja, előre nem látható problémák megoldására, vagy éppen felvetésére. Számos kiváló média alkotás létrejöttét tették lehetővé ezek a

¹² A Processing, John Maeda, Design By Numbers c. könyvéhez köthető, nyílt forráskódú programozási nyelv és integrált fejlesztői környezet (IDE). Ez a programozási nyelv a Java-n alapul, de egyszerűsített szintaxist és grafikus programozási modellt használt. <http://www.processing.org> Ez volt az első olyan programozói környezet, amely kifejezetten művészek, tervezők számára készült.

¹³ Az openFrameworks, kreatív kódolásra tervezett, nyílt forráskódú szoftver bázis. <http://www.openframeworks.cc> Célkitűzése hasonló a Processing-éhez, de C++ nyelvre épül, ami sokkal gép közelebb, így hatékonyabb szoftverek előállítására alkalmas.

rendszerek, mestermunkám is egy ilyen programozási keretrendszerre épül, fejlődését az Openframeworks körül létrejött közösség is befolyásolta. Mára már számos olyan oktatási intézmény létezik, amely kifejezetten a technológia és művészet területére fókuszál, ilyen többek között a New York-i ITP, ami röviden úgy definiálja magát mint “az új lehetőségek központja”, vagy a School for Poetic Computation, ami a digitális technológiára, mint önkifejező eszközökre tekint. Ez hozhatta létre azoknak a digitális művészeknek a generációját, akik folyamatosan kutatják és feszegetik a technológia határait, megkérdőjelezik és egyben formálják is azt.

Kreatív digitális eszközök tervezése

Digitális szerszámok

A szoftverekre általános értelemben tekinthetünk úgy, mint a jelen kor szerszámaira. Ezek olyan eszközök, melyek segítségével kisebb, nagyobb feladatokat oldhatunk meg. A problémamegoldás ősi, alapvető emberi képességünk. Az embert többek között az különbözteti meg a többi élőlénytől, hogy környezetét nem passzív módon, hanem aktív módon kezeli és alakítja, hogy az jobban megfeleljen saját igényeinek. Az ember a környezetébe kerülő tárgyokról belső képet alkot, ezeket tértől is időtől függetlenül egy mentális térbe helyezi. Ezek a reprezentációk csak arra várnak, hogy egy jövőbeli probléma felmerülésekor egymással összeállva, együtt alkalmazva megoldást nyújthassanak. Az ember alapvető tulajdonsága, hogy akár önkéntelenül is de reagál, alakítja az őt körülvevő hétköznapi világot.¹⁴ A mindennapi élet és az emberi praktikum számos példát szolgáltat ilyen a viselkedésekre.



(Jane Fulton Suri "Thoughtless acts" című könyvéből)

¹⁴ Jane Fulton Suri könyve a "Thoughtless acts" olyan jelenségek és emberi viselkedések gyűjteménye, ahol megfigyelhetjük, hogy az emberek tudat alatt milyen módon reagálnak az őket körülvevő világra. A könyv a következő témaköröket vizsgálja: reakció a térre, formákra, felmerülő lehetőségek kihasználása, kontextusok szabad kezelése.

A kreativitás legtöbb esetben a fizikai tárgy, fizikai manipulálásával, vagy más tárgyakkal való kombinálásával jelenik meg. A digitális szerszámok esetében ez a lehetőség nem adott, mivel ezek általában számítógépen, vagy telefonokon tárolt app-ok formájában léteznek. Ezek rendkívül hasznosak bizonyos cél feladatok megoldására. Tervezésüknél a legfontosabb szempont az adott probléma egyszerű és gyors megoldása. Ezek az eszközök általában egy dologban jók, de abban nagyon. Eltérően a nem digitális eszközöktől, a digitális eszközök fizikai test nélküli, virtuális tárgyak. A hardver fizikai megjelenése szoftver tekintetében nem releváns. Ergonómiai szempontból leválik róla. Ezt azt eredményezi, hogy a működését kizárólag az interfészen keresztül ismerhetjük meg. Legtöbb esetben a tervező egy nagyon "vékony" réteget kínál fel a külvilág számára, hogy az interakcióba kerülhessen vele. Emiatt, eltérően a fizikai testtel bíró szerszámoktól, ezek ritkábban használhatók fel kreatív módon más célra, mint ami a tervezők szándéka. Egy fizikai eszköz formai kialakítása magában hordozza használatának lehetőségeit, működése átlátható, más tárgyakkal kombinálható, vagy akár módosítható is. A digitális eszközöknél tudatosan kell megteremteni a kombinálhatóságot, azaz a szabad kísérletezés lehetőségét, hogy ezáltal az egyéni kreativitás megjelenhessen. Olyan tervezési módszereket kell alkalmazni, melyekkel a rendszer átláthatóvá válik, teret enged a szabad próbálgatásra, problémafelvetésre és válaszadásra is.

Kreatív problémamegoldók

Ahhoz, hogy kreatív embereknek készíthessünk eszközöket meg kéne vizsgálni a kreativitás működését. Erre a pszichológia és a pedagógia számos elméletet kínál. A témában való kutatásom alatt a konstruktivista tanuláselmélet tűnt jó kiindulópontnak, ami az emberek tanulási és problémamegoldási módszereit vizsgálja, ez alapján kategorizálja azokat. Az elmélet¹⁵ leegyszerűsítve azt állítja, hogy léteznek a rutin-probléma és kreatív-probléma megoldó emberek. Az elsődleges különbség a két típus között, hogy a kreatív-problémamegoldók képesek úgynevezett nem-rutin problémákat megoldani. Hogy tisztábban lássuk, hogy mi is ez a két típus meg kell néznünk, hogy mi a különbség a két módszer között. Egyszerű példával élve a rutin-probléma lehet egy szorzás: 42×98 . Ezt minden iskolát végzett ember meg tudja oldani, hiszen ismert a megoldáshoz vezető utat. A rutin-probléma fogalma persze nem azonos az egyszerű probléma fogalmával. Természetesen egy ilyen probléma is lehet nehéz, összetett, de a megoldás minden esetben a birtokunkban van. Sok számot összeszorozni nem könnyű, de rutin feladat. Ezzel szemben egy nem-rutin probléma, olyan probléma ahol a megoldás nem egyértelmű, nem látszik azonnal. Ilyen a például következő lehet:

¹⁵Jeremy Kilpatrick: Helping Children Learn Mathematics c. írása alapján

“Egy kerékpár boltban összesen 36 bicikli és tricikli található. Ezeknek összesen 80 db kereke van. Ebben az esetben hány bicikli és hány tricikli van a boltban? Erre az egyik lehetséges megoldás a következő: Abban biztosak lehetünk, hogy minimum két kereke van egy ilyen járműnek, tehát $36 \times 2 = 72$ kerék lenne, ha csak kerékpárok lennének, ezért a fennmaradó hiányzó kerekek biztos, hogy a tricikliké, tehát $(80-72)$ 8 tricikli és $(36 - 8)$ 28 bicikli található. Másik megoldás lehet a kísérletezés, azaz a próba szerencse módszere, ahol az ember addig próbálgatja a számokat, míg azok a helyes eredményt ki nem adják. A problémát akár matematikailag formalizált módon egy többváltozós egyenlettel is meg lehet oldani. A kreatív problémamegoldók nem csak, hogy több megoldással képesek előállni, hanem képesek kiválasztani az adott problémára a legoptimálisabbat. Az ilyen emberek mentális képet alkotnak a problémáról, hogy könnyebben felismerjék a megoldáshoz vezető utat. Különbség valakit megtanítani horgászni, vagy pedig engedni, hogy egy tó körül feltaláljon dolgokat zsinórok és kampók felhasználásával. Ezek az emberek általában jó autodidakta tanulók, vagyis olyanok akik jobban kedvelik a frontális oktatásnál azt, amikor saját maguk térképezhetik föl és kereshetik meg a megoldást egy adott problémára. Így nem csak egy adott utat találnak a megoldáshoz, hanem minden egyes megoldással a problémamegoldó képességüket gyarapítják. Általánosságban nézve a kreatív szakmában dolgozó jó szakemberek ilyenek. Mivel ők azok, akik nap mint nap problémamegoldásra kényszerülnek. Csak úgy lehetnek sikeresek, ha újabb és újabb megoldási stratégiák kifejlesztésére képesek. Eltérően más képességektől a problémamegoldást nehéz tanulni, tanítani, mivel nincs konkrét eset, nem lehet a létező, közösségi tudásra építeni. Minden embernek egyedi, saját útja van, a tanulás empirikus módon, tapasztalás útján történik. Az ilyen gondolkodású embereknek történő szoftver készítés több szinten is igazi kihívást jelent, ezért a következő témakörök megpróbálnak támpontot nyújthatnak a probléma megoldásában:

Belépési küszöb

Egy digitális eszköznek könnyen elsajátíthatónak kell lennie. Nem szabad, hogy olyan absztrakt fogalmakat használjon, melyek gátolják a felhasználót az értelmezésben. Meg kell találni azt az alapot, ami, mind az felhasználó korábbi tapasztalataiban, mind a szoftver működésében közös pont jelent. Ennek az egyik legelterjedtebb módszere a metaforák alkalmazása. Egy kalapácshoz, vagy ollóhoz nem kell használati útmutató, mert az objektum fizikai kialakítása hordozza az összes szükséges információt. A digitális eszközöknél nincs ilyen, itt egy olyan mentális összefüggésrendszert kell kialakítani, ami alapján a működés érthetővé válik. A metafora segít a virtuális felületeket értelmezésében, használatuk elsajátításában.

Metafora¹⁶

A metafora eredetileg egy görög szó, jelentése „keresztül visz”. A metafora lényege, hogy egy ismeretlen fogalmat bemutasson (illusztráljon) egy másik, fogalom felhasználásával, aminek eredetileg ahhoz semmi köze nincs. A metafora két független fogalmat kapcsol össze, ahol az összehasonlítás módszere nem meghatározott, a befogadóra van bízva. A metafora általában egy irányú, mint ebben a nyelvészek¹⁷ által sokat elemzett mondatban: “Ez a sebész egy hentes.”

megfeleltetés

egy dolog	->	egy másik dolog
forrás	->	cél
hentes	->	sebész
bárd	->	szike
hús	->	páciens
vágóhíd	->	műtő

¹⁶ Szabó Katalin “Metaforák és a felhasználói felület”, Dan Saffer “The Role of Metaphore in Interaction Design” című írása alapján

¹⁷ Kövecses Zoltán: Versengő metaforaelméletek

Metafora mint kognitív építőköve

A 70-es évek végén nyelvészek¹⁸ rámutattak arra, hogy a metafora nem csak nyelvi jelenség, a gondolkodás sajátossága, az, ahogy egy mentális fogalmat egy másik viszonylatában értelmezzük. A metaforák alapvető szerepet töltenek be az emberi gondolkodásban. Az egész gondolati rendszerünk metaforikus jellegű¹⁹. Metaforát használunk az érveinkhez, vagyis cselekedeteinket határozzák meg. Egy szó, vagy egy kép metaforikus használata az asszociációk komplex hálózatát hozza elő. Ez a komplex fogalomháló aztán kapcsolódási pontokat keres az adott kontextus fogalom terében. A metaforáknak fontos szerepük van az ismeretátadásban is, amikor egy új, ismeretlen fogalmakat szeretnénk bemutatni. (Ez általában igaz egy új interfész megalkotásánál) Ezt hatékonyan csak úgy lehet elérni, ha az új információk valamire épülni tudnak. Egy összetettebb tudást csak úgy lehet átadni, ha az oda vezető utat, egyszerűen átléphető metaforákkal tudjuk kikövezni, amin a befogadó kényelmesen tud haladni. (Érdemes megfigyelni, hogy semmilyen elvont fogalomról, úgy a metaforáról sem lehet beszélni, metaforák alkalmazása nélkül.)

Absztrakt gondolataink, mint tér és idő, ok-okozat is át vannak szöve metaforákkal: Az idő előre halad. Elhagyjuk a múltat. Valaki szerelembe esik, depresszióba zuhan. Ezek nem költői eszközök, hanem hétköznapi beszélgetések részei, melyek arra mutatnak rá, hogy mi emberek, hogyan is gondolkodunk. A metaforák átszövik mentális keretrendszerünket. Az absztrakt fogalmakban való gondolkodás elsődleges eszköze is ez, hiszen mi emberek fizikai testtel rendelkezünk és a valóság tapasztalatain keresztül szemléljük a világot. Emiatt az új gondolatokat, absztrakt fogalmakat is ebből a keretrendszerből származó fogalmakkal írjuk le.

A metafora segíthet különböző tudásháttérrel rendelkező emberek közötti kommunikációban. Digitális eszközök tervezésénél rendszeresen előfordul az, hogy két különböző szakma találkozik. A "Making Tea: Iterative Design Through Analogy" könyv, egy olyan példára épít, ahol egy kémiai kutatólabor szoftverfejlesztőket alkalmaz digitális rendszerük kifejlesztésére. Mivel a fejlesztőknek nem volt kémiai előképzettségük, "a kémia mint teakészítés" analógiát alkalmazták, és ezen keresztül tudtak elindulni a tervezésben.

A metafora inspirációs eszközként is használható. Majdnem minden új ötlet születése, meglévő gondolatok egymáshoz társításával jött létre²⁰. A tervezésben az ötletelés (brainstorming) sokszor a metafora eszköztétét használja. Az emberi elme nem viseli el az értelmi űrt. Ha két dolgot össze akarunk hasonlítani, szükségünk van arra, hogy megértsük az összehasonlítás módját. Egy ilyen

¹⁸ George Lakoff, Mark Johnson, Michael Reddy

¹⁹ George Lakoff: "The Contemporary Theory of Metaphor"

²⁰ Arthur Koestler, "The Act of Creation" c. könyvéből

gondolati összekapcsolás lehet egyszerű, mint pl. a mosoly és a virág. Itt az kapcsolat az, hogy mind a két fogalom kellemes és szép. Ezzel ellentétben, ha az összehasonlítás tárgyai távol esnek egymástól, a kapcsolat bizonytalan, az elménk az ismeretlent kitöltve akár téves asszociációkat alkalmazva nem megfelelően dekódolja a metaforát. Ezeket figyelembe véve az asszociációs képességünk hatékony eszköz lehet a jól megválasztott metaforák esetében.

A metaforák tájékozódásunk eszközei is lehetnek a digitális terekben. Ha egy metaforával keretet tudunk biztosítani egy gondolatnak, eszközt is kell találni a haladásra ebben a térben. A web-oldalakat “likek”, azaz hivatkozások kötik össze, az angolban “breadcrumbs” (kenyérmorzsák, Jancsi és Juliska meséből) segítségével könnyen megtudhatjuk, hogy éppen hol vagyunk az adott strukturában. A nyugati civilizációban, mint írásunkban is, a balról jobbra való haladás, a múlt és a jövő tengelye is egyben. A zene, vagy videó idejében is előre - hátra haladunk. Ezek majdnem minden szoftver interfészben megjelennek, ahol az idő ábrázolásra kerül. Ha a felhasználó nem képes orientálódni az általunk megalkotott térben, akkor az alapvetően használhatatlan²¹.

Metafora mint tervezői eszköz

A metafora, digitális felületeken való alkalmazásának első számú célja, hogy segítségével egy ismert fogalmat egy új, ismeretlenhez rendelhessünk. Ez lehetővé teszi, hogy a tanulási folyamatot ne kelljen megszakítani definíciók, magyarázó szövegek megjelenítésével. Ezt a tulajdonságot érdemes kihasználni, ha a szoftver még nem ismert működési módszereket használ. Amikor a metafora jól működik, hatalmas gondolati szakadékokat lehet vele átlépni. Alkalmazásával előnyöket lehet kiemelni, hátrányokat elfedni. Az emberek természetesen és gyakran alkalmazzák a metaforákat, ezért ha a tervező nem definiálja metaforáját megfelelően, a használó saját maga alkotja meg azokat. Ez nem feltétlenül egy szerencsés helyzet, hiszen abban az adott pillanatban még nincs birtokában a teljes tudás, így könnyen félreértelmezheti azt. Ez tekinthető a tervező felelősségének, hogy ellássa a felhasználót a megfelelő kapaszkodópontokkal. A asztal metafora²² talán a legjobban elterjedt megoldás a digitális felületeknél. Megjelenése a személyi számítástechnika megszületésével egyidejű. Vagy máshogy megfogalmazva, ez segítette közelebb hozni az átlag embereket a számítástechnikához. A gondolat alapja az, hogy tekinthetünk az operációs rendszerünkre úgy mint egy munkaasztalra. Ez egy olyan ismert fogalom, ami annak idején, minden potenciális

²¹ Dan Saffer: “The Role of Metaphore in Interaction Design”, “Using Metaphor to Orient” fejezetben.

²² Az asztal metafora a mai operációs rendszerekben leggyakrabban alkalmazott grafikus felhasználói felület. A koncepció a hagyományos munkaasztalt helyezi át digitális környezetbe, hogy ezáltal a kezdők számára is megfelelő kiindulási alapot biztosítson a számítógép használatához.

számítógép-vásárlónak a munkakörnyezete volt. Az asztalon elhelyezett írógép, mellette mappák, iratok, számlógép, vagy éppen egy szemetes. Ezen a képi és gondolati metaforán keresztül el lehetett érni, hogy bárki viszonylag rövid időn belül sikeres felhasználója legyen a rendszernek. Ezt a gondolatot WIMP paradigmának is nevezik, azaz "windows, icons, menus, pointer", vagyis "ablakok, ikonok, menük, mutató". Ennek a metaforának az hasznosságát mi sem bizonyítja jobban, mint az, hogy mai napig is erre épülő rendszereket használunk. Természetesen a metafora nem lehet a tervezés Szent Grálja, használatának veszélyei is vannak. Számos kritika²³ éri felhasználásukat, melyek csoportosítva a következők²⁴:

- Ha egy fizikai objektum egy absztrakt objektummal kerül összehasonlítása, az ember hajlamos a metaforát rosszul értelmezni, mert hisz abban, hogy az absztrakt objektummal is ugyanazokat a műveleteket el lehet végezni, mint fizikai társával. Ez félreértésekhez és helytelen használathoz vezethet. Ugyanez fordítva is igaz. Ha az összehasonlítás túl távoli, a két dolog nem, vagy nagyon kicsit fed egymást, az absztrakt tárgy lehetőségeit nem tudjuk kihasználni, mivel rosszul értelmezzük annak határait. Pl. a "dokumentum", vagy a "mappa" fogalmában nincs benne az, hogy lemásolhatja önmagát, egyszerre több helyen is létezhet.
- A metaforák nehezen skálázhatóak. Ha egy rendszer következetes és jól működteti metafora-rendszerét, nem biztos, hogy a jövőben, egy újabb verziója, újabb tulajdonságokkal beleilleszkedik ugyanabba a gondolatmenetbe, így használata erőltetetté válhat.
- A metaforák idővel elhalnak, jelentésüket veszítik. Az asztal metafora már több mint 20 éve jelen van, de már senki nem gondol rá ugyanabban az értelemben (főleg, ha íróasztalán már csak egy számítógép foglal helyet).
- A metaforák túl vannak használva. Ami segít egy kezdőt az eligazodáshoz, az sok esetben hátráltatja a haladókat a hatékony munkában.
- Ha hiányzik az értelmezési tartomány a metafora szintén félrevezető lehet. Az asztal metafora egy "kétkezi munkás" számára nem biztos, hogy ugyanazt jelenti, mint egy klasszikus irodai dolgozó számára.

²³Alan Cooper a metaforák használatát kritizálja az interakció tervezésben

²⁴ Dan Saffer "The Role of Metaphore in Interaction Design", "Criticism of Metaphor in Interaction Design" fejezetben

Az asztal metaforának az ideje lassan leáldozik. Habár ezt minden évben megjósolják, még mindig erősen kitart és jelen van. A kis méretű, kézbe vehető eszközeinkkel már nem ugyanazokat a feladatokat látjuk el, mint régebben. Ezek méreteikből adódóan sem helytálló a munkaasztal metafora. Egy sokkal átfogóbb, tágabb értelmezési tartományban használható metafora igénye kezd felmerülni.

Material design

Erre a problémára adtak széleskörű megoldást a Google kutatói, amikor megalkották és publikálták a "material design" tervezési stratégiájukat. Ez egy jól megfogalmazott szabályrendszer, ami absztrakt metaforákat használ. Az elmélet kiindulópontja az "anyag" metafora, amit a papír tapintható valósága inspirált. Ezt az anyagot ruházták fel "természetfölötti" képességekkel, szabad utat biztosítva a képzeletnek és a varázslatnak. Az "anyag" felülete a valóság érzetét kelti, ez a hasonlóság segíti a felhasználót az jelenségek értelmezésben. A valós érzet megtartása mellett ez az "anyag" változékony és flexibilis. A fények, felületek és a mozgás kulcsfontosságúak abban, ahogyan az "anyag" viselkedik, kölcsönhatásba lép a térrel és a többi "anyaggal". Ez a tervezési minta, az "anyagra" úgy tekint, mint minden eredőjére, egyfajta tulajdonság kötegre, hálózatra. A metafora nem konkrét, de a valóság, vagyis a következetes viselkedés érzését kelti. Vizuális fogalomkészlete a nyomtatott grafikából származik, ilyenek például a tipográfia, arányok, űr, színek és a képzelet ereje. A képanyelv nem dekorációs elem. Segítenek láttatni a hierarchiákat, jelentést hordoznak és vezetnek a figyelmet. A felhasználó cselekedeteinek a kiemelése és vezetése is alapvető fontosságú. Minden mozgás jelentéssel bír. Ebben a rendszerben a felhasználó az elsődleges mozgató, ezeket a gesztusokat kiemeli és erősíti a mozgás. Minden mozgás egy rendszerben történik, egy folyamat része. Ezt a folyamatot nem szabad megszakítani még akkor sem, ha az "anyag" átalakul, a tér átrendeződik. A mozgás jelentéssel bír, vezeti a figyelmet, segít megtartani a folyamatosságot érzetét, vagyis segít abban, hogy soha ne kelljen kilépni az adott kontextusból. Ebben a rendszerben nem létezhet olyan, hogy valami hirtelen megjelenik, vagy eltűnik. Minden valahonnan jön, születik, vagy elmegy, megsemmisül. A vizuális visszajelzéseknek határozottnak kell lenniük, egyben finomnak és nem tovakodónak. Azt kell kommunikálni a felhasználóval, hogy ez az "anyag" tele van energiával, ezt az energiát egy érintéssel bármikor fel lehet szabadítani. Ez a képzeletbeli "anyag", bizonyos szabályokkal rendelkezik. Szilárd, meghatározott helyet foglal el a térben, áthatolhatatlan, hajlíthatatlan, képes formát váltani, összekapcsolódhat más "anyagokkal", osztódhat. Minden "anyagnak" a saját síkjában kell maradnia. A térbeli mozgás is megengedett, de ez is jelentéssel bír. A felhasználóhoz közeli elemek

fontossága nagyobb. Abban segítik a felhasználót, hogy az adott időpillanatban mihez lehet hozzáférni, vagy mi a lehetséges következő lépés.

A material design működési mechanizmusa az érzet következetességében, a felállított szabályrendszer kifinomultságában keresendő. Habár felhasználása tervezői szempontból nem egyszerű, hiszen ellentétben az asztal metaforához, ezt nem lehet egy mondatban leírni, mégis fontos kiinduló pontját képezi a modern interfész tervezésnek. Az, hogy a használt metafora elég általános, biztosíthatja, hogy a digitális világ előre nehezen megjósolható fejlődéséhez hosszú ideig alkalmazkodni tudjon.

Lehetőség-tér

A kreatív eszközöknél a lehetőség-tér írja le, hogy milyen problémákkal foglalkozhatunk és azokra milyen megoldásokat leszünk képesek felfedezni. Saját, játékok készítésével szerzet tapasztalataim alapján, erős hasonlóságot fedeztem fel a játék és a kreatív eszközök tervezésében. A hagyományos, egyszerű játékok tanulmányozása sok esetben támpontot nyújthat a digitális eszközök tervezésben. Egy játék terét is az alap elemek és a rajtuk alkalmazott szabályok határozzák meg. Érdekes összehasonlítani két ősi játékot, a sakkot és a go²⁵-t. Habár mind két játéknak hatalmas mélységei vannak, az első és legszembetűnőbb különbség a szabályok számosságában kereshető. Míg a sakk közel 100 szabályból áll a go-nak csak 9 van. A sakk lehetőség-dimenzióit tekintve korlátozott, a viszonylag bonyolult és kötött szabályrendszere miatt. Ezzel szemben a go-nak megszámlálhatatlan dimenziója van, minden egyes lépéssel egy új, még nem látott tér keletkezik. Jól példázza a bonyolultságot az is, hogy az az első algoritmus (Deep Blue²⁶), ami megverte a sakk világbajnokot 1997-ben készült, amikor a go algoritmusok még labdába sem rúghattak a mesterekkel szemben. A közelmúlt és a mesterséges intelligencia egyik meghatározó eseménye az AlphaGo²⁷ győzelme Li-Se Dol go világbajnok felett. A sakk algoritmus “nyers-erő” (végigpróbálás) módszerével és egy hatalmas adatbázis segítségével az összes lehetséges játékkimenetet szimulálta és ezek közül megkereste a legmegfelelőbbet. A go esetében esély sincs ilyen módszerekkel próbálkozni, mivel nagyon hamar

²⁵ A go egy ősi kínai területfoglalós stratégia játék. Edward Lasek sakkmeister szerint „a szabályrendszere annyira finom, szervezett és szigorúan logikus, hogy ha univerzumban valahol máshol is léteznének intelligens életformák, azok majdnem biztosan gót játszanak.

²⁶ Deep Blue az IBM által kifejlesztett számítógép, amely 1997-ben, egy hatjátzmás páros mérkőzésen New Yorkban legyőzte Garry Kasparovot, az emberi sakkozás akkori világbajnokát.

²⁷ AlphaGo a Google DeepMind által kifejlesztett program, ami a gép tanulás eszközt alkalmazta. A Dél-koreai Goszövetség a mérkőzés után 9 danos mesteri címet adományozott az AlphaGo programnak, elismerve az „őszinte erőfeszítést”, hogy elsajátítsa a go taoista alapjait és hogy elérjen az isteni szférához közeli szintre.

elérjük azt a számosságot, ahol a hagyományos algoritmusok szóba sem jöhetnek. Egy 19x19-es go táblán a variációk száma emberi ésszel alig felfogható, 2 után 170 nulla. Ez is jól példázza, hogy a szabályok és a rendszerben szereplő elemek számossága milyen mértékben befolyásolja az általuk előállítható variációk mennyiségét.

Az olyan rendszereket amire igaz, hogy a részek összessége nem egyenlő az egész összességével, emergens rendszereknek nevezzük. Nem csak a játékokkal, vagy az egyéb gondolati rendszerekre illeszkedik ez a definíció. A természetben is számos ilyen példát találunk. Ilyenek a hal rajok, vagy a hangyák viselkedése. Az egyes ágensek a saját szintjükön nagyon egyszerű feladatokat látnak el, míg összességük hatalmas, külső szemmel leírhatatlanul változatos viselkedéseket mutatnak. Hasonló problémával szembesülnek a meteorológusok is. Egy terület időjárása, nem csak az adott területen jelen levő időjárás változók összessége. Paradox módon, ennek a rendszernek a változói, az azt körülvevő változók eredményei. A föld felületének lokális változói globális szinten olyan összetettséget mutatnak, amely mai tudásunk szerint kiszámíthatatlan. Az időjós kifejezés mai napig is teljes mértékben fedi a valóságot. Ha az ilyen rendszereket elemeire szedve, összetevőit izoláltan vizsgálánk, a rendszer összeomlana, nem működne tovább. Az kulcs az elemek közötti kölcsönhatásokban, az azokat leíró szabályrendszerben keresendő.

Vegyük az emberi játékok egyik legrégebbi, egyben legegyszerűbb példáját a homokozást. A játék szabályrendszere magából az anyagtulajdonságokból fakad, míg a játék célja teljesen kötetlen. Egyes nézetek így definiálják magát a játékot: olyan viselkedés, amelyben egy magasabb rendű élőlény, saját maga elé állít akadályokat és ezzel saját maga szabályozza örömszerzésének mértékét. A homokozás célja nem meghatározott, az adott játékostól, játékosoktól függ. Színpada lehet az építés öröme, a háborúk kudarcának, vagy egyéb természeti jelenségek pusztító erejének. Ebben a játékban a hiba nem büntetés, hanem a tapasztalatszerzés útja, az anyag természetes része, a játék velejárója. A homokozó analógia ígéretes kiinduló pontja lehet a kreatív eszközök készítésénél. Számos digitális eszköz vagy játék hasonló elvekre épül. Az ilyen jellegű működési mechanizmusokat "sand-box" (homokozó) játékoknak is neveznek. Ezekre jellemző, hogy kevés és egyszerű szabállyal rendelkeznek. Érthető, hiszen ki szereti a fölösleges bonyolultságot. Ezzel szemben elvitathatatlan, hogy korunk technológiai termékei egyre összetettebbek lesznek, ezáltal egyre bonyolultabbakká is válnak. A termékek eladhatósága szempontjából meghatározó szerepet töltenek be a új tulajdonságok (features) halmozása. Minden egyes szoftver frissítés, vagy új verzió ezek miatt adható el. Emiatt a piaci befektetők körében is elterjed nézet, hogy az egyszerűség nem jó hívószó, az ilyen termékeket nehéz eladni. Mindeközben egy egyszerű, de jól megtervezett eszköz felszabadíthatja használóját. A szabályok és elemek redukálása, vagyis a tudatos korlátok felállítása kreativitást szül. A kreatív emberek szeretik a szabadságot, de egyben igénylik a határokat is, hiszen ezek feszegetése vezethet a meglepő, nem várt eredményekhez.

Transzparencia

A transzparencia, vagyis az, hogy az interfész mennyi betekintést enged a rendszer működésébe, talán az egyik legmeghatározóbb tervezői döntés. Minden rendszer alkalmaz fekete-dobozokat, ez elkerülhetetlen, hiszen a rendszerek legmélyére (tulajdonképpen az atomok szintjére) senki nem akar lenézni, hisz ez nem is cél. A kérdés az, hogy milyen funkciókat akarunk, hogy a felhasználó láthasson, milyen alapvető építőköveket adunk a kezébe. Bizonyos feladatoknál, amikor a felhasználót egyáltalán nem érdekli a mögöttes működés, a fekete-doboz alkalmazása teljesen kielégítő. Egyszerű példával élve, ha ki akarjuk kapcsolni számítógépünket elég egy gombot megnyomni rajta, holott egy rendszer kikapcsolása igen összetett feladatsor, a gépnek több száz műveleten kell keresztül mennie, hogy a végén valóban lekapcsolódhasson. Az ilyen, felhasználó szemszögéből egyszerű feladatoknál igen hasznosak a fekete dobozok, hiszen segítenek a fontosabb feladatokra fókuszálni. Itt a kérdés az egyensúly megtalálása az adott cél érdekében. Ha egy szoftver alap-építőkövei nagyon általánosak (ilyenek a programozási nyelvek), segítségükkel szinte bármilyen problémát meg lehet oldani, viszont használatuk bonyolulttá válik, elsajátításuk sok időt vesz igénybe. Ha viszont túl speciálisak, használatuk könnyű lesz, viszont a rendszer egészébe véve korlátossá válik. Mindig az adott cél érdekében kell megválasztani az építőköveket. Amikor egy gyerek a Logo²⁸ nyelv segítségével egy utasítást állít össze, nagyon egyszerű elemeket használ, pl.: menj előre és fordulj jobbra. Egy utasítás ami négyzetet rajzol, így néz ki: repeat 4 [forward 50 right 90], vagy egy kör lerajzolásához: repeat 360 [forward 1 right 1]. Ezzel a maguk szintjén éppen elég betekintést nyernek az algoritmikus gondolkozásba, azaz a programozás alapjaiba, miközben igazából fekete-dobozokat használnak. Az "előre" utasítás belső működése el van rejtve, hiszen a gép ilyenkor trigonometriai műveleteket alkalmaz az új pozíció kiszámolására. Ha ez a nyelv a trigonometriát is szeretné megértetni, akkor az ilyen fekete-dobozok alkalmatlanok lennének a feladatra. Mivel a Logo nyelv elrejt a bonyolultabb, a célt nem erősítő műveleteket, az ezt használó gyerekek felszabadultan kísérletezhetnek, a magasabb szintű problémákkal foglalkozva. Ez éppen elég információ a matematikai és geometriai szabályszerűségek felfedezéséhez²⁹. Ebből is látszik, hogy a cél és a célközönség pontos definiálása mennyire kulcs fontosságú az alapelemek egyensúlyának meghatározásában. A kreatív eszközöknél fontos, hogy a lehetőség-tér elég tág legyen, így az alapelemeknek is általános funkciókat kell betölteniük, hiszen cél, hogy felhasználók ezek

²⁸ A Logo egy egyszerű számítógépes nyelv, főként "teknőcgrafikájáról" ismert, leggyakrabban gyerekek programozás oktatására használják.

²⁹ Mitchel Resnick: "Design Principles for Tools to Support Creative Thinking" c. írása alapján

összekapcsolásával szabadon kísérletezzenek. Ha egy rendszer elég általános és megengedi a kísérletezést, egyben a hibázás lehetőségét is megteremti. Fontos, hogy itt a hibázás ne “büntetéssel” járjon, legyen lehetőség nem működő dolgok létrehozására is. Ha mindig vissza lehet vonni az előző cselekedetet (undo), vagy a rendszer nem engedi a nem összeillő elemek összepárosítását, nem lesz kockázatos a próbálgatás. Az olyan rendszerekben ahol nincs, vagy kicsi a kockázat, hatalmas kreatív ugrásokat lehet meglépni³⁰. A hibázás természetes velejárója a tanulásnak. Ha kreatív embereknek akarunk tervezni, ezt fontos figyelembe venni. A hibát a mai kor gondolkodása még mindig egy negatív dolognak tekinti. Társadalmi rendszerünk arra épít, hogy minél kevesebbet hibázzunk. Általában egy nagyon hierarchikus rendszer nem tolerálja a egyéni szabadságfokot, hiszen az instabillá, a piramis csúcsából nézve, kiszámíthatatlanná teszi az egész működését. Oktatási rendszerünk arra treníroz, hogy minél kevesebb hibát ejtsünk, pedig a hiba egy új gondolat születésének kezdete is lehet. Történelmünknek a legtöbb nagy felfedezéséért is a hibák a felelősek. Kolumbusz “hibája” miatt indult el Amerika irányába, tévedése az egész világot megváltoztatta.

A kreatív eszközöknél is meg kell jelenni annak a lehetőségnek, hogy adott esetben a felhasználó által létrehozott hiba, többletet adhasson, új felhasználási területeket nyisson. A hiba csak olyan esetben jelent problémát, amikor nem értjük az okát, ezért nem is tanulhatunk belőle. Amikor egy rendszer átlátszó, az ok megfejthető, a hiba értelmet nyer.

A fekete dobozok alkalmazásának megfelelő egyensúlya és a szabad játéktér biztosítása hozzásegíthet ahhoz, hogy az eszköz által létrehozott megoldások egyénenként eltérhessenek, személyes jegyeket hordozzanak. Egy szoftver tervező számára, a legnagyobb siker és legértékesebb visszajelzés, ha az általa létrehozott eszközzel olyan dolgot képes egy másik ember előállítani, amire ő egyáltalán nem számított. Amikor egy adott jelkészletből nem várt többlet születik, az annak a jele, hogy a felhasználó egyéni kreativitása képes a felszínre törni.

Az ilyen rendszerekre jellemző a nyelv-szerűség, hiszen a természetes nyelveknek is sajátossága, hogy a részek összege nem egyenlő az egészszel. Jelkészletük nem határozza meg az általuk létrehozott mondatok, gondolatok tartalmát. A lényeg a betűk, később a szavak, gondolatok közötti kapcsolatokban keresendő. A szoftverek tekintetében a jelkészletnek tekinthetjük azokat a funkciókat, melyek a felhasználó rendelkezésére állnak. Az üzenet, az ezek közötti relációk megalkotásából jön létre, vagyis a különböző funkciók összekapcsolásából.

³⁰John Maeda: “Creative Leadership” c. könyvéből

A következőkben sikeres, nemzetközi példákkal illusztrálnám a nyitott rendszerek lehetőségeit. Habár mindegyik különböző kreatív problémára ad megoldást, hasonlóságuk a nyelv-szerúségben rejlik. Abban hogy olyan jól megtervezett eszközkészlettel rendelkeznek, melyek teret adnak a személyes problémák megoldásának, az egyéni kreativitás kibontakozásának.

IFTTT - *nagy számosságú jelkészlet, egyszerű szabályrendszer*

A digitális technológiával dolgozó szakemberek (programozók) alapvetően saját problémájukra keresik a megoldásokat. Sok esetben, amikor a probléma nagyon személyes, vagy újszerű, még nem léteznek eszközök hozzá. A tervezőnek így saját magának kell megalkotnia azokat. Ezek az eszközök a használat alatt egyre tökéletesebbek lesznek, míg nem elérik azt a szintet, hogy egy nagyobb közösség számára is hasznosak lehessenek, így más emberek is beépíthessék saját problémamegoldó folyamataikba. Hasonló módon történt ez az IFTTT azaz "If-this-then-that" (ha, ez, akkor legyen az) szoftver kialakulásánál is.

Ez egy olyan szolgáltatás, amivel a felhasználók automatizálhatnak bizonyos digitális folyamatokat. Az ötlet megszületése előtt a tervező azzal foglalkozott, hogy hogyan tud adatokat, digitális folyamatokat vizuálisan láttatni. Ezen való gondolkodása világított rá, hogy társadalmunkat mennyire átszövik és formálják a digitális eszközeink és a hozzájuk kapcsolódó virtuális szolgáltatások. Ez arra is rámutatott, hogy mennyire fontos, hogy nem passzív, hanem kreatív módon alakíthassuk a technológiát saját igényeinkre. Ez a képesség viszont csak azoknak adatott meg, akik járatosak a programozási nyelvekben és átlátják a különböző cégek szolgáltatásainak működését. A IFTTT a mai korunk digitális pillanat ragasztója. Segítségével az átlagos felhasználó képes különböző már létező szolgáltatásokat összekötni, így automatizálva a számára fontos folyamatokat. Az egyes szolgáltatások kimenetét pár mozdulattal más szolgáltatások bemenetéhez lehet rendelni. Mivel ez egy nagyon egyszerű és tiszta gondolat, az interfész is igen könnyen használható. A szabályrendszer maga a név: ha **ez** bekövetkezik, akkor legyen **az**. Ha a munkahelyről **haza indulok**, akkor **kapcsoljon** be a fűtés. Ha a tőzsdén **változik** egy részvényem árfolyama, **kapjak SMS-t** a telefonomra. Ha **besötétedik**, akkor **kapcsolódjanak fel** a lámpák. Itt az eszközkészlet folyamatosan bővül (jelenleg 347 db), ahogy újabb és újabb "okos tárgyak" (IoT³¹), vagy szolgáltatások jelennek meg a piacon. A rendszer másik, figyelemre méltó tulajdonsága az, hogy közösségileg bővíthető szókinccsel

³¹ Az "Internet of things" gyűjtőneve azoknak az eszközöknek (tárgyak, járművek, épületek) melyek a világhálón keresztül összeköttetésben állnak egymással, vagy más számítógépekkel. Ezek az eszközök hálózaton keresztül cserélnek információkat, automatizálva működésüket. Erre a már meglévő információs architektúrát használják. Ez az első lépés a kiber-fizikai tér megalkotásához.

rendelkezik. A tervező felismerte, az emberek nagy része, hasonló problémákat akar megoldani. Erre és a tanulás megkönnyítése céljából létrehozott egy folyamatosan bővülő galéria (eszközpark), ahol a praktikus összeköttetéseket (logikai gépezeteket) lehet egyszerűen megszerezni, felhasználni. Ezt a galériát a közösség fejleszti, önszervező, önszabályozó módon.

Reactable - *jelkészlet és szabályrendszer kifinomult egyensúlya*

A “reactable”, ami egy digitális hangszer. Működésének alapja egy olyan vizuális nyelv, melynek segítségével a felhasználó szabadon kísérletezhet, ismerkedhet meg a digitális zenekészítés alapfogalmaival. A készítő Sergi Jordá, maga is a kísérleti elektronikus zenei szférából származik. Koncertjein, performanszein olyan módszerekkel állított elő zenét, melynek alapjai az algoritmusok és az élő improvizáció keveredéséből származnak. Sok év előadói tapasztalat és annak a megfigyelése, hogy mások hogyan használják a saját maga számára készített szoftvereket, tette lehetővé ennek a kiforrott, a digitális világában paradigmaváltónak tekinthető hangszer megszületését.



(Reactable asztal működés közben)

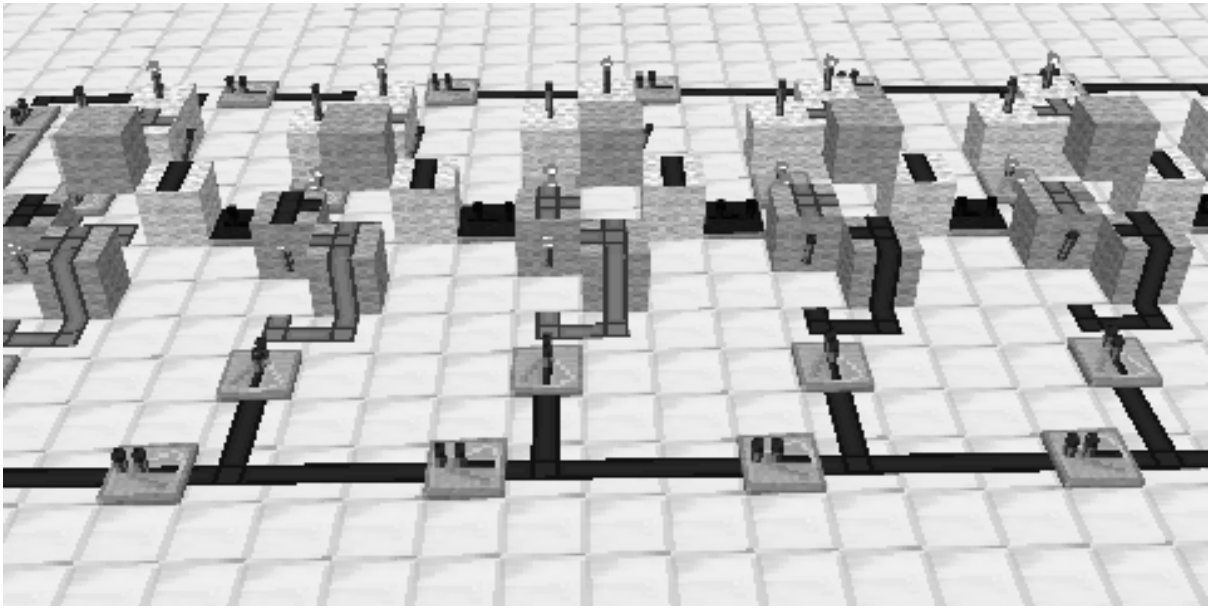
A hangszer egy kör alakú vetített felületből és az erre elhelyezhető fizikai objektumokból áll. A fizikai objektumok a rendszer funkcióit, azaz jelkészletét reprezentálják. Az objektumok egymás mellé helyezésével automatikusan kapcsolat alakul ki közöttük. A kapcsolatot és az objektumok típusát az asztalra történő vetítés láttatja. Nincs használati útmutató, nincsenek instrukciók. A szoftver megtervezésénél különös figyelmet fordítottak arra, hogy a rendszer mindig működjön, vagyis a tanulási folyamatot, vagy adott esetben az előadást ne szakítsák meg az esetleges hibák. A folyamatos képi visszacsatolás segíti értelmezni a rendszer működését. A szoftver gondolati háttere a hang szintetizátorokból, majd az ezekből fejlődő vizuális programnyelvekből ered. A nyelv-szerűség jegyeit hordozza, bár nem tekinthető valódi programozási nyelvnek, pont a specializált jelkészlete miatt. Viszont ez a redukció egy tudatos tervezői döntés eredménye, hiszen itt egy speciális feladat, speciális követelményeit kell kielégíteni. Ilyen a kollaboráció lehetősége, az, hogy egyszerre többen is használhatják, valamint, hogy a "programozás" taktilis úton, objektumok térbeli rendezésével történik. Figyelemre méltó, hogy az interakció alapját képező, reactivision³² szoftver modult, a készítőik szabad forráskóddal publikálták. Így ez a projekt több szempontból is meghatározó szerepet töltött be a digitális művészet fejlődésében.

Minecraft - közösség által bővülő jelkészlet, kötött, egyszerű szabályrendszer

A 2011 körül jelent meg a Minecraft nevű számítógépes játék, ami 2014-re a legkelendőbb játék lett világszerte. (A játékot többféle képen lehet játszani: túlélő, kreatív, vagy kaland módban. A mi esetünkben a kreatív mód az érdekes.) A játék nem vizuális megjelenése miatt, hanem a benne rejlő kreatív potenciában keresendő. Népszerűsége arra mutat rá, hogy ezekre a kreatív játékokra hatalmas igény van. A játék alapja egy adott felbontású 3 dimenziós tér (térbeli rács-háló), aminek egyes részeire különböző tulajdonságú elemeket lehet elhelyezni. A különböző elemek különböző viselkedésekkel bírnak. Alapvetően ezzel térbeli objektumokat, tájakat, épületek lehetett modellezni. A játék köré épülő játékos közösség nagy befolyást tett a játék kialakulására, így később a fejlesztő kiegészítette a jelkészletét mechanikus, mozgó, robbanó, áramot vezető és még számos³³, funkcióiban eltérő elemmel.

³² A Reactivision egy olyan nyílt forráskódú szoftver, amely a kamerakép elemzésével képes megállapítani a látótérbe elhelyezett speciális objektumok helyzetét, típusát és elfordulását

³³ Egy speciális építőkocka (redstone) felhasználásával komplex rendszerek építhetők, mechanikus gépek, vagy elektronikus áramkörök felhasználásával



(Minecraft-ban épített működő számológép, részlet: logikai kapcsolók, forrás: <http://minecraft.gamepedia.com>)

A játék sikerének a másik kulcsa, hogy több ember egyszerre, egy hálózatban, egy virtuális térbe kerülhet, másokkal együtt oldhat meg problémákat. A kooperáció és a verseny első számú mozgató rugó, a játékosok előre nem meghatározott, közösségre kialakított célokért küzdenek. A játékosok rengeteg időt töltenek el ezekben a virtuális kommunákban, virtuális tereket építve, így a szoftver érthető módon hamar felkerült az iskolák fekete listájára. Egészen addig, míg pár nyitottabb gondolkodású pedagógus fel nem fedezte, hogy a játékokban rejlő kreatív energiát saját céljukra is használhatják. Terepe lehet egy kollaboratív, problémaközpontú oktatásnak, ahol hasonlóan más pedagógiai megközelítéshez, szimulált környezetben, szimulált problémákra kell megadni a megfelelő választ. Nem is beszélve arról, hogy egy játék orientált oktatás számos olyan frusztrációtól³⁴ kíméli meg a tanulót, ami sok esetben gátolná a szabad gondolkodást, asszociációt, kreatív problémamegoldást. A játék hatékony oktatási eszköznek bizonyult fizika, kémia, történelem órákon is, ezért létre jött egy kissé módosított változata a MinecraftEdu, ami már kifejezetten ezekre a célokra fókuszál. Az ilyen eszközök, megfelelő körültekintéssel alkalmazva, a jövő oktatási segédeszközei is lehetnek. A játékosok a rendszerben készített alkotásaikat megosztják egymással, találmányaikat bemutatják, olyan leírásokat készítenek, melyekkel mások is felhasználhatnák azokat. A szabad szoftverekhez hasonlóan ez is egy közösségre folyamatosan fejlődő tudásbázis. Itt a kód szintjénél eggyel

³⁴ A játszva tanulásra jellemző a tanulási folyamat egyéni irányíthatósága, a hibázástól való félelemnek a hiánya, a tanulás eredményének azonnali és vizuális megjelenése, belülről jövő motiváltság (Joey J. Lee: Gamification in Education: What, How, Why Bother?)

magasabban, az interfész és az interakció szintjén jelenik meg ugyanaz a minta. Ez a számítógépes nyelveknél sokkal emberközelibb “nyelv”, tanulása játékos, empirikus módon történik. Az, hogy ezt a rendszert korosztály nélkül bárki játszhatja, a “nyitott világ” metaforából ered, abból, hogy a virtuális világ építőköveit a valós életből vett elemek alkotják. Emiatt a hasonlóság miatt létrejöhet egy fajta tudás átjárás a valós és a virtuális tapasztalatok között.

Mestermunka

Cím

Animata

Leírás

Az Animata egy meta-eszköz. Egy olyan, általam fejlesztett szoftver, amely interaktív felületek előállítására szolgál. A működését meghatározó vizuális nyelv, intuitív, teret enged a kísérletezésre, a tanulásra, az egyéni kreatív megoldások megjelenésére. A szoftver olyan eszközkészlettel és szabályrendszerrel rendelkezik, ami lehetőséget teremt az animáció és az interakció lehetőségeinek széles körű feltérképezésére.

Előzmények

Mestermunkám egy hosszú távú kutatás aktuális állapotának tekinthető. Mint azt a korábbi példákból is láthattuk, ezeknek az eszközöknek hosszú utat kell megtenniük, hogy ne egy személyre szabott egyedi, hanem egy közösség által használható eszközzé válhassanak. Eredete itt is személyes motivációból származik. Nagy hatással volt rám a SodaPlay nevű, rugó-fizikán³⁵ alapuló kísérleti játék. Ebből kiindulva kezdtem, magam számára létrehozni egy rendszert, ahol csomópontokat lehetett "rugókkal" összekötni. A pontok összekötéséből, különböző fizikai strukturák alakultak ki, melyekre úgy tekintettem, mint egy petri csészében létrehozott kezdetleges életformákra. Egy másik kísérletben ugyanezt a technikát alkalmaztam egy állókép megmozgatására, úgy, hogy a csomópontok összekötéséből egy rácsháló született. Ez az állókép a kép egyes pixeleit deformálta. Ez az új elasztikus-kép-anyag a egy különös típusú fizika törvényszerűségeknek megfelelően viselkedett, intuitív módon lehetett deformálni, forgatni, mozgatni. Ezek a kísérletek és az abban az időben kibontakozó magyarországi VJ kultúra hozta meg bennem az igényt, egy élő vetítésekhez használható mozgókép-szintetizátor létrehozására. Az állóképekre felépített csontvázrendszer segítségével,

³⁵ A verlet integráció egy olyan matematikai módszer, amivel a Newton mozgásegyenletei egyszerűen kiszámíthatóak. Az Animatában alkalmazott implementáció két pont távolságának megtartására a következő pszeudó kód formájában:

```
myPos += normalize( otherPos - myPos ) * ( magnitude( otherPos - myPos ) - targetDistance ) * damping
```

„digitalizálni” lehetett a képen szereplő karakter, vagy tárgy belső összefüggéseit. Miután az ember elvégezte ezt a manuális munkát, a szoftver magától, vagyis a felhasználói interakció hatására, volt képes, a fizika törvényeinek megfelelően deformálni az állóképeket. Ezt a rendszert neveztem el Animatának. Gyenei Péter (akkoriban VJ) barátommal közösen használtuk a szoftvert, mindkettőnk igényeinek megfelelően bővítve a rendszer képességeit. A Kitchen Budapest lehetőséget adott rá, hogy egy profi programozó csapattal közösen új, sokkal korszerűbb alapokra helyezzük eredeti gondolataimat. Számos installációban alkalmaztuk sikeresen az így létrejött szoftvert, ami később megnyerte a Pixelache³⁶ az „év szoftvere” díjat. Ezek miatt a sikerek miatt sok külföldi workshop-ot tartottam, sok ember véleményét megismertem. Mivel abban az időben (ahogyan most is) inkább a kísérletezés és a felfedezés öröme motivált, így inkább más, újabb ismeretlen területek feltérképezése fele indultam a szoftverek területén. Az Animata fejlesztése jó pár évre megállt, egy olyan állapotban, amiből egy nagyon fontos elem hiányzott. E nélkül, habár a szoftver jól használható és intuitív volt, a rendszer nem rendelkezett megfelelő mennyiségű építőköccel, amivel mozgásokat variálni lehetett volna. Így minden egyes érdekesebb mozgáshoz, külső, erre a célra írt szoftvereket kellett fejleszteni, ami tulajdonképpen az eredeti céllal teljesen ellenkező irányba vitte a használatot. Csak olyan emberek voltak képesek a rendszer lehetőségeit ténylegesen kihasználni, akik tudtak programozni, vagyis a célközönség beszűkült. Figyelemreméltó, hogy még így is sokan felhasználták, dokumentációkat készítettek a használatáról, bővítették és projektjeikbe integrálták a rendszert. A DLA tanulmányaim idején, más alkalmazott munkáimhoz egyre gyakrabban merült fel az igény az eredeti rendszer újragondolására és újraírására. Ehhez nagy lendületet adott a Tünet Együttessel közösen létrehozott darab³⁷ interaktív látványának megalkotása. Az viszonylag limitált, rugó-fizika helyett, a rendszer magjaként, egy sokkal kifinomultabb fizikai szimuláció³⁸ kapott helyet. Ennek az eszköztára rengeteg új lehetőséget adott kísérletezésre, komplex, meglepő fizikai rendszerek megalkotására. A képek, csontvázhoz való rendelése is sokat finomodott, a lehetőségek egész tárházát nyitva a különböző képi megoldások tekintetében. A színdarab fontos részét alkotta a szabadkézi rajz alkalmazása, így a rendszer által megmozgatott állóképeket digitális ceruza segítségével is elő lehetett állítani. Így jött létre egy egész különleges felhasználási terület, ahol az alkotó üres lappal érkezik, folyamatosan rajzolja, építi és mozgatja fantáziájának szüleményeit, lehetőséget nyújtva a szabad képi asszociációk megteremtésére. A szoftver használata lehetőséget ad arra, hogy a színészeknek ne kelljen előre definiált helyekre állniuk, vagy betanulni a rögzített dramaturgiát. A látványt létrehozó alkotó, ennek a módszernek a segítségével a színészekkel egyenrangú módon képes reagálni az

³⁶ Helsinki, digitális művészeti fesztivál (2010)

³⁷ Tünet Együttessel, Az éjszaka csodái c. előadás (2015), vizuális alkotók: Samu Bence, Szabó Réka, Mezei Ildikó, Taskovics Éva

³⁸ Box2D egy C++ nyelvben írt, nyílt forráskódú, 2 dimenziós fizikai szimulációs rendszer. Elsősorban számítógépes játékoknál alkalmazzák, egyik legsikeresebb alkalmazása az Angry Birds nevű játék.

eseményekre, improvizálni a táncosokkal, vagy korrigálni azok esetleges hibáit. A Dürer mozgásban c³⁹.
. installáció elkészítésénél lehetőségem adódott egy komplex, interaktív történet megalkotására. Itt több, különböző mélységben elhelyezett rétegből állt össze a látvány, melynek mozgatása különleges térbeli hatást adott a jeleneteknek. Ezekből a tapasztalatokból leszűrt következtetések vezettek a jelenlegi rendszer kialakításához.

Célcsoport

A szoftver célcsoportjában olyan kreatív emberek állnak, akik nyitottak az új média és a vizuális kommunikációban rejlő lehetőségekre. Belépési pontot biztosíthat kezdők számára, de hasznos eszköz lehet azoknak, akik ötleteiket vizuális, interaktív módon szeretnék megjeleníteni, de nem rendelkeznek a programozás képességével. A szoftver lehetőséget nyújt a nem lineáris történetmesélésre, kísérletezésre az interaktív média, és a játékkészítés területén. A szoftver akár oktatási célra is alkalmazható, fizikai, logikai és interakciós alapelvek elsajátítására. Felhasználási területe tágas, szándékosan nem definiált. Az eddigi tapasztalatok alapján alkalmas lehet interaktív könyvek, színházi vetítések, adatvizualizációk, digitális bábozás, vagy akár önálló játékok megalkotására is.

³⁹ Gönci Biblia múzeum számára készített installáció, mely Dürer fametszeteit felhasználva mutatja be Jézus életét. (2015)

Tervezési stratégiák

Egyszerűség

Metafora

A szoftver tervezés az egyik legmeghatározóbb alapelv az egyszerűség, a könnyen kezelhetőség. Ennek elérése céljából az Animata egy komplex metafora rendszert alkalmaz. Távolról nézve, az egész szoftvert tekinthetjük úgy mint egy jól felszerelt laboratórium, ahol fizikai és elektronikai kísérleteket végezhetünk. A felhasználó ebben a laborban különböző gépeket, robotokat, vagy bábokat készíthet. Az Animata elnevezés az animáció és az automata, vagy automatizálás szavak összetétele. De a anima (lélek) jelentés is fontos szerepet játszik a lehetőségek feltárásában, mivel a rendszerben megalkotott gépeket, bábokat mozgásba lehet hozni, autonóm viselkedésre bírni. A rendszer nyitott az emberi interakcióra, vagyis a felhasználó egyszerű gesztusokkal foghatja meg, rendezheti át az elemeket, szerkesztheti a jeleneteket.

Intuíció

A rendszer működése fizikai szimuláción alapul, azaz a virtuális tárgyak viselkedését a valós világhoz hasonló törvényszerűségek vezérik. Emiatt az emberi tapasztalataink könnyen kamatoztathatóak ebben a virtuális világban is, hiszen intuitív módon értjük, érezzük a szimuláció működését. Ez igaz lehet az ellenkező irányba is. A virtuális gépek működésének tanulmányozása, hasznos lehet a valós élet jelenségeinek megértésében is.

Redukált képi fogalmazásmód

A szoftver síkbeli felületekkel dolgozik. Ez egy tudatos tervezői döntés, ami az interakció szempontjából fontos. Jelenlegi 2 dimenziós interfészekkel, képernyőkkel a 3 dimenziós szerkesztés nem intuitív, egyelőre nem létezik rá jó megoldás. A 3 dimenzió lehetőségei nem adnak annyit hozzá az végeredményhez, mint amilyen körülményessé teszik a használatot. Szerencsére, a hagyományos, 2 dimenziós mozgóképes nyelv igen gazdag tárházat biztosít. Ezekből kiindulva a szoftver vizualitását a papírkivágásos animáció, a kép-morfózis, illetve a háttérben meghúzódó pszeudó-fizika határozza

meg. A felhasználó a virtuális bábok különböző elemeihez fénykép, vagy egyéb módon előállított állókép részeit rendelheti, úgymond felöltöztetheti azokat. Ez a működési elv is alátámasztja a 2 dimenzióra való redukciót, hiszen ilyen képeket nagyon könnyen, bárki előállíthat, erre már számos, bevált eszköz áll a rendelkezésünkre. (fénykép, grafika, festmény, stb). Ez a vizuális nyelv éppen elég nyitott és zárt is egyszerre, hogy felszabadíthassa a felhasználó kreatív, vizuális és mozgóképes gondolkodását. Az így kialakítható képi világ teljes mértékben személyre szabható.

Transzparencia

Jelkészlet

Mint minden kreatív eszköznél, ebben az esetben is az alapelemek meghatározása és a szabályrendszer kialakítása a legmeghatározóbb tervezői feladat. Ez dönti el, hogy a rendszer mennyire lesz nyitott, ez határozza meg a jelkészletet, vagyis magát a szoftver anyagát. Ez a rendszer nagyon általános elemekből építkezik, kis redundanciával⁴⁰ rendelkezik. Csak olyan elemek kerülhetnek bele, melyek más alapelemekből nem állíthatók elő. Erre példa lehet az, hogy a forgó mozgást egyszerűen lehet egyenes vonalú mozgássá alakítani (forgattyú mechanizmus), így az ilyen fajta mozgásra nincs külön funkció. Az ilyen elemek nem bővítik az eszköztárat, csak a bonyolultságot. Azt, hogy ne kelljen a felhasználónak minden esetben megépítenie ezeket a gépeket, hogy ne kelljen fölösleges időt töltenie egyszerű feladatok megoldásával, a szoftver más stratégiákat alkalmaz. A hasznosnak bizonyuló mechanikai rendszereket, mint alapegység lehet megőrizni, hogy azok könnyen, újra felhasználhatóak legyenek. Ez a megoldás segíti a rendszer transzparenciáját megtartani, csökkenti a fekete dobozok számát, egyúttal erősíti a nyelv-szerűségét. Hasonlóan a természetes nyelvekhez, ahogy a betűkből előállított szavak, ezek az alapegységek is, egy magasabb szintű építkezés alapkövei. A jelenlegi rendszerben a következő építőelemek szerepelnek: szilárd testek, forgópontok, motorok, hajtóművek, szenzorok. Ezek az elemek a szoftver fejlődésével együtt változtak, számos gyakorlati helyzetben bizonyítva hatékonyságukat.

Szabályrendszer

A szabályrendszert két koncepció alkotja. Az **első** a mechanika nyelve, vagyis a szilárd testek viselkedése. Ezt a nyelvet a emberiség már több ezer év óta tanulmányozza és használja. Egyik

⁴⁰ Nyelvtudományi fogalom; a közlésben az egyértelmű megértéshez elegendő minimumon felüli, ezért fölösleges többlet. Terjengős kifejezések alkalmazása egyszerűbb szavak helyett, pl. javasol - javaslatot tesz.

legrégebbi példa erre Al Jazari⁴¹ 1206-ban megírt könyve a *Book of Knowledge of Ingenious Mechanical Devices* (Ötletes mechanikus eszközök tudásának a könyve). A feltaláló ebben olyan működési mechanizmusokról ír, melyek tekinthetően a mai programozás alapjának is. Az Iszlám automaták illúzió keltésre készültek, urak szórakoztatására, lenyűgözésére. Varázsuk abból származott, hogy a háttérben meghúzódó mechanikus programnyelvvél mozgatott szereplők, élő hatást keltettek. Habár ezek az automaták programozhatóak voltak, megalkotóit nem a mai értelemben vett programozás motiválta. Arra a kérdésre keresték a választ, hogy hogyan lehet az anyagba információt tárolni, majd azt az idő felhasználásával kinyerni és ezáltal a bábokat mozgásba hozni.



(1.: Al Jazari leghíresebb találmánya az “Elefánt óra” - 2-3.: Japán Karakuri bábuk, forrás: wikipedia)

Theo Jansen⁴² kinetikus szobrai még ennél is tovább mennek. Az általa megalkotott gépek, mechanikus szenzorok segítségével képesek érzékelni az őket körülvevő környezetet. Egyik munkájában víz-érzékelő berendezést használ arra, hogy a kinetikus-lény mozgásának irányát megváltoztassa. Az így létrejövő viselkedés túlmutat az automaták “programozott” mozgásán, inkább a valódi élőlények működésére emlékeztet. Ezek a lények nem passzívak és ha csak korlátozott mértékben is, de reagálni képesek a külvilágra.

⁴¹ Al Jazari (1136-1206) Arab feltaláló, mérnök, művész, matematikus.

⁴² Theo Jansen (1948-...) Holland kinetikus művész, 1990-ben kezdte el készíteni PVC csövekből konstruált, önműködő gépeit. Állítása szerint a művészet és a mérnöki megoldások közötti különbség csak a gondolatainkban létezik.

Ez a *második* koncepció ami megjelenik az Animata szabályrendszerében is. A szoftver működése számos módon eltér a hagyományos animációs eszközökétől. Ami talán a legszembetűnőbb, hogy itt nem jelenik meg idővonal (timeline). A mozgást az elemek fizikai tulajdonságai, illetve a rájuk gyakorolt erőhatások határozzák meg. Az időbeliséget a mozgás és az ok-okozati összefüggések teremtik meg.



(Theo Jansen: Strandbeest, szél által hajtott, PVC csövekből konstruált kinetikus szobra)

Fizika

A szoftver szabályrendszerét több szinten lehet vizsgálni. Az alapvető működés szabályrendszerét a fizikai szimuláció alkotja. Mivel ezt a felhasználó intuitív módon érzi, bizonyos tekintetben egyszerűnek is mondható. A tárgyak nem hatolnak át egymáson, hanem összeütköznek, az összecsavazott testek a forgópont mentén elfordulnak, a gravitáció lefele húzza a tárgyakat. Persze a szimuláció mélyén számtalan szabály húzódik, de ezt a fekete-dobozok tudatos használata elfedi a használója elől, és csak azokat a funkciókat, tulajdonságokat láttatja, melyek hozzáadnak a mozgóképes nyelv színesítéséhez. Ilyen lehet a testek tömege, a felületek tapadása, a forgópontok súrlódása, a motorok ereje.

Erők

Az Animatával megépíthető gépek, bábuk erőhatások nélkül csak mint rongybabák viselkedhetnének. A rendszer jelenleg három fajta erőhatással rendelkezik. A testek forgópontjába motorokat helyezhetünk el. Ezek a motorok automatikusan, vagy külső esemény hatására, mint izmok mozgásba hozhatják az építményeket. A hajtómű bizonyos irányba lökheti az elemeket. A harmadik erő, maga a felhasználó interakciója. Egyszerű gesztusokkal egy báb könnyen megmozgatható, egy animáció pillanatok alatt eljátszható.

Szenzorok

Az szenzorok a rendszer utolsó építőkövei. Az erőt kifejtő elemekhez (motor, meghajtó) érzékelőket rendelhetünk. A szenzorok a tér bármely távoli pontjára elhelyezhetjük. A szenzorok más testekkel érintkezve ingereket közvetítenek az "izmokba", az így összekötött szerkezeteknek bizonyos autonómiát biztosítva. A szenzorok alkalmazásával lehetségessé válnak ok-okozati összefüggések kialakítása, ezek segítségével a gépek képesek érzékelni környezetüket, interakcióba kerülhetnek más elemekkel, reagálni tudnak környezetükre.

Nyelv-szerűség

A szoftver "kinetikus-nyelvét" a fenti elemek kombinációja alkotja. A redukált eszközkészlet és szabályrendszer biztosítja a tág lehetőség-teret. A felhasználó szabadon kísérletezhet az elemek összekapcsolásával, vizsgálhatja a különböző kombinációk viselkedését. A rendszer lehetőségeit tekintve hasonlít az egyszerű programozási nyelvekhez, bár teljesen eltérő megközelítésből. A tapasztalat, az intuíció, a játék, a kísérletezés alapvető eszközei ennek a nyelvnek. A rendszerben való kísérletezés alatt a felhasználó számos fizikai és logikai rendszert megismerhet. Büntetlenül készíthet nem működő struktúrákat, hogy a hibákból tanulva újabb és újabb megoldásokat keressen problémáira.

Tág felhasználási terület

Egy kreatív eszköznél fontos, hogy kezdők és profik is egyaránt hasznosíthassák. A kezdők az egyszerű használat és a koherens belső logikai miatt hamar ismerősként érezhetik magukat a rendszerben. A szoftver alapját képező "kinetikus-nyelv" lehetőséget biztosít a bonyolult struktúrák létrehozására is. Az elemek variációinak száma tulajdonképpen végtelen. Egyszerű és összetett, apró és hatalmas szerkezetek megalkotására is alkalmas. Karakteres vizuális nyelve könnyen személyre szabható. A rendszer nyitott, szoftveren kívüli impulzusok befogadására. Ilyenek lehetnek a valós fizikai szenzorok jelei, kamerák, mozgásérzékelők, online adatok is. A rendszer professzionális média műalkotások elemeként, élő előadásokon, oktatási helyzetben, animációk készítéséhez, vagy gyors interaktív prototípusok előállítására is felhasználható.

Tézisek

1. Összevetve a hagyományos és a digitális írástudás képességét, észre kell, hogy vegyük, a mai kor emberei nagyban függenek az általuk használt digitális eszközök készítőitől.
2. Eltérően a nem digitális eszközöktől, a digitális eszközök fizikai test nélküli virtuális tárgyak. A hardver fizikai megjelenése szoftver tekintetében nem releváns. Ezt azt eredményezi, hogy működését kizárólag az interfészen keresztül ismerhetjük meg, felhasználási területük a tervező által korlátozott.
3. Egy fizikai eszköz formai kialakítása magában hordozza használatának lehetőségeit, működése átlátható, más tárgyakkal kombinálható, vagy akár módosítható is. A digitális eszközöknél tudatosan kell megteremteni a kombinálhatóságot, azaz a szabad kísérletezés lehetőségét, hogy ezáltal az egyéni kreativitás megjelenhessen.
4. Ha egy digitális eszköz alap-építőkövei nagyon általánosak, segítségükkel szinte bármilyen problémát meg lehet oldani, viszont használatuk bonyolulttá válik, elsajátításuk sok időt vesz igénybe. Ha viszont túl speciálisak, használatuk könnyű lesz, viszont a rendszer egészébe véve korlátossá válik. Mindig az adott cél érdekében kell megválasztani az építőköveket, megtalálni a megfelelő egyensúlyt.
5. A kreatív digitális eszközre jellemző a redukált jelkészlet és szabályrendszer. Az egyszerűség egyben korlátokat is teremt, ami ebben az esetben erény, hiszen a kreativitás legtöbbször a határok feszegetéséből ered.
6. A meglévő tudásra mindig könnyebb építeni. A jól megválasztott metaforák segítik a felhasználót a szoftver működésének értelmezésben, mivel ismert fogalmakat társítanak az új, ismeretlen gondolatokhoz.

Kivonat

Értekezésem a kreatív digitális eszközök tervezésének kérdéseit járja körül. A probléma abból adódik, hogy míg a nem digitális kreatív eszközök rendelkeznek fizikai testtel, a digitális eszközök tárgyi mivolta, habár jelen van (hardver), az adott szoftver működését nem reprezentálja. A kreativitás legtöbb esetben a fizikai tárgy, fizikai manipulálásával, vagy más tárgyakkal való kombinálásával jelenik meg. A fizikai eszközök formai kialakítása magában hordozza a használatának lehetőségeit, működésük átlátható, más tárgyakkal kombinálható, vagy akár módosítható is. A digitális eszközök működésére csak az interfészen keresztül lehet következtetni. Mivel a kreativitás legtöbbször a tárgyak, gondolatok meglepő, és egyedi egymáshoz rendeléséből ered, a digitális eszközökbe tudatosan kell beletervezni ennek a lehetőségét. Egy szoftver funkcióinak, alapelemeinek a kombinálása is lehetőséget nyújt a kreativitás megjelenésére. Erre a legjobb példa a programozás, ahol az alapelemek nagyon általános építőkövek. Segítségükkel szinte bármilyen probléma megoldható, ezek kombinációinak számossága végtelen. A digitális eszközök tervezésnél az alapelemek és a szabályrendszer megválasztása határozza meg az eszköz lehetőség-terét. Ha egy szoftver alap-építőkövei nagyon általánosak, segítségükkel szinte bármilyen problémát meg lehet oldani, viszont használatuk bonyolulttá válik, elsajátításuk sok időt vesz igénybe. Ha viszont túl speciálisak, használatuk könnyű lesz, viszont a rendszer egészébe véve korlátossá válik. Az építőkövek és a szabályok egyensúlyát, mindig az adott cél érdekében, az adott célközönség felmérése után kell kialakítani. Egy jó kreatív eszköz lehetőségeinek tárháza széles, használata viszont egyszerű és intuitív. A használatot nagyban segítik a metaforák alkalmazása. Ezek általában a fizikai világ jelenségeit azonosítják a szoftverek belső működésével. Segítségükkel a digitális eszközök használata megkönnyíthető, a felhasználó, már létező tudására építkezve, azt kiegészítve a virtuális eszközök lehetőségeivel a tanulási és megértési folyamat lerövidíthető. A kreatív eszközök ereje, nem a használatuk bonyolultságában, hanem az általuk megfogalmazható problémák összetettségében keresendő. Az egyszerűség egyben korlátokat is teremt, ami ebben az esetben erény, hiszen a kreativitás legtöbbször a határok feszegetéséből ered.

Bibliográfia

- CASTELLS, Manuel: The Rise of the Network Society, Wiley-Blackwell, 2000
- FRAZER, John: An Evolutionary Architecture. Nature as a basis for design, Wiley, 1999
- FULTON, Suri Jane: Thoughtless Acts: Observations on Intuitive Design, IDEO, 2005
- HAQUE, Usman: The Architectural Relevance of Gordon Pask. In: Architectural Design Magazine, Vol. 77, No. 4, London, 2007
- JOEY, J. Lee: Gamification in Education: What, How, Why Bother? (2011)
- KANGYAL, András, Lauffer László: Gépéret, L'Harmatta, 2011
- KILPATRICK, Jeremy: Helping Children Learn Mathematics
- KOESTLER, Arthur: "The Act of Creation" 1964
- KURTZWEIL, Ray: The Age of Spiritual Machines. Phoneix, London, 1999
- LAKOFF, George: The Contemporary Theory of Metaphor, 1992
- MAEDA, John: Creative Code. Thames & Hudson, Masseurchusetts, 2004
- MAEDA, John: Design By Numbers, MIT Press, 1999
- MAEDA, John: The Laws of Simplicity, MIT Press, London, 2006
- MÉRŐ, László: Észjárások, Budapest, 1989
- NADARAJAN, Gunalan: Islamic Automation: Al-Jazari's Book of Knowledge of Ingenious Mechanical Devices:
<http://www.muslimheritage.com/article/islamic-automation-al-jazari%E2%80%99s-book-knowledge-ingenious-mechanical-devices>
- NAGY, Ágoston: Összegyűjtött írások. <http://www.binaura.net/stc/wrx/text>
- NOBLE, Joshua: Programming Interactivity, O'Reilly Media, Sebastopol, 2009
- PANGARO, Paul: Cybernetics: <http://www.pangaro.com/>
breslin steve: the history and theory of sandbox:
http://www.gamasutra.com/view/feature/132470/the_history_and_theory_of_sandbox_.php?print=1
- POLLACK, B. Jordan: Computer Creativity in the Automatic Design of Robots, MIT Press, 2003
- RESNICK, Mitchel: Design Principles for Tools to Support Creative Thinking, MIT Press, 2005
- ROBINSON, Ken: Out of Our Minds: Learning to be Creative, 2011
- ROSZAK, Theodore : Az információ kultusza, avagy a számítógépek folklórja és a gondolkodás igaz művészete, Európa, 1990.
- SAFFER, Dan: The Role of Metaphor in Interaction Design, Pittsburgh, Pennsylvania, 2005
- SCHRAEFEL, M.C. : Making Tea: Iterative design through analogy , New York, 2004

- SERGI, Jordá: The reactable, Paper, Universitat Pompeu Fabra
- SHIFFMANN, Daniel: The Nature of Code: Simulating Natural Systems with Processing, 2012
- STERLING, Bruce: Shaping Things, MIT Press, 2005
- SZABÓ, Katalin: Metaphores and the user interface, 1995
- TILLMANN, J. A.: Merőleges elmozdulások. Palatinus Kiadó, Budapest, 2004
- WEINER, Norbert: „Cybernetics in History,” The Human Use of Human Beings, 1954. In: Multimedia, New York, 2002.

Képek forrásai

1. borító: Dürer mozgásban c. installáció
2. <https://annarzepczynski.wordpress.com/2012/02/05/thoughtless-acts-are-truly-thoughtful/>
3. <https://en.wikipedia.org/wiki/Reactable>
4. <http://minecraft.gamepedia.com>
5. https://en.wikipedia.org/wiki/Ismail_al-Jazari , <http://karakuri-tamaya.jp/en/knowledge.html>
6. <https://www.flickr.com/>

Kiállítások, művészeti megjelenések, installációk

2016

Flow. ROM, Budapest - generatív animáció

Írói fogások, installáció, Budapest - interaktív installáció idézetek megjelenítésére

Heródes Temploma, installáció, Biblia Múzeum, Budapest - interaktív játék, projekció épített makettre

Szintükör, installáció, Iparművészeti Múzeum, Budapest - látogatók ruháiból nyert színek alapján készített adatvizualizáció

2015

Kenyérszaporítás, installáció, Biblia Múzeum, Budapest - interaktív animáció, ami a látogatók kezéről készített fényképeket szervezi vizuális rendszerbe

TimeTable: Installáció, József Attila Emlékház, Budapest - interaktív információ pult és játék, József Attila életrajzának megjelenítésére és az idővel kapcsolatos gondolatainak ábrázolására

Dürer mozgásban: installáció. Biblia Múzeum, Gönc - interaktív installáció, Jézus életének bemutatása, Dürer fa-metszeteinek felhasználásával (Animata segítségével)

Az éjszaka csodái, színházi interaktív látvány, Tünet Együttes, Budapest

2014

Váratlan elágazások, egyéni kiállítás, Budapest - Binaura csoport retrospektív kiállítása (Nagy Ágostonnal)

Szombati Asztal, installáció, Néprajzi Múzeum, Budapest - Zsidó vacsora bemutatása asztalra vetített animáció formájában

Hamvas Béla kiállítás, animáció, Szentendre - Hamvas idézetekből készített animáció

Madách Retro, installáció, PIM, Budapest - Játék az Ember tragédiája c. könyv alapján

Shape Composer, MUPA, Budapest - A MÚPA orgonájához fejlesztett vizuális komponálást lehetővé tevő szoftver

Alpha, installáció, Érzékelés Határán, Műcsarnok, Budapest - Interaktív projekció és hang kövekre

2013

Interaktív kísérletek, installáció sorozat, A38, Budapest

Weöres100, interaktív installációk, PIM, Budapest - Weöres Sándor írásainak bemutatása játékos, interaktív formában

2012

Flow: SWEART, Olaszország - generatív animáció

2011

Relációanalízis. Szabad Laboratóriumi Műveletek2 , tánc-performenzs előadás
SphereTones: vizuális hangszer, - vizuális hangszer app, Google és iOS platformra

2010

Analóg akusztikus tér: Prototypen fesztivál, Dortmund
Kinetikus tér-animáció, szoftver, Magyar pavilon, Shanghai - Kinetikus animációt vezérlő szoftver és hang-tér kialakítása
audió-vizuális játékok, interaktív installációk, Sajtódíj átadó, Műcsarnok

2009

flow, interaktív belsőépítészeti felületek, Pannon Art Projekt, Budapest - generatív animáció
AV-manifest, installáció, A22 galéria, Budapest - egyéni kiállítás
live-act, (binaura projekt) Synoptic fesztivál, Pécs
Alpha, (binaura projekt) installáció, Relatív áthallások, Artus színház, Budapest - Interaktív projekció és hang kövekre

2008

Dream Project, interaktív installációk, Mercedes design verseny, Stuttgart
Metamorph, interaktív installáció, Art'n science in CMB, Berlin
Live-act, (binaura projekt) Ultrahang fesztivál, Budapest

2007

Strings, (binaura projekt) interaktív installáció, live-act, Audio art festival, Krakkó
Strings, (binaura projekt) interaktív installáció, New Music Expo, Műcsarnok, Budapest
GrafiT, (binaura projekt) elektro-akusztikus performenzs, Sziget fesztivál, Budapest
Animata, Opensource software of the year award, Pixelache fesztivál, Helsinki

2006

egyéni kiállítás, MMG, Budapest
live-act, (binaura projekt) Enter4 fesztivál, Székesfehérvár
Csend szoba, (binaura projekt) interaktív installáció, Sziget fesztivál, Budapest
Metamorph, videó performenzs, Taipei, Taiwan
VJ előadás, Rotterdam

2005

Mandala, (binaura projekt) interaktív installáció, Sziget fesztivál, Budapest

Thesis

1. When comparing digital writing to traditional writing, we must notice that people of the digital age are dependent on the tools they are using.
2. In contrast to non digital tools, digital tools lack physical body. Concerning the software the physical appearance is irrelevant. It results that the only way one can obtain information of the inner logic of the system is the interface itself. The fields of usage are strictly specified by the designer of the tool.
3. A physical object by itself carries the possibilities of its usage. Its structure is clearly observable, it is open for combinations with other objects, it is even modifiable. A software designer must create consciously the possibility of interconnectivity into the system, hence the opportunity for creativity to arise.
4. In a system where the primitive elements are general enough, it is possible to solve almost any kind of problems, therefore the usage becomes difficult and learning becomes time consuming. On the other hand, where elements are too specific, usage is easy, but the system in general becomes restricted. The designer must balance those components depending on the goals and the target audience.
5. Creative tools must have a reduced tool kit and rule system. While simplicity creates limits, creativity often arises by stretching them.
6. It is always easy to build upon an existing knowledge. Carefully chosen metaphors can help users to understand the underlying operating principles, because they associate known concepts with new and unknown ideas.

Abstract

This thesis is about the challenges of designing creative digital tools. The problem itself comes from the fact that while non-digital tools rely on their physical body, digital tools are bodiless. Creativity often arises from the physical manipulation and interconnection of objects. Since digital tools has no physical representation, the software designers must implement the ability of creativity into their tools. In a system where primitive elements are general enough, it is possible to solve almost any kind of problems, therefore the usage becomes difficult and learning becomes time consuming. On the other hand, where elements are too specific, usage is easy, but the system in general becomes restricted. The designer must balance those components depending on the goals and the target audience. A creative tool must be easy to use, and it should offer a broad field of usage. When designing a new tool, it is always easy to build upon an existing knowledge. Carefully chosen metaphors can help users to understand the underlying operating principles, because they associate known concepts with new and unknown ideas.