



■ ТОМ 3/БРОЙ 1/2022

Петко Танчев

ГЕНЕРАТИВНО СЪДЪРЖАНИЕ В ДИГИТАЛНОТО ИЗКУСТВО

Дигиталното изкуство съдържа разнообразни методи за създаване на съдържание. Настоящата статия описва един от най-експерименталните подходи – директното генериране на графика чрез автономни системи. В този случай художникът изразжда произведението изцяло чрез софтуерни и хардуерни инструменти, като програмен код, виртуални библиотеки, сензори и други. Той задава формални инструкции и алгоритми, на базата на които компютърът взема решения. Генерирането на съдържание радикализира функциите на машината като „мислещ инструмент“ готовкова, че понякога тя може да се счита дори за съавтор на творбата. В книгата „Генеративно изкуство. Практическо ръководство за Processing“, Мат Пиърсън пише:

Ролята на художника в този вид производствен процес е може би по-близка до ролята на куратора, отколкото на създателя. Той прави, моделира и усъвършенства системата, но в крайна сметка собствеността му върху крайния продукт не е нищо повече от гордостта на родителите при отглеждане на тяхното потомство (Pearson 2011, p. 4).

В тези идеи можем да открием много прилики с някои революционни художествени движения от началото и средата на ХХ век, като гага, флюксус и концептуалното изкуство. Още тогава те скъсват с класическото разбиране за изкуство, а произведението се превръща в естетически конструирана игра от предварително създадени условия. В основата на всичко стои случайността, разбиването на формата и пренареждането на елементите. През 1967 година американският художник Сол Лют разсъждава така:

Концептуалното изкуство не е непременно логично. Логиката на една творба или серия от творби е устройство, което се използва на моменти, само за да бъде съсипано. Логиката може да се използва за маскиране на реалното намерение на художника, за приспиване на зрителя във вратата, че той разбира произведението, или за извеждане на парадоксална ситуация (например логика срещу алогичност). Някои идеи биват логични в концепцията си и нелогични перцептивно (LeWitt 1967).

Десетилетия по-късно непредсказуемостта е все още важен елемент в творческите търсения на нашето съвремие и продължението на тази концепция е именно генеративното изкуство. Една от главите в началото на книгата на Мат Пиърсън е озаглавена „Художник на хаоса“. Там той дава следната дефиниция:

Да обявим работата си за „изкуство“ е смело и арогантно нещо. Правейки това, ние казваме, че нашата работа е извън обикновената полезност: тя е израз на нашата човечност и индивидуалност. (...) Нещо повече, ако искаме да определим своето произведение като „генеративно“ изкуство значи, че ние не само възнамеряваме то да изразява нашата индивидуалност, но също така искаме от него да показва и хаоса на процесите, върху които така или иначе нямаме контрол. Без тази непредсказуемост нашето изкуство би принадлежало на нещо друго, което не може да се обозначи като генеративно. Генеративните художници са художници на хаоса. Те породиха непредсказуемостта, приветстваха го, опитомиха го и вече могат да го представят в приятни форми. (...) Хаосът не е нещо, от което трябва да ни е страх. В крайна сметка, не правим нищо по-опасно от това да създаваме атрактивни изображения с компютри. Сериозно, кое е най-лошото, което може да се случи? Да бъдем въвлечени в генеративни процеси означава, че очакваме непредсказуемостта – то не е нежелан посетител. Ще ни стане дори комфортно с липсата на контрол върху работата ни. Ще приемем този хаос и ще се научим да го обичаме (Pearson 2011, p. 12).

Предстои да разгледам някои конкретни примери за генеративни инструменти в платформата за визуално програмиране TouchDesigner (наречан още Touch или TD), като се фокусирам върху техническите им функции. Една от основните концепции в този софтуер е конвертирането на данни между различни типове оператори. Това е работен принцип, който позволява на художниците да превръщат автоматично числовите стойности от CHOP и DAT операторите¹ в цветове, форми, точки, линии и повърхности. Компютърът сам изчислява и създава визуалното съдържание, а информацията се актуализира в

¹ CHOP и DAT са възприети като специфични технически съкращения, които означават съответно *Канални оператори* (от англ. CHOP или Channel Operators) и *Информационни оператори* (от англ. DAT или Data). Те представляват част от основните градивни частици на визуалното програмиране в приложението TouchDesigner и се отнасят към управлението и манипулацията на данни, числови стойности, компютърни кодове и други.



Визуално програмиране в платформата TouchDesigner, Проект Last Tear, Петко Танчев, 2015

реално време чрез CHOP to, DAT to и SOP to операторите.² Модулацията на крайния резултат зависи от входящите параметри, които могат да бъдат конструирани от анализ на жива музика, математически уравнения, данни от сензори и т.н. Така лесно могат да бъдат програмирани генеративни аудио-визуални системи и други интерактивни компоненти.

Един от операторите за създаване на неконвенционални геометрични обекти в Touch е LSystem SOP.³ Той се използва за имплементация на т.нар. L-системи или Линденмайер системи, кръстени на името на унгарския биолог Аристид Линденмайер (1925 – 1989). В тях се използва математически език, чийто първоначален низ от знаци се изпълнява многократно, а резултатите автоматично генерират геометрия. Всяко поредно изпълнение става основа за следващите итерации на формата, създавайки илюзия за растеж. L-системите могат да се използват за моделиране и анимация на органични разклоняващи се обекти, като растения, дървета, цветя, светкавици и снежинки. Цялата теория, която стои зад принципите на тези специфични обекти, е изчерпателно описана в книгата „Алгоритмичната красота на растенията“ от Пшемъслав Прушинкиевич и Аристид Линденмайер. Според двамата учени L-системите могат да бъдат много полезни практически в изграждането на

виртуална ботаническа лаборатория, която те описват така:

Виртуалната лаборатория, също като „истинската“, е площадка за експерименти. Тя се предлага с набор от обекти, свързани с нейната научна перспектива (в случая модели на растения), инструменти, които оперират с тези обекти, справочник и бележник. След като понятията и инструментите биват разбрани, потребителят може да разшири лабораторията чрез добавяне на нови обекти, създаване на нови експерименти и записване на описания в бележника. (...) Технически виртуалната лаборатория представлява микросвят, който може да бъде изследван под ръководството на хипертекстова система. Терминът „микросвят“ означава интерактивна среда за създаване и провеждане на симулирани експерименти. Ръководството може да бъде създадено и под формата на традиционна книга, но като електронен документ то би било по-подходящо за интеграция с микросвета (Prusinkiewicz, Lindenmayer 1996, pp. 194–195).

Ще обърна внимание и на друг TD оператор, който може да се използва като основа на сложни органични симулации. Това е Noise TOP⁴, чиято функция е да генерира автоматични вариации на

² Други оператори от същия вид, които служат за пренасяне и преобразуване на различни видове параметрични данни в програмата TouchDesigner.

³ Оператор за триизмерни обекти от фамилията Оператори за повърхности, наречена още SOP (съкращение от англ. Surface Operators). Пълното техническо описание на този оператор е налично в официалната документация на TouchDesigner на следния линк: https://docs.derivative.ca/LSystem_SOP

⁴ Оператор за създаване на двуизмерно визуално съдържание от фамилията Текстурни оператори, която се означава съкратено като TOP (от англ. Texture Operators). Неговото пълно описание е налично онлайн на следния линк: https://docs.derivative.ca/Noise_TOP

модели динамичен шум, като използва набор от алгоритми за управление на различни свойства на пикселите. Такива са например интерполацията, текстурните координати, хармоничните честоти и други. Резултатът може да бъде имплементиран в множество проекти, като например визуализации на флуидни графики. За да дам повече яснота относно употребата на Noise TOP, ще използвам цитат от книгата „Изкуството на флуидната анимация“, в която Джос Стам описва процесите по създаване на флуиди:

Целта в анимацията на флуидите е да се изчислят различни състояния на флуида във времето, които да бъдат превърнати в плоски двуизмерни изображения. Художниците манипулират и контролират свойствата на флуидите в сложното 4D пространство, а физиката автоматично се грижи за движението. Последният ред в уравнението е: как да стигнем от концепцията на художника за флуиден ефект до генериране на набор от пиксели на екрана (Stam 2016, p. 7).

Освен в семейството на TOP операторите, можем да открием Noise също така в CHOP и SOP. Съответно, когато използваме Noise CHOP, ние ще генерираме числови стойности, а Noise SOP ще действа като филтър за геометрия, променяйки позицията на точките според гаген модел шум. И двата оператора използват един и същ тип математика, която създава както гладки криви, така и произволен шум.

TouchDesigner предоставя взраден достъп и до още една изключително мощна технология за автоматично рисуване на сложни 2D и 3D форми. Това е т. нар. програмен език за дефиниране на шейдъри GLSL (OpenGL Shading Language).⁵ Шейдърите представляват програми в текстови формат, които позволяват на потребителя да комуникира директно с графичния хардуер на компютъра. Като описание на GLSL ще използвам една характеристика от популярното електронно издание „Книга за шейдърите“ на Патрисио Гонсалес Виво:

Ако вече имате опит в рисуването чрез компютър, знаете, че първо трябва да нарисувате кръг, след това правоъгълник, линия, няколко триъгълници, докато постепенно не изградите желаното от вас изображение. Този процес наподобява писането на писмо или книга на ръка – той е набор от инструкции, които вършат задачите една след друга. Шейдърите

те също са набор от инструкции, но инструкциите се изпълняват наведнъж за всеки един пиксел на екрана. Това означава, че кодът, който пишете, се държи различно в зависимост от позицията на самия пиксел. Подобно на печатарска преса, вашата програма ще работи като функция, която получава позиция и връща цвят, а когато бъде компилирана, тя ще функционира изключително бързо (Vivo, Lowe 2015).

За да обясня точно по какъв начин функционират шейдърите, ще навляза по-дълбоко в спецификата на работните им процеси. Тези програми използват т.нар. паралелна обработка на съдържанието (от англ. parallel processing), която има огромно значение за оптимизацията при скоростта на рендера⁶ в реално време. В „Книга за шейдърите“ е описано как работи тази технология по следния начин:

Поради архитектурата на компютрите задачите са принудени да се изпълняват в серия; всяко изчисление трябва да бъде завършено едно след друго. За да поддържат нивото си на производителност, съвременните компютри обикновено са снабдени с групи от четири процесора. Те работят като тръбопроводи, през които постоянно минават различни типове операции. (...) Видеоигрите и други графични приложения изискват много повече работна мощност от другите програми. Поради тяхното графично съдържание, те трябва да извършват огромен брой операции пиксел по пиксел. Всеки един пиксел на екрана трябва да бъде изчислен, а в 3D игрите трябва да се изчисляват също така и геометриите и перспективите. (...) Тук като подходящо решение идва паралелната обработка. Вместо няколко големи и мощни микропроцесори или тръбопроводи, по-интелигентно е тези задачи да се разпределят в множество на брой малки микропроцесори, които работят паралелно по едно и също време. Точно това представлява графичния процесор на компютъра (...) Другата негова „супер сила“ са специалните математически функции, които са хардуерно ускорени. Така сложните изчислителни команди се решават директно от микрочипове, вместо от софтуер. Следователно, процесорът е изключително бърз при тригонометричните и матричните операции – толкова бърз, колкото е скоростта на електричеството (Vivo, Lowe 2015).

⁵ GLSL или OpenGL Shading Language представлява специфичен вид език за програмиране, който се причислява към т.нар. C-езици. Той се използва основно за създаване на графични изображения и симулации в реално време, директно в хардуерната система. GLSL използва различни оптимизации на процесите, което го прави подходящ инструмент за генериране на сложни обекти. Програмите, написани на GLSL, се наричат с техническия термин шейдъри (от англ. shaders). Официалната документация на този език е налична онлайн на следния линк: https://www.khronos.org/opengl/wiki/OpenGL_Shading_Language

⁶ Рендер или рендеринг (от англ. rendering) е технически термин в компютърното програмиране, който означава процесът за създаване на специфично цифрово съдържание. При него гаген 2D или 3D модел се използва, за да се възпроизведе съответно изображение.

На практика тази техническа характеристика означава, че всяка промяна в параметрите и описването на нови функции се отразяват върху крайния резултат „на живо“ още в момента на кодиране. GLSL е технология за напреднали потребители, която дава достъп до ниско ниво на програмиране на графичната карта и използва по-продуктивни методи за генериране на съдържание. Това, заради което GLSL езикът е толкова мощно средство за рендер, е достъпът до неговите C++ библиотеки с графични ефекти и симулации.⁷ Независимо дали целта е 2D-стилизирана художествена абстракция или 3D динамично фотореалистично изображение, в света на шейдърите всичко е възможно. Качеството на рендерите зависи единствено от възможностите на програмиста и ограниченията на системните ресурси.

Днес съществуват много онлайн платформи, форуми и галерии за артистични експерименти с GLSL, като едни от най-известните са ShaderToy и Three.js. Те използват приложно-програмния интерфейс на WebGL⁸, за да компилират различни

интерактивни програми директно в браузъра. Обществото за тези платформи съществува още от началото на 70-те години, когато в дигиталната субкултура се заражда артистичното движение *Демосцена*. Неговите членове организирани публични сесии, т.нар. демопартита, в които според условията всеки участник трябва да програмира вариации на интерактивно съдържание, което да отговаря на предварително зададени условия. В тези презентации можем да открием и корените на техниката „кодиране на живо“ (от англ. live coding), която е все още актуална в много артистични среди и фестивали за дигитално изкуство. Не липсват и офлайн софтуерни платформи, които са създадени специално, за да покрият нуждите на този вид творческо програмиране. Една от най-популярните подобни апликации е KodeLife на Hexler. Според неговата философия ... *светът има нужда от бърз и мощен, мултиплатформен, мулти-API⁹ въграден инструмент, който да дава достъп до пълната, изначална мощност на графичния процесор, с цената на минимални усилия и разходи (Introduction · KodeLife | hexler.net [no date]).*

GLSL шейдър в KodeLife и програмния интерфейс на приложението Rouge

Аудиовизуален пърформанс на Петко Танчев, 2017



⁷ Специфични цифрови библиотеки, които се отнасят към програмния език C++. Те представляват пакети с функционален код, чиято цел е да бъдат многократно използвани от различни приложения.

⁸ Софтуерен интерфейс, който се основава на отворения графичен език OpenGL.

⁹ Приложно-програмен интерфейс (от англ. Application programming interface, съкратено API).

Тук ще обърне внимание на видовете шейдъри в TouchDesigner и практическото им приложение. Техният синтаксис се съдържа в Text DAT оператори.¹⁰ Всички графични операции с двуизмерни изображения се изпълняват в GLSL TOP.¹¹ Този оператор представлява шейдър, приложен върху четириъгълник с размерите на видимото работно поле. Кодът на шейдъра в GLSL TOP се стартира по веднъж за всеки изобразен пиксел. Той се компилира като пиксел шейдър или като по-сложния изчислителен шейдър.¹² Докато пиксел шейдърът само оцветява пикселите, като записва еднократно резултатите в буфери за цвят и височина без да може да прочита резултатите на съседните изходящи пиксели, концепцията на изчислителния шейдър е по-различна. Той няма ограничен набор от входове и изходи, може да извършва повече изчисления върху всички микропроцесори в графичната карта едновременно и да записва стойности на произволни места. За сметка на това, при него съществуват повече ограничения на производителността, които за висят най-вече от особеностите на хардуера. Синтаксисът на изчислителните шейдъри е сравнително по-сложен за писане и се поддържа само в GLSL версия 4.30 или по-нова.¹³

Потребителят може да използва шейдъри и като персонализирани материали за триизмерни обекти. В този случай той дефинира кода в GLSL MAT оператор¹⁴, а програмата се разделя на два основни компонента: върхни и пикселни шейдъри (от англ. vertex and pixel shaders). Подобно на първата ситуация с двуизмерните шейдъри, в GLSL MAT те също се прилагат към всеки връх на геометрията и към всеки видим пиксел на екрана. Върхните шейдъри трансформират върховете на обектите, а пиксел шейдърите ги растеризират. Допълнително съществува и още един вид геометрични шейдъри, които работят като междинни етапи между върховете и пикселите. Те взимат единични точки от примитивите, линиите или триъгълниците и връщат набор от точки, линейни или триъгълни стрипове.¹⁵

Какви са преимуществата на GLSL технологията пред останалите вградени оператори в TouchDesigner? На първо място, както вече стана дума, тези програми се изпълняват винаги с по-висока скорост, защото всички процеси се пресмятат в графичния процесор, освобождавайки по този начин повече памет в централния процесор на компютъра. На практика, когато гаген проект в TD използва само оператори за генериране и управление на съдържание, системните ресурси могат бързо да бъдат заети поради спецификата на медийната архитектура. Възможностите за оптимизация в този случай са ограничени от самите оператори, защото те представляват затворени в себе си програми. Например SOP операторите за моделиране на геометрия натоварват бързо централния процесор. Ако GPU или CPU паметта¹⁶ бъде напълнена, потребителят не може да направи много извън вградените методи за олекотяване на процесите. Тук на помощ идват шейдърите. Освен че са оптимизирани за графика в реално време, те дават достъп до по-ниски нива за програмиране на GPU, което позволява и повече контрол върху цялата система. Също така синтаксисът им позволява добавяне на всякакви персонализирани функции, които липсват в параметрите на вградените оператори. Пример за такива са мултикомпозициите на множество текстури, различните стилове филтри за постобработка на видео, материалите за деформация на геометрични обекти и т.н. Накратко, ползата от GLSL в Touch е в по-гъвкавата оптимизация на рендер системите и разширяването на възможностите за генериране на динамично съдържание.

След като се запознахме с организацията на работните процеси при шейдърите, нека разгледаме и някои конкретни проекти, базирани на тези програми. Като пример ще дам френския художник Винсент Хозе, който е известен в TD обществото със своите сложни и динамични GLSL симулации. В мащабната инсталация „Същества“ той изгражда един абстрактен свят, който репродуцира богатството на природните елементи

¹⁰ Информационни оператори в TouchDesigner, чиито функции служат за въвеждане на програмни скриптове и друго текстово съдържание.

¹¹ Текстурен оператор, който съдържа синтаксис с GLSL код. Неговата официална документация е налична на следния линк: https://docs.derivative.ca/GLSL_TOP

¹² Приложеното техническо описание на GLSL TOP е отчасти взаимствано от статията „Писане на GLSL TOP“, налична в официалната TD документация на следния линк: https://docs.derivative.ca/Write_a_GLSL_TOP

¹³ Цялото описание на споменатите функции е налично в статията „Изчислителен шейдър“, линк: https://docs.derivative.ca/Compute_Shader

¹⁴ Оператор за 3D геометрични обекти, част от фамилията Оператори за материали в TouchDesigner (от англ. Material Operators или съкратено MAT).

¹⁵ Приложеното техническо описание е отчасти взаимствано от статията „Писане на GLSL материал“, налична в официалната Touch документация на следния линк: https://docs.derivative.ca/Write_a_GLSL_Material

¹⁶ Памет на графичния и централния процесор в компютъра (от англ. Graphics processing unit или съкратено GPU и Central processing unit или съкратено CPU).



Снимка от проект *Съществува*, Винсент Хозе, 2018

по своеобразен начин. Реалистичните триизмерни обекти и движения са програмирани изцяло чрез персонализирани шейдър оператори, за да отговорят максимално точно на търсенията на художника. Според неговото описание:

... Проектът е „ода за разнообразието от форми на живот на земята. Инсталацията изследва различни полуабстрактни екосистеми – от микроскопични до макроскопични, изградени от алгоритмични пейзажи, растения и същества, чието комбинирано просто поведение поражда възникваща сложност. „Съществува“ е потапящо аудио-визуално изживяване с интерактивни графики и пространствен звук, които се създават от взаимодействието си в реално време. Проектът е осъществен за първи път през 2018 като авторски пърформанс в Дружеството за изкуства и технологии¹⁷ в Монреал и като инсталация през 2019 на изложението Майнпарк в Шенжен (Hoze 2018a).

В друг свой проект, наречен „Флуидна структура 360“, Хозе използва инфрачервени сензори и шейдъри в Touch, за да програмира интерактивна флуидна форма, която взаимодейства на живо с телата на зрителите, движещи се в залата. Визуализацията е изпълнена като 360-градусова прожекция, а премиерата се състои през 2018 година в Културен център „Ла Гите Лирик“ в Париж. Авторът дефинира проекта си така:

Това е потапяща интерактивна инсталация, която изследва как една ефимерна и аморфна форма реагира на различни стимули, вътрешни и външни. Силите и сблъсъците огъват формата, докато не се счупи, като я рекомбинират в нови агрегати. Резултатът е постоянно променящ се пейзаж, мистериозен, но в същото време познат. Драматичната визуализация подчертава вътрешната структура на формата и нейното движение. Използвайки компютърно зрение¹⁸, аудиторията става неразделна част от процеса, оставяйки временния си физически отпечатък, който в

¹⁷ На англ. Society for Arts and Technology (SAT).

¹⁸ От англ. computer vision.

крайна сметка винаги постепенно изчезва. Системата се задвижва от стилизиран флуиден разтворител, способен да процесира в реално време силите и ограниченията, на които е подложена формата (Houzé 2018b).

Работните методи за създаване на генеративно съдържание размиват границите между изкуство, технология и наука, а палитрата от инструменти в платформи като TouchDesigner дава на художниците повече творческа свобода за експерименти. TD операторите и възраженият

достъп до GLSL шейдъри предоставят нови възможности за създаване на компютърни произведения чрез комбинация от визуално програмиране и креативно кодиране. Освен това шейдърите представляват добра среда за оптимизация на процесите в изчислителната система. Допълнителните разширения, в съчетание с високата интероперативна съвместимост и останалите характерни качества на Touch, определят този софтуер като мултифункционален и универсален инструмент на генеративното изкуство.

БИБЛИОГРАФИЯ

Compute Shader – Derivative, 2018. *TouchDesigner by Derivative*. Online. [Accessed 15 February 2019]. Available from: https://docs.derivative.ca/Compute_Shader

HOUZÉ, Vincent, 2018a. Vincent Houzé and Dave & Gabe – Créatures – Vincent Houzé. Online. 2018. [Accessed 9 November 2019]. Available from: <https://vincenthouze.com/portfolio/creatures/>

HOUZÉ, Vincent, 2018b. Fluid Structure 360 – Vincent Houzé. Online. 2018. [Accessed 10 November 2019]. Available from: <https://vincenthouze.com/portfolio/fluid-structure-360/>

Introduction KodeLife | hexler.net, [no date]. Online. [Accessed 2 February 2020]. Available from: <https://hexler.net/kodelife/manual/introduction>

LEWITT, Sol, 1967. Paragraphs on Conceptual Art. *Artforum*. Online. Summer 1967. Vol. 5, no. 10. [Accessed 28 July 2022]. Available from: <https://www.artforum.com/print/196706/paragraphs-on-conceptual-art-36719>

PEARSON, Matt, 2011. *Generative Art: A Practical Guide Using Processing*. Shelter Island, NY: London: Manning; Pearson Education [distributor]. ISBN 978-1-935182-62-7.

PRUSINKIEWICZ, Przemysław and LINDENMAYER, Aristid, 1996. *The Algorithmic Beauty of Plants*. New York Berlin Paris [etc.]: Springer-Verlag. The virtual laboratory. ISBN 978-0-387-94676-4.

STAM, Jos, 2016. *The Art of Fluid Animation*. Boca Raton, Florida: CRC Press. ISBN 978-1-4987-0020-7.

TouchDesigner Documentation – GLSL TOP, [no date]. *TouchDesigner099 Documentaion*. Online. [Accessed 16 February 2019]. Available from: https://docs.derivative.ca/GLSL_TOP

TouchDesigner Documentation – LSystem SOP, [no date]. *TouchDesigner099 Documentaion*. Online. [Accessed 14 January 2019]. Available from: https://docs.derivative.ca/LSystem_SOP

TouchDesigner Documentation – Noise TOP, [no date]. *TouchDesigner099 Documentaion*. Online. [Accessed 5 February 2019]. Available from: https://docs.derivative.ca/Noise_TOP

VIVO, Patricio Gonzalez and LOWE, Jen, 2015. *The Book of Shaders*. The Book of Shaders. Online. 2015. [Accessed 10 January 2019]. Available from: <https://thebookofshaders.com/01/>

Write a GLSL Material – Derivative, [no date]. *TouchDesigner by Derivative*. Online. [Accessed 16 February 2019]. Available from: https://docs.derivative.ca/Write_a_GLSL_Material

Write a GLSL TOP – Derivative, [no date]. *TouchDesigner by Derivative*. Online. [Accessed 16 February 2019]. Available from: https://docs.derivative.ca/Write_a_GLSL_TOP
