

Program zajęć Creative Coding

Program na pierwsze 2.5 tygodnia zajęć. Blok Creative Coding.

Materiały do pobrania na stronie:

<https://github.com/CreativeCodingPL/CreativeCoding>

Tydzień 01

środa część I :: Kod tworzywem projektanta

Prezentacja: Kod tworzywem projektanta

- Mieszanie bitów i atomów.
- Interfejsy namacalne i idea radical atoms
- Każdy może kodować!
- Szkoła i efekt wyuczonej bezradności
- https://pl.wikipedia.org/wiki/Wyuczona_bezradność
- Kod w sztuce, nauce, designie, społeczeństwie
- Embodiment, Embodied interaction design
- Przykłady instalacji i obiektów.

wprowadzone polecenia, typy, zmienne

- point, line, ellipse, rect, triangle
- kolory: RGB, G, #, RGBA
- stroke, fill, strokeWeight, noFill
- setup, draw
- mouseX, mouseY

Materiały pomocnicze

- Color: <https://processing.org/tutorials/color/>
- Coordinate System and Shapes:
<https://processing.org/tutorials/drawing/>

Zadanie domowe

Proszę obejrzeć: <http://hello.processing.org>

środa część II :: Ruch

- zmienne
- random
- saveFrame, ####.tif
- keyPressed
- typy zmiennych float, Pimage
- “aaa” znaczy String
- if
- noise

- ruch browna
- loadImage, image

Materiały pomocnicze

- Interactivity:
 - <https://processing.org/tutorials/interactivity/>
- Noise i ruch browna:
 - <http://natureofcode.com/book/introduction/>
 - <https://vimeo.com/58492076>
- Instrukcja jak zrobić film z processing:
 - <https://youtu.be/G2hI9XL6oyk>

Zadanie domowe

Proszę zrobić film w Movie Maker z animacji poklatkowej zapisanej za pomocą saveFrame. Jak zrobić film z processing:

<https://youtu.be/G2hI9XL6oyk>

Czwartek część I :: Plakat vol. 1 - generatywna grafika wektorowa

- wprowadzenie, inspiracje z pinterest:
<https://pl.pinterest.com/jkcozniewski/cc-generative-poster/>
- dist
- własne kształty: curveVertex, vertex, beginShape, endShape(CLOSE)
- wadowanie grafiki wektorowej z pliku: PShape, loadShape(), shape()
- zapisywanie do PDF: beginRecord(), endRecord(), biblioteka PDF

(zainstalowana w porcessing na stałe) ale import w kodzie musi być

- `image.get(x,y)`, `brightness()`, `hue()`, `saturation()`

Materiały pomocnicze

- <https://pl.pinterest.com/jkozniewski/cc-generative-poster/>
- Krzywe: <https://processing.org/tutorials/curves/>

Czwartek część II :: Plakat vol. 2 - generatywna typografia

- podstawy: `Strign`, `translate`, `rotate`, `scale`
- co można zrobić ze string: sklejanie (+), cięcie `substring()`, wybieranie i zamienianie liter `charAt()`, znak czyli typ `char`, liczenie liter inaczej długość tekstu `.length()`
- wypisywanie wartości zmiennych w oknie terminala: `println()`
- mieszanie grafiki z tekstem polecenie `text()`
- modyfikacje działania polecenia `text()`: ustawianie fontu `textFont()` i zmienna `PFont`, rozmiar tekstu `textSize()`, wyrównywanie `textAlign(CENTER)`; `textAlign(LEFT)`; `textAlign(RIGHT)`, długość wypisanego tekstu w pixelach `textWidth()`
- biblioteka `geomorative`: instalacja, zamiana tekstu na krzywe `RG.getText()` i zmienna kształtu `RShape`, rysowanie kształtu `.draw()`, podział kształtu na kształty składowe `.children[]` np tekstu na litery, geometryczny środek kształtu `getCenter()` i zmienna `RPoint`, obrót kształtu `.rotate()`, pobieranie listy punktów leżących na krzywych kształtu `.getPoints()`, ustalenie równomiernego sposobu rozłożenia pobieranych punktów `RG.setPolygonizer(RG.UNIFORMLENGTH)`, ustawienie odstępu pomiędzy punktami dla równomiernego

rozkładu punktów `RG.setPolygonizerLength()`

- Samodzielna eksploracja przykładów kodu rysujących w oparciu o pobrane punkty. Przykłady s05e05 do s05s14.

Materiały pomocnicze

- Strings and Drawing Text:
 - <https://processing.org/tutorials/text/>
- Type + Code: Processing For Designers:
 - https://issuu.com/jpagecorrigan/docs/type-code_yeohyun-ahn
- BEYOND PRINT: <http://www.yeoahn.com>

Praca domowa

Stworzyć plakat łączący wektorową grafiką generatywną z typografią generatywną. Zapisać do pdf i przynieść na zajęcia wydrukowany :D

Tydzień 02

**środa część I :: Symulacja systemów
naturalnych vol. 1 - rekurencja i
automaty komórkowe**

- Wektory i co można z nimi robić PVector.
- Tworzenie własnych poleceń (fachowo pisanie funkcji) i wkładanie ich w zakładki.
- “Sadzenie drzew” :), czyli o naśladowaniu natury i tworzeniu obrazów samo podobnych (fraktale :)). Drzewo rysuje drzewo, które rysuje drzewo itd. (fachowo zwane rekurencją).
- “Szaleństwa Wolframa” (jak nie znacie to zerknijcie na wolfram alpha), czyli automaty komórkowe. Na początek jednowymiarowe, czyli typowa “ulicówka” i plotkujący sąsiedzi.

dokończyć z poprzednich zajęć

- wadowanie grafiki wektorowej z pliku: PShape, loadShape(), shape()
- image.get(x,y), brightness(), hue(), saturation()

Materiały pomocnicze

- Automat komórkowy: <http://www.wolframalpha.com/input/?i=rule+110>
- PVector: <https://processing.org/tutorials/pvector/>

Praca domowa

Proszę posadźcie własne piękne drzewo, a najlepiej cały las!

środa część II :: Symulacja systemów naturalnych vol. 2 - systemy agentowe

- Klasy, obiekty, metody i atrybuty.
- Tworzenie własnych klas.
- Symulowanie na wektorach oddziaływania sił: prędkość, przyspieszenie, grawitacja, odbicia, podążanie do celu.
- Systemy cząsteczkowe.
- Zachowania stadne.

Wprowadzone polecenia

Class, new, ArrayList, ArrayList.size(), lerpColor(), PVector.sub(), PVector.limit(), PVector.heading()

Materiały pomocnicze

- Autonomous Agents <http://natureofcode.com/book/chapter-6-autonomous-agents/>
- Objects <https://processing.org/tutorials/objects/>
- Vectors <http://natureofcode.com/book/chapter-1-vectors/>

Praca domowa

Przeróbcie przykład so8eo5Arrive tak, żeby wiele obiektów podążało do jednego celu. Podpowiedź: do rozwiązania posłużcie się przykładem

so8eo3arrayListFor.

Czwartek część I :: Przetwarzanie danych

- wczytywanie i zapisywanie danych w plikach tekstowych.
- wczytywanie i przetwarzanie danych zapisanych w formacie JSON
- pobieranie danych online: YahooWeather, Facebook, Instagram

Lista wprowadzonych poleceń i typów

- String: split(), join(), append(), saveStrings(), loadStrings()
- JSON: JSONObject, loadJSONObject()

Materiały pomocnicze

- Data: <https://processing.org/tutorials/data/>
- JSON Introduction:
http://www.w3schools.com/js/js_json_intro.asp

Praca domowa

Zmienić kod so6eo3FileSaveLoad1 tak żeby oprócz pozycji i rozmiaru kółka zapisywana była jakaś dodatkowa, np. kolor, nazwa itp.

Czwartek część II :: Dźwięk i tablice

- int - typ zmienne całkowite
- for - pętla, powtarzanie określoną liczę razy, przechodzenie przez tablicę, zagnieżdżanie czyli pętle w pętli
- array - zmienna tablicowa, odwoływanie się do elementów tablicy poprzez indeks, tworzenie tablicy na podstawie listy wartości, tworzenie tablicy o zadanym wymiarze poprzez operator new
- new - operator tworzenia nowego elementu o złożonej strukturze np. `int[] dane = new array[100];`
- `pixels[]` - bezpośredni dostęp do pixeli w grafice bitmapowej.
- Instalowanie bibliotek
- Biblioteka Sound
- `map()` - mapowanie zakresu wartości
- SinOsc - generator fali dźwiękowej o przebiegu sinusoidalnym.

Materiały pomocnicze

- Arrays
 - <https://processing.org/tutorials/arrays/>
- Sound
 - <https://processing.org/tutorials/sound/>
- Images and Pixels
 - <https://processing.org/tutorials/pixels/>
- Do pracy domowej jeszcze to:
 - [https://pl.wikipedia.org/wiki/Dźwięk_\(muzyka\)](https://pl.wikipedia.org/wiki/Dźwięk_(muzyka))
 - http://www.fizykon.org/muzyka/muzyka_tabela_czestotliwosci_tonow.htm

Zadanie domowe

Melodia odgrywana z danych w tablicy

Przykładowe rozwiązania znajdziecie w katalogu “prace domowe rozwiązania”.

Tydzień 03

środa część I :: Epilog :: Generative Art (koniec CC)

- Przetwarzanie wideo na żywo (Capture).
- Kodowanie na żywo na przykładzie biblioteki Voronoi. “Ręczna” instalacja biblioteki.
- Jak modyfikować kod nawet jak nie rozumiemy do końca jak to działa :). na przykładzie Reaction diffusion i Curl Vector Field with Noise.

Wprowadzone polecenia

warunek?wartość1:wartość2, image.get(x,y), brightness(), hue(), saturation(), biblioteka processing.video: Capture, read(), start().

Materiały pomocnicze

- Reaction-Diffusion Tutorial: <http://www.karlsims.com/rd.html>
- https://en.wikipedia.org/wiki/Reaction-diffusion_system
- the idea of the curl of a vector field:
http://mathinsight.org/curl_idea
- [https://en.wikipedia.org/wiki/Curl_\(mathematics\)](https://en.wikipedia.org/wiki/Curl_(mathematics))

Duża praca domowa :: Wystawa Creative Coding

Przygotować pracę na wystawę creative coding.

Jest to podsumowanie waszych umiejętności po ukończeniu warsztatu Creative Coding.

Co Was zainspirowało? Jakie nowe pomysły pojawiły się po tym warsztacie? Niech podsumowanie tego będzie większa praca, którą będziemy mogli pokazać światu :D

Wymagania do pracy

Nieprzekraczalny termin oddania pracy to **07.11.17**.

Praca powinna składać się z dwóch części:

1. Plakat lub grafika generatywna dostarczona w pliku PDF do druku w formacie B2 do zamieszczenia na wystawie.

2. Wideo w formacie MP4 720p 16:9. Wideo pokazuje proces generowania grafiki/plakatu kodem. Może to być seria zdjęć pokazujących wiele wariantów wygenerowanych kodem, z których jeden został wybrany na plakat/grafikę. Jeżeli grafika powstaje w wyniku symulacji zjawiska (np. ruch stada, wzrost drzewa, rozwój automatu komórkowego itp.) procesu ciągłego budującego końcową grafikę, wtedy wideo powinno zawierać zapis tego procesu. I mogą to też być warianty mieszane np. ciąg zdjęć wariantów przepleciony z poszukiwaniem przez uruchamianie symulacji z różnymi parametrami, aż w końcu pojawi się ten upragniony efekt ;). Super wygląda to tu: <https://vimeo.com/17142501>

Instrukcja jak zrobić film z processing: <https://youtu.be/G2hI9XL6oyk>

Jeżeli wolicie, można też nagrać zawartość ekranu i dźwięk za pomocą innego narzędzia np.:

- <https://www.monosnap.com/>
- <http://osxdaily.com/2010/11/16/screen-recorder-mac/>

Inspiracje do pracy końcowej

- Simulation and Nature in Design: <http://n-e-r-v-o-u-s.com/education/simulation/index.php>
- Generative Design: <http://www.generative-gestaltung.de/code>
- Studio A N F: <http://anf.nu>
- Generative Art: <https://youtu.be/cKzK88y3o1Q>
- Casey Reas (współtwórca Processing)
<http://www.bitforms.com/artists/reas> <http://reas.com>
- onformative: <http://www.onformative.com/work/actelion-imagery-wizard/>
- Quayola: <http://www.bitforms.com/artists/quayola>

<http://www.quayola.com>

- Canogar: <http://www.bitforms.com/artists/canogar>
- OpenProcessing: <https://www.openprocessing.org>
- Marcin Ignac: <http://marcinignac.com> <http://variable.io>
- Joanie Lemercier: <http://joanielemercier.com>
- Marius Watz: <http://mariuswatz.com>
- Daniel Shiffman: <http://shiffman.net>
- Anders Hoff: <http://inconvergent.net>
- Joshua Davis: <http://www.joshuadavis.com>
- Reza Ali: <http://www.syedrezaali.com>
- Generative Typography by Kyuha Shim:
<https://vimeo.com/138371379>