

Projekt Grafika Komputerowa

100 skrótów klawiszowych Blender

1. Temat projektu

Tematem projektu jest *100 skrótów klawiszowych w programie Blender*.

2. Omówienie

Doszedłem do wniosku, że aby nadać projektowi bardziej indywidualny wyraz wykonam go w postaci strony internetowej, na której przy użyciu prostego menu będzie można filtrować skróty według sekcji, których dotyczą.

3.Skróty klawiszowe

Ogólne(Okna):

1.Shift + F2

Otwiera okno Movie Clip.

2.Shift + F3

Otwiera okno Nodes.

3.Shift + F4

Otwiera okno Python Console.

4.Shift + F5

Otwiera okno 3D Viewport.

5.Shift + F6

Otwiera okno Graph.

6.Shift + F7

Otwiera okno Properties.

7.Shift + F8

Otwiera okno Video Sequencer.

8.Shift + F9

Otwiera okno Outliner.

9.Shift + F10

Otwiera okno UV/Image.

10.Shift + F11

Otwiera okno Text Editor.

11.Shift + F12

Otwiera okno Dope Sheet.

Ogólne(Object Mode):

12.G + X

Pozwala na przesuwanie po osi X.

13.G + Y

Pozwala na przesuwanie po osi Y.

14.G + Z

Pozwala na przesuwanie po osi Z.

15.X

Usuwa obiekt.

16.H

Ukrywa obiekt.

17.Alt + H

Odkrywa wszystkie ukryte obiekty.

18.S + a następnie X/Y/Z

Pozwala na skalowanie obiektu.

19.R i RR

Pojedyncze kliknięcie pozwala na obrót z perspektywy jaką patrzymy na obiekt, a podwójne na swobodne obracanie obiektu wzdłuż wszystkich osi.

20.R + X

Obrót wzdłuż osi X.

21.R + Y

Obrót wzdłuż osi Y.

22.R + Z

Obrót wzdłuż osi Z.

23.Alt + G

Reset przesunięcia obiektu.

24.Alt + R

Reset rotacji obiektu.

25.Alt + S

Reset skalowania obiektu.

26.Ctrl + +/-

Zoom kamery lub jej oddalenie.

27.Ctrl + N

Otwiera panel tworzenia nowego projektu.

28.Ctrl + O

Otwiera panel otwierania projektu.

29.Ctrl + strzałki przesuwania obiektu

Obiekt będzie przesuwany się skokowo.

30.Ctrl + 0/1/2/3/4/5

Zmiana poziomu Subdivision.

31.Ctrl + C i Ctrl + V

Skopiowanie i następne wklejenie obiektu.

32.Ctrl + J

Połączenie zaznaczonych obiektów w jeden.

33.ŚPM

Obrót widoku.

34.Shift + ŚPM

Przesuwanie swobodne kamery wzdłuż osi X i Y ekranu.

35.Shift + A

Otwiera panel dodawania obiektów.

36.Shift + ~

Włączenie tryby „latania” kamery.

37.N

Włączenie okna właściwości.

38.Shift + B

Narzędzie do zoomowania kamery.

39.Shift + H

Ukrycie wszystkich obiektów poza zaznaczonymi.

40.Shift + D

Duplikowanie obiektu i automatyczne włączenie jego swobodnego przemieszczania po wszystkich osiach.

41.Shift i przeciągnięcie LPM

Zaznaczenie obiektów w obszarze jego działania.

42.T

Pokazania/Schowanie Toolbara.

Edit Mode:

43.1

Tryb manipulacji wierzchołkami.

44.2

Tryb manipulacji krawędziami.

45.3

Tryb manipulacji ścianami.

46.Ctrl + B

Dodanie otaczających krawędzi do zaznaczonej.
Skantowanie krawędzi.

47.Ctrl + E

Otworzenie panelu krawędzi.

48.Ctrl + F

Otworzenie panelu ścian.

49.Ctrl + V

Otworzenie panelu wierzchołków.

50.Ctrl + T(w trybie edycji ścian)

Podział ściany na trójkąty.

51.C

Tryb zaznaczania kołowego.

52.M

Złączenie, które w zależności od trybu, w którym pracujemy(Wierzchołki, Krawędzie , Ściany) połączy je ze sobą i zastąpi jednym.

53.O

Włączenie/Wyłączenie trybu proporcjonalnego.

54.E

Ekstrakcja zaznaczenia.

55.F

Wypełnienie pomiędzy zaznaczeniami.

56.K

Tryb Noża, który pozwala ciąć ściany dodając do nich krawędzie.

57.P

Wyeksportowanie zaznaczenia do nowego obiektu.

58.S

Skalowanie zaznaczenia.

59.Shift – N

Przeliczenie normalnych.

60.Shift + W

Pochylanie zaznaczenia.

61.S + X

Skalowanie wzdłuż osi X.

62.S + Y

Skalowanie wzdłuż osi Y.

63.S + Z

Skalowanie wzdłuż osi Z.

64.TAB

Włączenie/Wyłączenie trybu edycji.

65.V

Rozerwanie obiektu w miejscu zaznaczenia.

66.Y

Oderwanie zaznaczenia od obiektu.

Sculpting:

67.Shift + Space

Włączenie panelu do wyboru typu pędzla.

68.X

Pędzel rysujący.

69.F

Edycja rozmiaru pędzla.

70.Shift + F

Edycja siły pędzla.

71.B

Dodawanie maski(box).

72.M

Pędzel maskujący

73.Alt + M

Wyczyszczenie maski.

74.E

Kontrola metody działania pędzla.

75.Shift + C

Tryb żłobienia.

76.Shift + T

Tryb wygładzania.

77.Shift + H

Ukrywanie obiektu.

Image Editor:

78.Alt + N

Otwarcie panelu tworzenia nowego obrazu.

79.Alt + O

Otwieranie nowego obrazu.

80.N(w trybie View).

Okno właściwości.

81.F

Edycja rozmiaru pędzla.

82.N(w trybie Paint)

Okno właściwości pędzla.

83.S

Pobranie koloru.

84.Shift + F

Edycja siły pędzla.

85.X

Zmiana pomiędzy kolorami(pierwszoplanowym i drugoplanowym).

86.Shift + Space

Okno wyboru trybu.

87.Shift + Alt + S

Zapis jako nowy obraz.

Dope Sheet:

88.Space

Włączenie/Wyłączenie odtwarzania animacji.

89.I (W oknie Layout)

Otwarcie panelu dodawania klatki kluczowej.

90.M

Dodanie znacznika do animacji.

91.Ctrl + M (z zaznaczoną klatką kluczową)

Panel odbijania lustrzanego klatki kluczowej.

92.Alt + ŚPM

Przewijanie animacji.

93.Strzałki Góra/Dół

Przeskakiwanie między klatkami kluczowymi.

94.Strzałki Lewa/Prawa

Przeskakiwanie między klatkami rzeczywistymi.

95.Ctrl + Strzałki Lewa/Prawa

Przeskakiwanie między pierwszą, a ostatnią klatką kluczową.

96.Ctrl + M(z zaznaczonym markerem)

Edycja znacznika.

97.Ctrl + Shift + Space

Odtwarzanie animacji od tyłu.

Render Mode:

98.F12

Renderowanie sceny

99.Ctrl + B

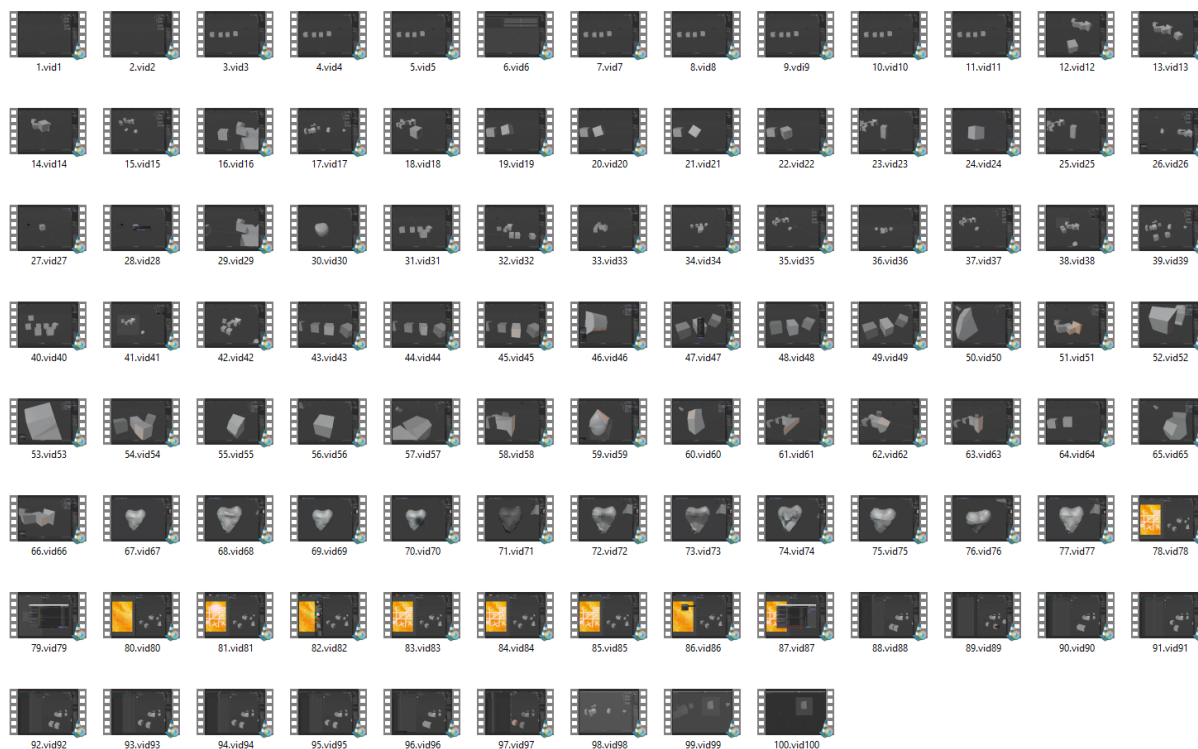
Wyznaczanie obszaru do wyrenderowania.

100. Ctrl + Alt + B

Usuwanie zaznaczenia obszaru do renderowania.

4. Wykonanie

Na sam początek zacząłem od nagrania 100 krótkich filmów, które przedstawiają działanie skrótów w programie.



Potem opisałem ich działanie w oddzielnym pliku tekstowym.

Gdy miałem już gotowe fundamenty do zbudowania strony przeszedłem do jej napisania.

Zrobiłem plik HTML i dodałem do niego style.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, initial-scale=1.0">
  <title>Projekt Blender | Skróty klawiszowe</title>
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link href="https://fonts.googleapis.com/css2?family=Antonio:wght@100;400;700&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="styl1.css">
  <script src="https://kit.fontawesome.com/b44de5aee5.js" crossorigin="anonymous"></script>
</head>

<body>
  <div class="wrap">
    <h1>Skróty Klawiszowe</h1>
    <div class="nav"><button>Test</button></div>
    <div class="skroty">
      <div class="skrot wszystkie">
        <h2>1. Shift + F2</h2>
        <div class="image"></div>
        <p>Otwiera okno Movie Clip.</p>
      </div>
    </div>
    <div class="nocny">
      <i class="fas fa-moon"></i>
    </div>
    <footer>Michał Mickiewicz 2021</footer>
    <script src="skript1.js"></script>
  </body>
```

```
1 {
2   margin: 0;
3   padding: 0;
4   box-sizing: border-box;
5 }
6
7 :root {
8   --primaryColor: rgb(255, 255, 255);
9   --secondaryColor: rgb(44, 44, 44);
10  --shadow: rgb(207, 207, 207);
11  --shadow2: rgb(243, 243, 243);
12  --shadowDark: rgb(104, 104, 104);
13  font-size: 20px;
14  font-family: "Antonio", sans-serif;
15 }
16
17 body {
18   width: 100vw;
19   overflow-x: hidden;
20   background-color: var(--primaryColor);
21 }
22
23 div.wrap {
24   width: 80%;
25   margin: 2vh auto;
26   text-align: center;
27 }
28
29 h1 {
30   font-size: 2rem;
31   color: var(--secondaryColor);
32 }
33
34 h1::after {
35   content: "";
36   display: block;
37   width: 12vw;
38   margin: 0.3rem auto;
39   border-top: 3px solid var(--secondaryColor);
40 }
```

W tym momencie pojawił się problem wynikający z ilości elementów, które należy dodać do strony i czasu jaki to zajmie.

W skrypcie JavaScript napisałem odpowiedni kod, który automatycznie dodaje do strony elementy zapisane w pliku .txt w formacie JSON.

```
function dodawanie(){
  dane.forEach(function(skrot){
    let addedEl = skroty.children[0].cloneNode(true);
    addedEl.children[0].textContent = skrot.h1;
    addedEl.children[1].style.backgroundImage = skrot.gif;
    addedEl.children[2].textContent = skrot.p;
    addedEl.classList.add(skrot.tags[0]);
    addedEl.classList.add(skrot.tags[1].replace(" ", ""));
    skroty.appendChild(addedEl);
  });
}
```

Teraz należało tylko zapisać sto elementów w postaci zapisanej poniżej.

```
{
  "h1": "2.Shift + F3",
  "p": "Otwiera okno Nodes.",
  "gif": "url('gify/2.vid2.gif')",
  "tags": [
    "wszystkie",
    "ogolne"
  ]
},
```

Gdzie *h1* jest nazwą skrótu, *p* jego opisem, *gif* ścieżką do odpowiedniego nagrania skonwertowanego na format .gif, a *tags* jest tablicą kategorii na podstawie, których potem odbywa się filtrowanie elementów.

Proces ręcznego zapisy do takiej postaci stu elementów byłby także dość czasochłonny.

Ten proces także postanowiłem zautomatyzować.

W tym przypadku pomocny okazał się język Python, w którym napisałem skrypt, który na podstawie szablonu obiektu JSON w formie tekstowej podczas iteracji w pętli wkleja odpowiednie zmienne w odpowiednie pola.

Kod programu

```
tekst = '''h1\": \"{0}\", \"p\": \"{1}\", \"gif\": \"url('{gify}/{2}.vid{2}.gif')\", \"tags\": [{\"wszystkie\", {3}}]'''
h1 = []
p = []

with open("dane.txt", "w") as file:
    with open("opisy.txt", "r") as opisy:
        numer = 1
        for line in opisy:
            line = line.replace("\n", "")
            if numer % 2 == 1:
                h1.append(line)
            else:
                p.append(line)
            numer += 1

    # print(h1)
    # print(len(h1))
    # print(p)
    # p.append("")
    # print(len(p))
    file.write("(")
    tag = "ogolne"
    for i in range(1, 100):
        file.write("{")
        if i + 1 > 97:
            tag = "render"
        elif i + 1 > 87:
            tag = "dope sheet"
        elif i + 1 > 77:
            tag = "image"
        elif i + 1 > 66:
            tag = "sculpting"
        elif i + 1 > 42:
            tag = "edit"
        file.write(tekst.format(h1[i], p[i], i + 1, tag))
        file.write("},")
    file.write(")")
```

Gdzie $h1$ jest tablicą z nazwami skrótów, a p jest tablicą opisów.

Są one tworzone na podstawie pliku, który stworzyłem w momencie opisywania skrótów na samym początku.

W zależności od numeru, który posiada skrót dodawany jest także *tag* z nazwą sekcji programu, której dotyczy.

Dane zapisane w formacie JSON

```
[
  {
    "h1": "2.Shift + F3",
    "p": "Otwiera okno Nodes.",
    "gif": "url('gify/2.vid2.gif')",
    "tags": [
      "wszystkie",
      "ogolne"
    ]
  },
  {
    "h1": "3.Shift + F4",
    "p": "Otwiera okno Python Console.",
    "gif": "url('gify/3.vid3.gif')",
    "tags": [
      "wszystkie",
      "ogolne"
    ]
  },
  {
    "h1": "4.Shift + F5",
    "p": "Otwiera okno 3D Viewport.",
    "gif": "url('gify/4.vid4.gif')",
    "tags": [
      "wszystkie",
      "ogolne"
    ]
  },
  {
    "h1": "5.Shift + F6",
    "p": "Otwiera okno Graph.",
    "gif": "url('gify/5.vid5.gif')",
    "tags": [
      "wszystkie",
      "ogolne"
    ]
  }
]
```

W ten sposób miałem już wszystkie rzeczy, których potrzebowałem do wykonania strony.

Elementy wygenerowały się poprawnie i teraz zostało jedynie wykonanie przycisków i filtrowania elementów.

Dodawanie przycisków na podstawie użytych w pliku JSON tagów

```
function categoriesLoad(){
  dane=JSON.parse(allText);
  const categories = dane.reduce(function (values, item) {
    item.tags.forEach(function (tag) {
      tag = tag.replace(" ", "");
      if (!values.includes(tag)) {
        values.push(tag);
      }
    });
    return values;
  }, ["ogolne"]);
  console.log(categories);
  const categoryButtons = categories.map(function (category) {
    return `<button class="${category}">${category}</button>`;
  }).join("");
  nav.innerHTML = categoryButtons;
  buttony = document.querySelectorAll("button");
}
```

Dodanie filtrowania przy wciśnięciu przycisku

```
tabSkroty = document.querySelectorAll("div.skrot");
buttony.forEach(function(butt){
  butt.addEventListener("click",function(){
    tabSkroty.forEach(function(skrot){
      if(skrot.classList.contains(butt.classList[0])){
        skrot.style.display = "block";
      }
      else{
        skrot.style.display = "none";
      }
    });
  });
});
```

Efekt końcowy

Skróty Klawiszowe

OGOLNE

WSZYSTKIE

EDIT

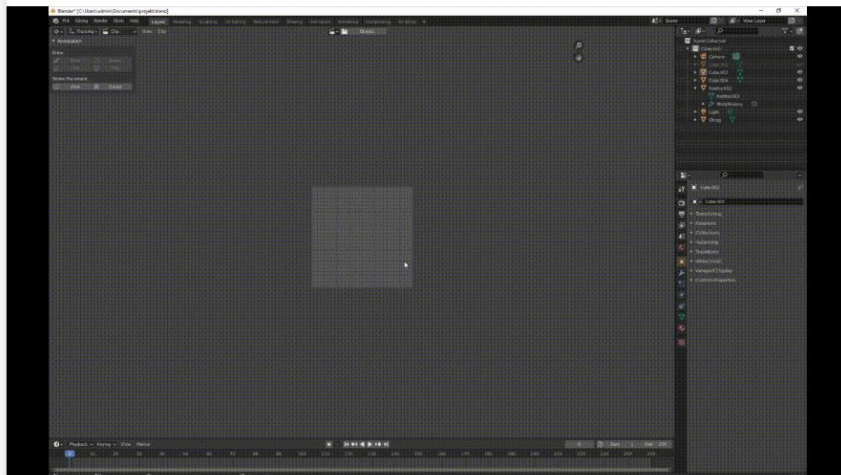
SCULPTING

IMAGE

DOPESHEET

RENDER

1. Shift + F2



Otwiera okno Movie Clip.

Filtrowanie także działa poprawnie:

Skróty Klawiszowe

OGOLNE

WSZYSTKIE

EDIT

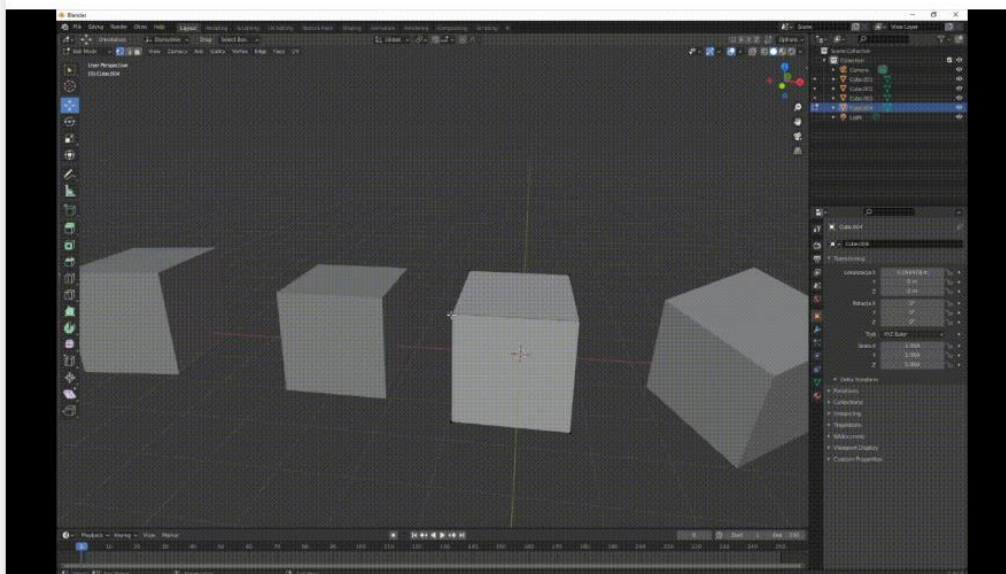
SCULPTING

IMAGE

DOPE SHEET

RENDER

43.1



Tryb manipulacji wierzchołkami.

Efekt końcowy projektu można zobaczyć pod adresem:

http://hackheroes.cba.pl/skroty_klaw/