



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS  
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS  
GRAFINIŲ SISTEMŲ KATEDRA

**Video technologijų  
Kursinis projektas  
“Loading spinner”**

Atliko: MKDf-17/2 gr. studentas Deividas Gineitis

Priėmė: lekt. Ernestas Čepulionis

Įvertinimas:.....

(pažymys, dėstytojo parašas)

Vilnius, 2021

## **TURINYS**

<b>TURINYS</b>	<b>1</b>
<b>PAVEIKSLĖLIŲ SĄRAŠAS</b>	<b>2</b>
<b>ĮVADAS</b>	<b>3</b>
<b>PROJEKTO APRAŠAS</b>	<b>4</b>
“Adobe After Effects” projektas	4
“ExtendScript” projektas	9
<b>IŠVADOS</b>	<b>19</b>
<b>ŠALTINIAI</b>	<b>20</b>

## PAVEIKSLĖLIŲ SĄRAŠAS

1 pav. Projekte naudojami sluoksniai	4
2 pav. “Spinner” sluoksnis	5
3 pav. “Spinner” sluoksnio išraiškos	5
4 pav. “Scooter” animacijos išraiška	5
5 pav. Antros linijos kadrai ir išraiška	6
6 pav. Pirmosios linijos išraiška	6
7 pav. “Main controller” valdiklis	6
8 pav. “Main controller” “clamp” išraiškos	7
9 pav. “Color controller” valdiklis	7
10 pav. Animacijos pradžia	8
11 pav. Projekto išvaizda	8
12 pav. Pradiniai projekto duomenys, trukmės “prompt”	9
13 pav. Sukuriamas projektas ir jame fonas	10
14 pav. Šuolio animacijos pradžios įvedimas	10
15 pav. “Main controller” sluoksnio kūrimas	11
16 pav. “Spinner” sluoksnio kūrimas	12
17 pav. “Outer spinner” sluoksnio kūrimas	13
18 pav. “Scooter” sluoksnio kūrimas, efekto pridėjimas, “Shape” kūrimas	14
19 pav. “Scooter” sluoksnio rėmo kūrimas su verteksais	14
20 pav. “Scooter” sluoksnio rėmo kūrimas su verteksais	15
21 pav. Išraiška parašyta keliose eilutėse naudojant “VS Code”	16
22 pav. “Speed lines” sluoksnio kūrimas	16
23 pav. Viršutinės linijos kūrimas	17
24 pav. Apatinės linijos kūrimas	17
25 pav. “Color controller” kūrimas	18

## IVADAS

“Adobe After Effects” programa plačiai naudojama kuriant kino filmų titrus, įžangas ir perėjimus, animacijas bei daugelį kitų grafinių elementų. Tai programa, sujungianti kinematografiją, grafinį dizainą, programavimą ir leidžianti kurti įvairius grafinius, vaizdo, 3D ar 2D elementus. Tad šiame projekte ši programa bus naudojama kuriant užsikrovimo suktuką (*angl.* “Loading spinner”).

### Tikslas:

Sukurti “Loading spinner” naudojant “Adobe After Effects” programą ir projektą atkurti pasitelkus “Javascript” programavimo kalbos plėtinį “ExtendScript”.

### Uždaviniai:

1. Sukurti projektą naudojant “Adobe After Effects” programą, kuriame:
  - 1.1. Panaudoti skirtingi sluoksniai
  - 1.2. Animacijos automatizuotos, joms panaudotos išraiškos (*angl.* “Expressions”).
  - 1.3. Tuščiam sluoksnyje (*angl.* “Null layer”) patalpinti naudojami išraiškos valdikliai.
2. Atkurti projektą .jsx faile naudojant “JavaScript” programavimo kalbą.

### Idėjos aprašymas

Projektas skirtas sukurti užsikrovimo suktuką, kuris bus naudojamas internetinėje svetainėje. Projektą sudaro keturios pagrindinės animacijos - rato nuolatinis sukimasis aplink apskritimą, logotipo atvykimas iki logotipo centro, logotipo šokinėjimas aukštyn ir žemyn bei linijų ilgejimas bei trumpėjimas atvaizduojant logotipo judėjimą. po logotipo pasirodymo visos kitos animacijos juda tol, kol pasibaigs projekte nustatytas laikas.

# PROJEKTO APRAŠAS

## 1. “Adobe After Effects” projektas

### Sluoksniai:

Projekte naudojami aštuoni sluoksniai. Sluoksnis “[Background]” yra projekto fonas. “Main controller” sluoksnis yra “null layer” ir jo pagrindinė paskirtis yra valdyti projekte naudojamoms išraiškoms. Sluoksnis “Spinner” yra pagrindinė suktuko animacija, o “Outer spinner” - apskritimo rėmas. “Scooter” sluoksnyje laikomas logotipas, “Scooter Mask” yra kaukė, kuri priverčia logotipui pranykti, jei jis nėra apskritimo viduje. “Speed lines” - tai linijos, kurios skirtos parodyti, jog paspirtukas juda. Galiausiai “Color controller” skirtas greitai pakeisti visų kitų (išskyrus “Main controller”) sluoksnių spalvoms.

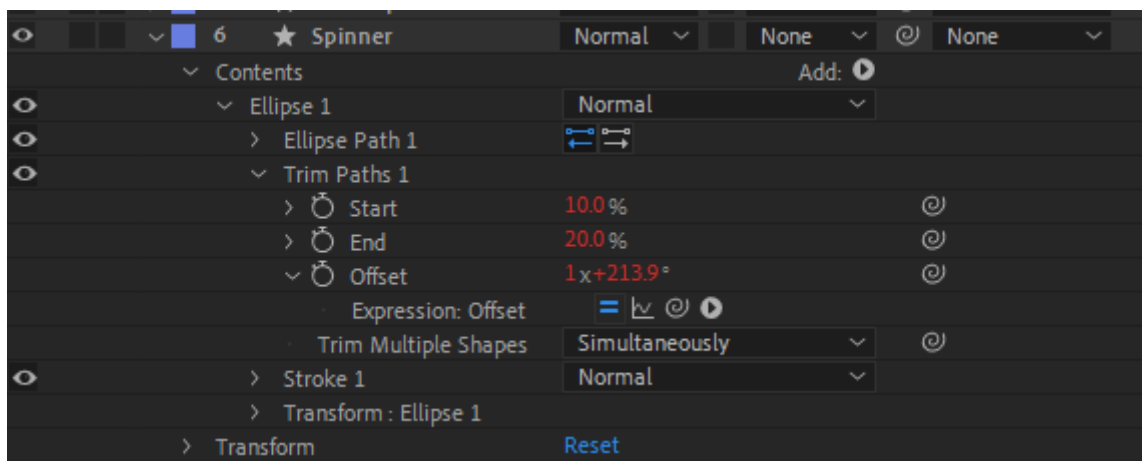


	Layer Name	Mode	T	TrkMat	Parent & Link
1	Color controller	Normal			None
2	Speed lines	Normal		None	4. Scooter
3	Scooter Mask	Normal		None	None
4	Scooter	Normal		Alpha	None
5	Outer spinner	Normal		None	None
6	Spinner	Normal		None	None
7	Main controller	Normal		None	None
8	[Background]	Normal		None	None

1 pav. Projekte naudojami sluoksniai

### Animacija:

Sluoksnis “Spinner” yra “Shape layer”, kuriame sukurtas apskritimas. Jam suteiktas “Trim paths” efektas, kuris nusako, kiek procentu nuo pradžios ar pabaigos nuimtas vektoriaus kelias. Naudojant išraiškas šiame efekte sukuriamą sukimosi iliuziją. “Start” ir “End” išraiškos aprašo animacijos pradžią, jog suktukas atsiranda, “Offset” suka vektorių aplink apskritimą viso projekto metu.



2 pav. “Spinner” sluoksnis

```
ease(time, 1, 2, 0, 10);
ease(time, 0, 2, 0, 20);
easeIn(time, 0, thisComp.duration, 0, 360*thisComp.duration);
```

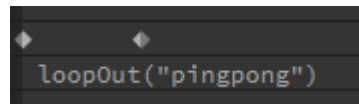
3 pav. “Spinner” sluoksnio išraiškos

“Scooter” sluoksnyje logotipo animacija prasideda su jo pasirodymu apskritimo centre, po to logotipas pradeda šokinėti. Šiai animacijai sukurti buvo naudojama išraiška keičiant “Position” koordinates. Pirmiausia išraiška tikrina ar jau praėjo vartotojo nustatytas laikas, praėjus prasideda šuolio animacija, jei ne, logotipas juda iki apskritimo centro. Tam naudojamas “delay” kintamasis, kurį nustato vartotojas. pavyzdžiui, jei nustato, jog šokinėjimas prasideda po 5s (delay = 5), tada logotipas keliauja iki centro pirmas 5 sekundes. Logotipui atvykus vyksta šuolio animacijos išraiška sukurta Natkin ir Maffitt (2009).

```
1 delay = thisComp.layer("Main controller").effect("Jump delay")("Slider");
2 if(time > delay){
3   surface = [960, 540];
4   jumpHeight = thisComp.layer("Main controller").effect("Jump height")("Slider");
5   jump = [960, jumpHeight];
6   period = thisComp.layer("Main controller").effect("Jump period")("Slider");
7   t = time % (period * 2);
8   if (t > period) t = 2 * period - t;
9   linear(Math.sin(t * Math.PI / period), 0, 1, surface, jump)
10 }else easeOut(time, 0, delay, [300,540],[960,540])
11
```

4 pav. “Scooter” animacijos išraiška

“Speed lines” sluoksnyje sukurtos dvi linijos. joms taip pat pritaikytas “Trim paths”, tačiau čia animuojamas tik “End” parametras. Antroje juostoje sukurti du kadrai, kurie per vieną sekundę keičia “End” procentus nuo 100% iki 20%. Tada išraiška tuos du kadrus kartoja viso projekto metu. Pirmosios linijos išraiška kopijuoja antrosios duomenis, tačiau su 0.1 sekundės uždelsimu.



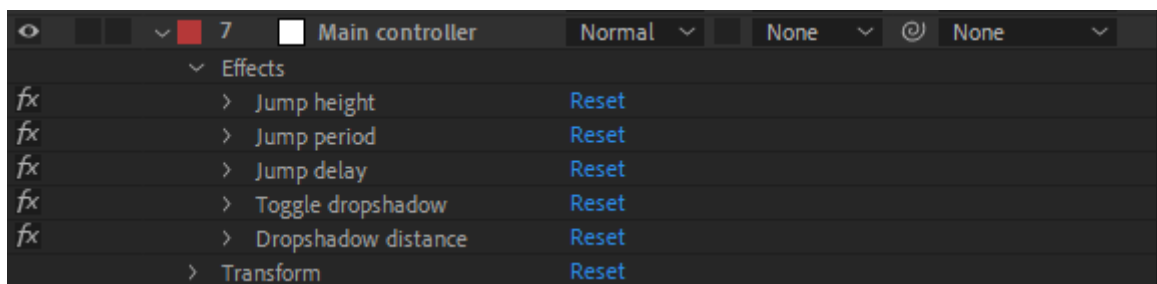
5 pav. Antros linijos kadrai ir išraiška

```
content("Shape 1").content("Trim Paths 1").end.valueAtTime(time - 0.1)
```

6 pav. Pirmosios linijos išraiška

### Valdikliai:

Projekte naudojami du valdikliai: “Main controller” ir “Color controller”. Šie valdikliai atskirti, kad būtų lengviau atskirti, vienas valdo animaciją, kitas - projekto spalvas. Pagrindinį valdiklį sudaro šolio aukščio, šolio periodo, šolio uždelsimo, šešėlio įjungimo ir atstumo valdikliai. Visi šie valdikliai yra “Slider” išskyrus “Toggle dropshadow”, kuris yra “Checkbox” tipo valdiklis. Visiems “Slider valdikliams sukurtos “clamp” išraiškos, kad vartotojas neįvestų per didelių ar per mažų reikšmių. Spalvų valdiklis tiesiog turi “Color” tipo valdiklius, kurie leidžia pasirinkti fono, logotipo, apskritimo, suktuko spalvas.



7 pav. “Main controller” valdiklis

```

clamp(value, -540, 540);

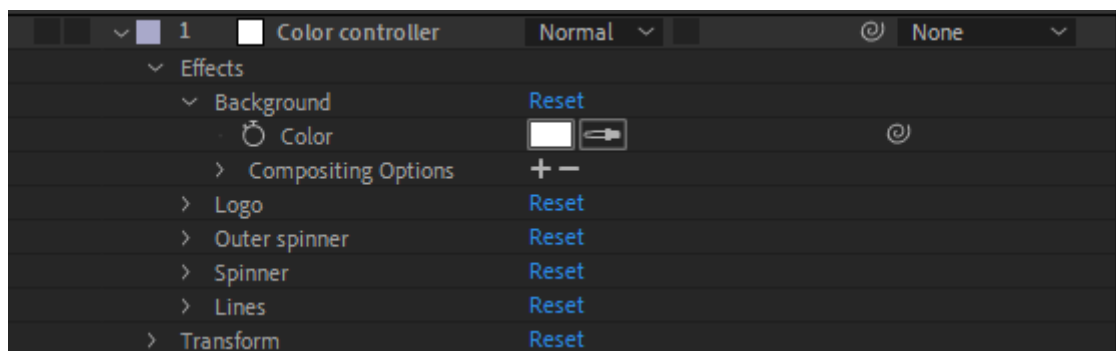
clamp(value, 0, thisComp.duration / 2);

clamp(value, 0, thisComp.duration);

clamp(value, 0, 100);

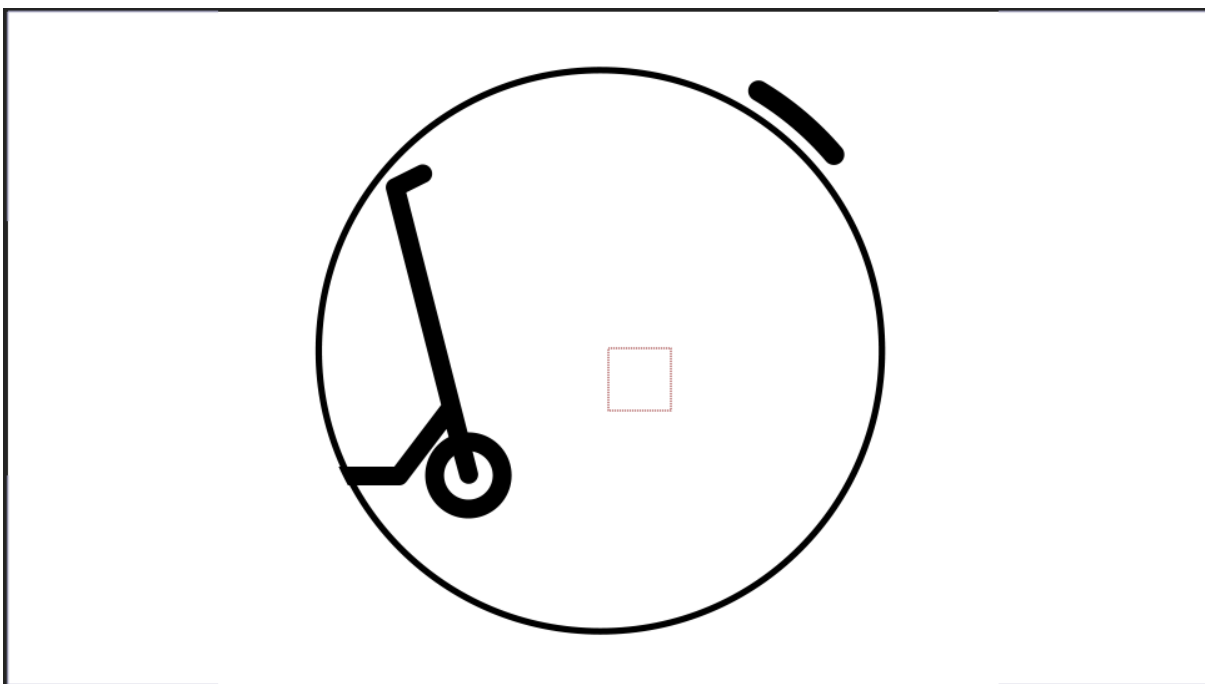
```

8 pav. “Main controller” “clamp” išraiškos.

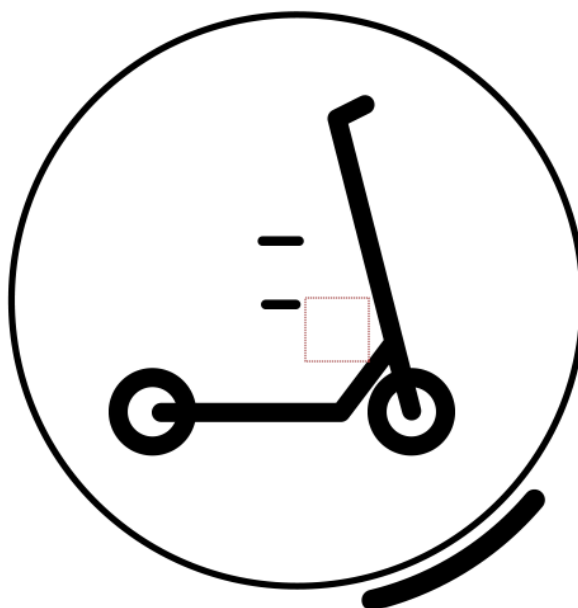


9 pav. “Color controller” valdiklis.





10 pav. Animacijos pradžia.



11 pav. Projekto išvaizda.

## 2. “ExtendScript” projektas

Sukūrus projektą naudojant “After Effects” pradėtas kurti tas pats projektas, tačiau naudojant “ExtendScript”. Programuojama buvo naudojant “VS Code” integruota kūrimo aplinka. Dažniausiai programuojant su “ExtendScript” naudojamas “ExtendScript Toolkit”, tačiau naudojant “ExtendScript debugger” plėtinį galima naudoti ir “VS Code”. Taip pat, visi komentarai rašomi anglų kalba, kad, esant reikalui, programa būtų prieinama visiems.

Pirmiausia sukuriamas pats projektas, nustatomi jo duomenys, prašoma vartotojui įvesti projekto trukmę, kuri yra tikrinama ir leidžiama įvesti tik sveikąjį skaičių nuo 1 iki 30, sukuriamas fonas.

```
2
3  app.project.close(CloseOptions.DO_NOT_SAVE_CHANGES);
4  app.newProject();
5
6  //pradiniai duomenys
7  var compWidth = 1920;
8  var compHeight = 1080;
9  var compDuration = null;
10 var aR = 1;
11 var compFPS = 25;
12
13 //Composition Duration
14 while (
15     compDuration == null ||
16     isNaN(compDuration) ||
17     compDuration % 1 !== 0 ||
18     compDuration < 1 ||
19     compDuration > 30
20 ) {
21     var compDuration = prompt("Enter composition duration", 10);
22     if (
23         compDuration == null ||
24         isNaN(compDuration) ||
25         compDuration % 1 !== 0 ||
26         compDuration < 1 ||
27         compDuration > 30
28     ) {
29         alert("Enter integer between 1 and 30");
30     }
31 }
32
```

12 pav. Pradiniai projekto duomenys, trukmės “prompt”.

```

54 //Generate project
55 var myComp = app.project.items.addComp(
56     "myComp",
57     compWidth,
58     compHeight,
59     aR,
60     compDuration,
61     compFPS
62 );
63
64 myComp.openInViewer();
65
66 //background
67 var background = myComp.layers.addSolid(
68     [255, 255, 255],
69     "background",
70     compWidth,
71     compHeight,
72     aR,
73     compDuration
74 );
75

```

13 pav. Sukuriamas projektas ir jame fonas.

Pradžioje taip pat yra “prompt”, kuris leidžia vartotojui įvesti šuolio animacijos pradžią. Įvedimas taip pat yra tikrinamas.

```

33 //Jump Delay Prompt
34 var jumpDelay = null;
35 while (
36     jumpDelay == null ||
37     isNaN(jumpDelay) ||
38     jumpDelay % 1 !== 0 ||
39     jumpDelay < 0 ||
40     jumpDelay > 5
41 ) {
42     var jumpDelay = prompt("Enter jump delay", 1);
43     if (var jumpDelay: string)
44         jumpDelay == null ||
45         isNaN(jumpDelay) ||
46         jumpDelay % 1 !== 0 ||
47         jumpDelay < 0 ||
48         jumpDelay > 5
49     ) {
50         alert("Enter number between 0 and 5");
51     }
52 }
53

```

14 pav. Šuolio animacijos pradžios įvedimas.

Visas kodas parašytas blokais. Kiekvienas blokas atsakingas už jam skirtą sluoksnio kūrimą. Visų sluoksnių kūrimas prasideda taip pat - pagrindinėje kompozicijoje sukuriamas reikiamas sluoksnio tipas, jam kuriami parametrai ir efektai kodo pagalba.

Pirma kuriamas “Main controller” sluoksnis. Sluoksniui kuriami valdikliai, jie pavadinimai, suteikiamos pradinės reikšmės bei priskiriamos išraiškos

```
76 {
77     app.beginUndoGroup("main controller");
78
79     var mainController = myComp.layers.addNull();
80     mainController.name = "Main controller";
81     //Jump height
82     var jumpHeightController = mainController.Effects.addProperty(
83         "ADBE Slider Control"
84     );
85     jumpHeightController.name = "Jump height";
86     jumpHeightController.property("Slider").setValue(500);
87     jumpHeightController.property("Slider").expression = "clamp(value,-540,540)";
88     app.endUndoGroup();
89
90     //Jump period
91     var jumpPeriodController = mainController.Effects.addProperty(
92         "ADBE Slider Control"
93     );
94     jumpPeriodController.name = "Jump period";
95     jumpPeriodController.property("Slider").setValue(0.5);
96     jumpPeriodController.property("Slider").expression = "clamp(value,0,5)";
97
98     //Jump delay
99     var jumpDelayController = mainController.Effects.addProperty(
100         "ADBE Slider Control"
101     );
102     jumpDelayController.name = "Jump delay";
103     jumpDelayController.property("Slider").setValue(jumpDelay);
104     jumpDelayController.property("Slider").expression = "clamp(value,0,thisComp.duration / 2)";
105     app.endUndoGroup();
106
107     //Toggle dropshadow
108     var dropShadowToggle = mainController.Effects.addProperty(
109         "ADBE Checkbox Control"
110     );
111     dropShadowToggle.name = "Toggle dropshadow";
112     dropShadowToggle.property("Checkbox").setValue(false);
113
114     //Dropshadow distance
115     var dropshadowDistance = mainController.Effects.addProperty(
116         "ADBE Slider Control"
117     );
118     dropshadowDistance.name = "Dropshadow distance";
119     dropshadowDistance.property("Slider").setValue(30);
120     dropshadowDistance.property("Slider").expression = "clamp(value,0,100)";
121     app.endUndoGroup();
122 }
```

15 pav. “Main controller” sluoksnio kūrimas.

Tada kuriamas “Spinner” sluoksnis. Vėl į kompoziciją pridedamas sluoksnis, tik šį kartą “Shape” tipo. Aprašoma pozicija, sukuriami vektorių grupė, kuri saugos apskritimo duomenis. Apskritimui aprašomas dydis, pridedami “Stroke”, kuri nusako suktuko spalvą, linijos storį bei linijos apvalumą, ir “Trim Paths”, aprašantis suktuko ilgį, parametrai. “Trim Paths” priskiriamos išraiškos.

```
//spinner
{
  app.beginUndoGroup("spinner");
  //create shapeLayer
  var spinner = myComp.layers.addShape();
  spinner.name = "spinner";
  spinner.property("Transform").property("Position").setValue([960, 544]);
  var spinnerContent = spinner
    .property("Contents")
    .addProperty("ADBE Vector Group");
  spinnerContent.name = "Ellipse 1";
  var spinnerGroup1 = spinner.property("Contents").property("Ellipse 1");
  //Add Ellipse
  var spinnerEllipse1 = spinnerGroup1
    .property("Contents")
    .addProperty("ADBE Vector Shape - Ellipse");
  spinnerEllipse1.property("Size").setValue([900, 900]); //Set Size
  //Stroke
  var spinnerEllipseStroke = spinnerGroup1
    .property("Contents")
    .addProperty("ADBE Vector Graphic - Stroke");
  spinnerEllipseStroke.property("Color").setValue([0, 0, 0]); //STROKE COLOR
  spinnerEllipseStroke.property("Stroke Width").setValue(30);
  spinnerEllipseStroke.property("Line Cap").setValue(2);
  //Trim Paths
  var spinnerEllipseTrimPaths = spinnerGroup1
    .property("Contents")
    .addProperty("ADBE Vector Filter - Trim");
  spinnerEllipseTrimPaths.property("Start").expression =
    "ease(time, 1, 2, 0, 10)";
  spinnerEllipseTrimPaths.property("End").expression =
    "ease(time, 0, 2, 0, 20)";
  spinnerEllipseTrimPaths.property("Offset").expression =
    "easeIn(time, 0, thisComp.duration, 0, 360 * thisComp.duration)";
  app.endUndoGroup();
}
```

16 pav. “Spinner” sluoksnio kūrimas.

Lygiai taip pat kuriamas ir “Outer spinner” sluoksnis, tik jam nėra priskiriamų išraiškų ar “Trim Paths”.

```
161 //outer spinner
162 {
163     app.beginUndoGroup("outer spinner");
164     //create shapeLayer
165     var outerSpinner = myComp.layers.addShape();
166     outerSpinner.name = "outerSpinner";
167     outerSpinner.property("Transform").property("Position").setValue([960, 544]);
168     outerSpinner.property("Transform").property("Scale").setValue([92.6, 92.6]);
169     var outerSpinnerContent = outerSpinner
170         .property("Contents")
171         .addProperty("ADBE Vector Group");
172     outerSpinnerContent.name = "Ellipse 1";
173     var outerSpinnerGroup1 = outerSpinner
174         .property("Contents")
175         .property("Ellipse 1");
176     //Add Ellipse
177     var outerSpinnerEllipse1 = outerSpinnerGroup1
178         .property("Contents")
179         .addProperty("ADBE Vector Shape - Ellipse");
180     outerSpinnerEllipse1.property("Size").setValue([900, 900]); //Set Size
181     //Stroke
182     var outerSpinnerEllipseStroke = outerSpinnerGroup1
183         .property("Contents")
184         .addProperty("ADBE Vector Graphic - Stroke");
185     outerSpinnerEllipseStroke.property("Color").setValue([0, 0, 0]); //STROKE COLOR
186     outerSpinnerEllipseStroke.property("Stroke Width").setValue(10);
187     app.endUndoGroup();
188 }
189
```

17 pav. “Outer spinner” sluoksnio kūrimas.

Kuriant “Scooter” sluoksnį darbo eiga taipogi prasidėjo taip pat, tačiau iškilo keletas sunkumų bei nesklandumų. Sukuriamas sluoksnis, jame pridedamas “Drop Shadow” efektas bei sukuriami “Shape” sluoksniai, kurie sukurs logotipo vaizdą. taip pat sluoksniui priskiriamas “Track Matte Alpha”, kad po jo einantis kaukės sluoksnis jį pradangintų.

```

190 //scooter
191 {
192     app.beginUndoGroup("scooter");
193     var scooter = myComp.layers.addShape();
194     scooter.name = "scooter";
195     scooter.trackMatteType = TrackMatteType.ALPHA;
196
197     //Drop Shadow
198     var scooterDropShadow = scooter.Effects.addProperty("ADBE Drop Shadow");
199     scooterDropShadow.property("Opacity").expression =
200     'if(thisComp.layer("Main controller").effect("Toggle dropshadow")("Checkbox") == 1) {50} else {0}';
201     scooterDropShadow.property("Distance").expression =
202     'thisComp.layer("Main controller").effect("Dropshadow distance")("Slider")';
203
204     //create shape groups
205     var frameContent = scooter
206     .property("Contents")
207     .addProperty("ADBE Vector Group");
208     var leftWheelContent = scooter
209     .property("Contents")
210     .addProperty("ADBE Vector Group");
211     var rightWheelContent = scooter
212     .property("Contents")
213     .addProperty("ADBE Vector Group");
214

```

18 pav. “Scooter” sluoksnio kūrimas, efekto pridėjimas, “Shape” kūrimas.

Sukurtame “Frame” sluoksnyje verteksu pagalba sukurtas paspirtuko rėmas. Aprašomos verteksu koordinatės, tie vertekasai sujungiami, pasirenkama, kad jie būtų atviri, tam, kad nesudarytų figūros, o pasilikytų kaip linijos.

```

215 //Frame
216 var frame = scooter.property("Contents").property("Group 1");
217 frame.name = "frame";
218 var framePath = frame
219     .property("Contents")
220     .addProperty("ADBE Vector Shape - Group");
221 var frameMask = framePath.property("Path");
222 var frameM = frameMask.value;
223 //frame vertices
224 var v1 = [-230, 204];
225 var v2 = [25, 204];
226 var v3 = [130, 104];
227 var v4 = [167, 204];
228 var v5 = [30, -204];
229 var v6 = [70, -214];
230 //combine vertices
231 frameM.vertices = [v1, v2, v3, v4, v5, v6];
232 frameM.closed = false;
233 frameMask.setValue(frameM);
234 //frame stroke
235 var frameStroke = frame
236     .property("Contents")
237     .addProperty("ADBE Vector Graphic - Stroke");
238 frameStroke.property("Stroke Width").setValue(30);
239 frameStroke.property("Line Cap").setValue(2);
240 frameStroke.property("Line Join").setValue(2);
241 frameStroke.property("Color").setValue([0, 0, 0]);
242

```

19 pav. “Scooter” sluoksnio rėmo kūrimas su vertekmais.

Kuriant ratus atskirai sukurti du sluoksniai kairiajam ir dešiniajam ratams.

```
243 //Left wheel
244 var leftWheel = scooter.property("Contents").property("Group 2");
245 leftWheel.name = "left wheel";
246 leftWheel.property("Transform").property("Position").setValue([-241, 204]);
247 //left wheel ellipse
248 var leftWheelEllipse = leftWheel
249     .property("Contents")
250     .addProperty("ADBE Vector Shape - Ellipse");
251 leftWheelEllipse.property("Size").setValue([108, 108]);
252 //left wheel stroke
253 var leftWheelStroke = leftWheel
254     .property("Contents")
255     .addProperty("ADBE Vector Graphic - Stroke");
256 leftWheelStroke.property("Stroke Width").setValue(30);
257 leftWheelStroke.property("Line Cap").setValue(2);
258 leftWheelStroke.property("Color").setValue([0, 0, 0]);
259
260 //Right wheel
261 var rightWheel = scooter.property("Contents").property("Group 3");
262 rightWheel.name = "right wheel";
263 rightWheel.property("Transform").property("Position").setValue([167, 204]);
264 //right wheel ellipse
265 var rightWheelEllipse = rightWheel
266     .property("Contents")
267     .addProperty("ADBE Vector Shape - Ellipse");
268 rightWheelEllipse.property("Size").setValue([108, 108]);
269 //right wheel stroke
270 var rightWheelStroke = rightWheel
271     .property("Contents")
272     .addProperty("ADBE Vector Graphic - Stroke");
273 rightWheelStroke.property("Stroke Width").setValue(30);
274 rightWheelStroke.property("Line Cap").setValue(2);
275 rightWheelStroke.property("Color").setValue([0, 0, 0]);
276
```

20 pav. “Scooter” sluoksnio rėmo kūrimas su verteksais.

Vienas iš kėblumų naudojant “VS Code”, o ne “ExtendScript Toolkit” yra tas, jog “VS Code” nepalaiko “Extenscript” naudojamo simbolio, kuris leistų rašyti “string” keliose eilutėse. “VS Code” naudojamas `` simbolis, “ExtendScript” naudoja trigubas kabutes ““ ““. Dėl šios priežasties ilgesnės išraiškos parašytos vienoje eilutėje.



```

277 //scooter animation
278 scooter.property("Transform").property("Position").expression =
279     'delay = thisComp.layer("Main controller").effect("Jump delay")("Slider");
280     if(time > delay){
281         surface = [960, 540];
282         jumpHeight = thisComp.layer("Main controller").effect("Jump height")("Slider");
283         jump = [960, jumpHeight];
284         period = thisComp.layer("Main controller").effect("Jump period")("Slider");
285         t = time % (period * 2); if (t > period) t = 2 * period - t;
286         linear(Math.sin(t * Math.PI / period), 0, 1, surface, jump)
287     }else easeOut(time, 0, delay, [300,540],[960,540]);
288

```

21 pav. Išraiška parašyta keliose eilutėse naudojant “VS Code”.

Greičio linijų kūrimas lygiai toks pat, kaip kuriant paspirtuką. Sukuriamos “Shape” grupės, jos sukuriamos verteksų pagalba, aprašomi jų parametrai, priskiriamos išraiškos bei pridedamas “Drop Shadow” efektas. Tačiau vėl, rašant išraišką teko rašyti ją vienoje eilutėje.

```

313
314 app.beginUndoGroup("speed lines");
315 //create Layer
316 var speedLines = myComp.layers.addShape();
317 speedLines.name = "speedLines";
318 speedLines.parent = scooter;
319 speedLines.property("Transform").property("Opacity").setValue(0);
320 speedLines.property("Transform").property("Opacity").expression =
321     'jumpStart = thisComp.layer("Main controller").effect("Jump delay")("Slider");
322     visible = jumpStart + framesToTime(5);ease(time, jumpStart, visible, 0, 100);';
323
324 //Dropshadow
325 var speedLinesDropshadow = speedLines.Effects.addProperty("ADBE Drop Shadow");
326 speedLinesDropshadow.property("Opacity").expression =
327     'if(thisComp.layer("Main controller").effect("Toggle dropshadow")("Checkbox") == 1) {50} else {0}';
328 speedLinesDropshadow.property("Distance").expression =
329     'thisComp.layer("Main controller").effect("Dropshadow distance")("Slider)';
330 //create shape groups
331 var topLineContent = speedLines
332     .property("Contents")
333     .addProperty("ADBE Vector Group");
334 var BottomLineContent = speedLines
335     .property("Contents")
336     .addProperty("ADBE Vector Group");
337

```

22 pav. “Speed lines” sluoksnio kūrimas.

```

337 //Top Line
338 var topLine = speedLines.property("Contents").property("Group 1");
339 topLine.name = "top line";
340 var topLinePath = topLine
341     .property("Contents")
342     .addProperty("ADBE Vector Shape - Group");
343 var topLineMask = topLinePath.property("Path");
344 var topLineM = topLineMask.value;
345 var vt1 = [-810, -50];
346 var vt2 = [-710, -50];
347 topLineM.vertices = [vt1, vt2];
348 topLineM.closed = false;
349 topLineMask.setValue(topLineM);
350 //top stroke
351 var topLineStroke = topLine
352     .property("Contents")
353     .addProperty("ADBE Vector Graphic - Stroke");
354 topLineStroke.property("Stroke Width").setValue(30);
355 topLineStroke.property("Line Cap").setValue(2);
356 topLineStroke.property("Color").setValue([0, 0, 0]);
357 //top trimp path
358 var topLineTrimPath = topLine
359     .property("Contents")
360     .addProperty("ADBE Vector Filter - Trim");
361 topLineTrimPath.property("Start").expression =
362     'content("bottom line").content("Trim Paths 1").start.valueAtTime(time -0.1)';
363

```

23 pav. Viršutinės linijos kūrimas.

```

364 //Bottom Line
365 var bottomLine = speedLines.property("Contents").property("Group 2");
366 bottomLine.name = "bottom line";
367 var bottomLinePath = bottomLine
368     .property("Contents")
369     .addProperty("ADBE Vector Shape - Group");
370 var bottomLineMask = bottomLinePath.property("Path");
371 var bottomLineM = bottomLineMask.value;
372 var vb1 = [-910, 50];
373 var vb2 = [-710, 50];
374 bottomLineM.vertices = [vb1, vb2];
375 bottomLineM.closed = false;
376 bottomLineMask.setValue(bottomLineM);
377 //bottom stroke
378 var bottomLineStroke = bottomLine
379     .property("Contents")
380     .addProperty("ADBE Vector Graphic - Stroke");
381 bottomLineStroke.property("Stroke Width").setValue(30);
382 bottomLineStroke.property("Line Cap").setValue(2);
383 bottomLineStroke.property("Color").setValue([0, 0, 0]);
384 //bottom trim path
385 var bottomLineTrimPath = bottomLine
386     .property("Contents")
387     .addProperty("ADBE Vector Filter - Trim");
388 bottomLineTrimPath.property("Start").setValueAtTime(0, 100);
389 bottomLineTrimPath.property("Start").setValueAtTime(1, 20);
390 bottomLineTrimPath.property("Start").expression = 'loopOut("pingpong")';
391
392 app.endUndoGroup();
393 }

```

24 pav. Apatinės linijos kūrimas.

Galiausiai kuriamas “Color controller” sluoksnis, kuris kuriamas lygiai taip pat, kaip ir “Main controller”, tačiau čia dar kiekvienam sukurtam sluoksniui priskiriamas “Fill” efektas ir jam kuriama išraiška, kur leidžia valdikliui valdyti spalva. Spalvos taip pat gauna įprastas reikšmes.

```
402 //Background
403 var backgroundColor = colorController.Effects.addProperty(
404     "ADBE Color Control"
405 );
406 backgroundColor.name = "Background";
407 backgroundColor.property("Color").setValue([255, 255, 255]);
408 //Adding color controller to background
409 var bgFill = background.Effects.addProperty("ADBE Fill");
410 bgFill.property("Color").expression =
411     'thisComp.layer("Color controller").effect("Background")("Color")';
412
413 //Logo
414 var logoColor = colorController.Effects.addProperty("ADBE Color Control");
415 logoColor.name = "Logo";
416 logoColor.property("Color").setValue([0, 0, 0]);
417 //Adding color controller to background
418 var logoFill = scooter.Effects.addProperty("ADBE Fill");
419 logoFill.property("Color").expression =
420     'thisComp.layer("Color controller").effect("Logo")("Color")';
421
422 //Outer spinner
423 var outerSpinnerColor = colorController.Effects.addProperty(
424     "ADBE Color Control"
425 );
426 outerSpinnerColor.name = "Outer spinner";
427 outerSpinnerColor.property("Color").setValue([0, 0, 0]);
428 //Adding color controller to background
429 var outerFill = outerSpinner.Effects.addProperty("ADBE Fill");
430 outerFill.property("Color").expression =
431     'thisComp.layer("Color controller").effect("Outer spinner")("Color")';
432
433 //Spinner
434 var spinnerColor = colorController.Effects.addProperty("ADBE Color Control");
435 spinnerColor.name = "Spinner";
436 spinnerColor.property("Color").setValue([0, 0, 0]);
437 //Adding color controller to background
438 var spinnerFill = spinner.Effects.addProperty("ADBE Fill");
439 spinnerFill.property("Color").expression =
440     'thisComp.layer("Color controller").effect("Spinner")("Color")';
441
442 //Lines
443 var linesColor = colorController.Effects.addProperty("ADBE Color Control");
444 linesColor.name = "Lines";
445 linesColor.property("Color").setValue([0, 0, 0]);
```

25 pav. “Color controller” kūrimas.

## **IŠVADOS**

Taigi, šio projekto iškelti uždaviniai įvykdyti. Sukurtas numatytas projektas naudojant “After Effect” programa, jam sukurti reikiami sluoksniai, animacijos, kurios išraiškų pagalba buvo automatizuotos ir kurias valdiklių pagalba galima valdyti. Taip pat sukurtas projektas atkartotas naudojant “ExtendScript”.

## ŠALTINIAI

Michael Natkin & Brian Maffitt, (2009). Bounce/Jump expression.

<http://www.motion-graphics-exchange.com/after-effects/Bounce-Jump/496a0da1290d2>

CS6 Scripting Guide: <https://ae-scripting.docsforadobe.dev/>