



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 7
по курсу «Разработка мобильных приложений»
«Визуализация 3D объекта руки с возможностью захвата
объекта»

Студент группы ИУ9-72Б Шемякин В.А.

Преподаватель Посевин Д. П.

Moskva 2025

1 Задание

1. Реализовать 3D-модель руки согласно видео.
2. Добавить в сцену любой OBJ объект — это может быть сфера, куб, череп или иная фигура.
3. Реализовать хватание кистью руки объекта из п.2 и перемещение по сцене.

2 Результаты

Исходный код программы представлен в листинге 1.

```
1 import 'package:ditredi/ditredi.dart';
2 import 'package:flutter/material.dart';
3 import 'package:vector_math/vector_math_64.dart' as vector;
4 import 'dart:math';
5
6 void main() {
7   runApp(const MyApp());
8 }
9
10 class MyApp extends StatefulWidget {
11   const MyApp({Key? key}) : super(key: key);
12
13   @override
14   State<MyApp> createState() => _MyAppState();
15 }
16
17 class _MyAppState extends State<MyApp> {
18   var indexAngle = 0.0;
19   var middleAngle = 0.0;
20   var ringAngle = 0.0;
21   var pinkyAngle = 0.0;
22
23   var handX = 0.0;
24   var handY = 0.0;
25   var handZ = 0.0;
26
27   var earthX = 0.0;
28   var earthY = 40.0;
29   var earthZ = 0.0;
30 }
```

```

31    final double grabRadius = 15.0;
32    final double collisionRadius = 5.0;
33    bool isGrabbed = false;
34    double grabbedOffsetX = 0.0;
35    double grabbedOffsetY = 0.0;
36    double grabbedOffsetZ = 0.0;
37
38    final Future<List<Mesh3D>> sphere = _generatePoints();
39
40    bool canGrab() {
41        final distance = sqrt(
42            pow(handX - earthX, 2) +
43            pow(handY - earthY, 2) +
44            pow(handZ - earthZ, 2)
45        );
46        return distance <= grabRadius;
47    }
48
49    bool checkCollision(double newHandX, double newHandY, double
50        newHandZ) {
51        final distance = sqrt(
52            pow(newHandX - earthX, 2) +
53            pow(newHandY - earthY, 2) +
54            pow(newHandZ - earthZ, 2)
55        );
56        if (!isGrabbed) {
57            return distance < collisionRadius;
58        }
59        return false;
60    }
61
62    final _controller = DiTreDiController(
63        rotationX: 0,
64        rotationY: 0,
65        light: vector.Vector3(-0.5, -0.5, 0.5),
66        maxUserScale: 10,
67        minUserScale: 0.5,
68        userScale: 1,
69    );
70
71    @override
72    Widget build(BuildContext context) {
73        return MaterialApp(
74            darkTheme: ThemeData.dark(),
75            title: 'DiTreDi Demo',

```

```

75     theme: ThemeData(
76         primarySwatch: Colors.blue,
77     ),
78     home: Scaffold(
79         body: SafeArea(
80             child: Flex(
81                 crossAxisAlignment: CrossAxisAlignment.start,
82                 direction: Axis.vertical,
83                 children: [
84                     FutureBuilder(
85                         future: sphere,
86                         builder: (BuildContext context, AsyncSnapshot<
87                         List<Mesh3D>> snapshot) {
88                         List<Widget> children;
89                         if (snapshot.hasData) {
90                             children = <Widget>[
91                                 Expanded(
92                                     child: DiTreDiDraggable(
93                                         controller: _controller,
94                                         child: DiTreDi(
95                                             figures: [
96                                                 TransformModifier3D(
97                                                     snapshot.data![0],
98                                                     Matrix4.identity(),
99                                                     ..translate(handX, handY,
100                                                       handZ)
101                                                     ..rotateX(-pi/2)
102                                         ),
103                                         TransformModifier3D(
104                                             snapshot.data![1],
105                                             Matrix4.identity(),
106                                             ..translate(handX, handY,
107                                               handZ)
108                                             ..rotateX(-pi/2)
109                                             ..translate(3.05, 1.15, 8.75)
110                                             ..translate(-0.2, -0.25, -2.2)
111                                             ..rotateX(-(indexAngle * pi
112                                               /18))
113                                             ..translate(0.2, 0.25, 2.2)
114                                         ),
115                                         TransformModifier3D(
116                                             snapshot.data![2],
117                                             Matrix4.identity(),
118                                             ..translate(handX, handY,
119                                               handZ)
120                                         )
121                                     ],
122                                 );
123                             }
124                         );
125                     );
126                 ],
127             );
128         );
129     );
130 
```

```

115          .. rotateX(-pi/2)
116          .. translate(0.7, 0.0, 9.75)
117          .. translate(0.0, -0.5, -2.25)
118          .. rotateX(-(middleAngle * pi
119          /18)))
120          .. translate(0.0, 0.5, 2.25)
121      ),
122      TransformModifier3D(
123          snapshot.data![3],
124          Matrix4.identity()
125          .. translate(handX, handY,
126          handZ)
127          .. rotateX(-pi/2)
128          .. translate(-2.0, -0.56, 9.1)
129          .. translate(0.0, -0.25, -2.2)
130          .. rotateX(-(ringAngle * pi
131          /18)))
132          .. translate(0.0, 0.25, 2.2)
133          .. rotate
134      ),
135      TransformModifier3D(
136          snapshot.data![4],
137          Matrix4.identity()
138          .. translate(handX, handY,
139          handZ)
140          .. rotateX(-pi/2)
141          .. translate(-4.65, -1.0, 7.15)
142          .. translate(0.0, 0.0, -1.25)
143          .. rotateX(-(pinkyAngle * pi
144          /18)))
145          .. translate(0.0, 0.0, 1.25)
146      ),
147      TransformModifier3D(
148          snapshot.data![5],
149          Matrix4.identity()
150          .. translate(earthX, earthY,
151          earthZ)
152          .. rotateX(-pi/2)
153          .. rotateY(pi)

```

```
154 const Padding(                                       
155   padding: EdgeInsets.all(8.0),  
156   child: Text("Drag to rotate. Scroll to  
157   zoom") ,  
158 ) ,  
159 Expanded(  
160   child: Column(  
161     mainAxisAlignment: MainAxisAlignment.min,  
162     children: [  
163       //  
164  
165       const SizedBox(height: 10) ,  
166       const Text("                                     ",  
167         style: TextStyle(fontWeight: FontWeight.bold)) ,  
168       const SizedBox(height: 10) ,  
169     Row(  
170       mainAxisAlignment: MainAxisAlignment.center ,  
171       children: [  
172         Column(  
173           children: [  
174             ElevatedButton(  
175               onPressed: () {  
176                 setState(() {  
177                   final newY = handY + 4.0;  
178                   if (!checkCollision(handX, newY, handZ)) {  
179                     handY = newY;  
180                     if (isGrabbed) earthY = handY +  
181                         grabbedOffsetY;  
182                   }  
183                 }));  
184             },  
185             child: const Icon(Icons.arrow_upward) ,  
186           ),  
187         Row(  
188           mainAxisAlignment: MainAxisAlignment.center ,  
189           children: [  
190             ElevatedButton(  
191               onPressed: () {  
192                 setState(() {  
193                   final newX = handX - 4.0;  
194                   if (!checkCollision(newX, handY, handZ)) {  
195                     handX = newX;  
196                     if (isGrabbed) earthX = handX +  
197                         grabbedOffsetX;  
198                   }  
199                 });  
200             },  
201           ),  
202         ),  
203       ),  
204     ),  
205   ),  
206 );  
207 );  
208 );  
209 );  
210 );  
211 );  
212 );  
213 );  
214 );  
215 );  
216 );  
217 );  
218 );  
219 );  
220 );  
221 );  
222 );  
223 );  
224 );  
225 );  
226 );  
227 );  
228 );  
229 );  
230 );  
231 );  
232 );  
233 );  
234 );  
235 );  
236 );  
237 );  
238 );  
239 );  
240 );  
241 );  
242 );  
243 );  
244 );  
245 );  
246 );  
247 );  
248 );  
249 );  
250 );  
251 );  
252 );  
253 );  
254 );  
255 );  
256 );  
257 );  
258 );  
259 );  
260 );  
261 );  
262 );  
263 );  
264 );  
265 );  
266 );  
267 );  
268 );  
269 );  
270 );  
271 );  
272 );  
273 );  
274 );  
275 );  
276 );  
277 );  
278 );  
279 );  
280 );  
281 );  
282 );  
283 );  
284 );  
285 );  
286 );  
287 );  
288 );  
289 );  
290 );  
291 );  
292 );  
293 );  
294 );  
295 );  
296 );  
297 );  
298 );  
299 );  
300 );  
301 );  
302 );  
303 );  
304 );  
305 );  
306 );  
307 );  
308 );  
309 );  
310 );  
311 );  
312 );  
313 );  
314 );  
315 );  
316 );  
317 );  
318 );  
319 );  
320 );  
321 );  
322 );  
323 );  
324 );  
325 );  
326 );  
327 );  
328 );  
329 );  
330 );  
331 );  
332 );  
333 );  
334 );  
335 );  
336 );  
337 );  
338 );  
339 );  
340 );  
341 );  
342 );  
343 );  
344 );  
345 );  
346 );  
347 );  
348 );  
349 );  
350 );  
351 );  
352 );  
353 );  
354 );  
355 );  
356 );  
357 );  
358 );  
359 );  
360 );  
361 );  
362 );  
363 );  
364 );  
365 );  
366 );  
367 );  
368 );  
369 );  
370 );  
371 );  
372 );  
373 );  
374 );  
375 );  
376 );  
377 );  
378 );  
379 );  
380 );  
381 );  
382 );  
383 );  
384 );  
385 );  
386 );  
387 );  
388 );  
389 );  
390 );  
391 );  
392 );  
393 );  
394 );  
395 );  
396 );  
397 );  
398 );  
399 );  
400 );  
401 );  
402 );  
403 );  
404 );  
405 );  
406 );  
407 );  
408 );  
409 );  
410 );  
411 );  
412 );  
413 );  
414 );  
415 );  
416 );  
417 );  
418 );  
419 );  
420 );  
421 );  
422 );  
423 );  
424 );  
425 );  
426 );  
427 );  
428 );  
429 );  
430 );  
431 );  
432 );  
433 );  
434 );  
435 );  
436 );  
437 );  
438 );  
439 );  
440 );  
441 );  
442 );  
443 );  
444 );  
445 );  
446 );  
447 );  
448 );  
449 );  
450 );  
451 );  
452 );  
453 );  
454 );  
455 );  
456 );  
457 );  
458 );  
459 );  
460 );  
461 );  
462 );  
463 );  
464 );  
465 );  
466 );  
467 );  
468 );  
469 );  
470 );  
471 );  
472 );  
473 );  
474 );  
475 );  
476 );  
477 );  
478 );  
479 );  
480 );  
481 );  
482 );  
483 );  
484 );  
485 );  
486 );  
487 );  
488 );  
489 );  
490 );  
491 );  
492 );  
493 );  
494 );  
495 );  
496 );  
497 );  
498 );  
499 );  
500 );  
501 );  
502 );  
503 );  
504 );  
505 );  
506 );  
507 );  
508 );  
509 );  
510 );  
511 );  
512 );  
513 );  
514 );  
515 );  
516 );  
517 );  
518 );  
519 );  
520 );  
521 );  
522 );  
523 );  
524 );  
525 );  
526 );  
527 );  
528 );  
529 );  
530 );  
531 );  
532 );  
533 );  
534 );  
535 );  
536 );  
537 );  
538 );  
539 );  
540 );  
541 );  
542 );  
543 );  
544 );  
545 );  
546 );  
547 );  
548 );  
549 );  
550 );  
551 );  
552 );  
553 );  
554 );  
555 );  
556 );  
557 );  
558 );  
559 );  
560 );  
561 );  
562 );  
563 );  
564 );  
565 );  
566 );  
567 );  
568 );  
569 );  
570 );  
571 );  
572 );  
573 );  
574 );  
575 );  
576 );  
577 );  
578 );  
579 );  
580 );  
581 );  
582 );  
583 );  
584 );  
585 );  
586 );  
587 );  
588 );  
589 );  
590 );  
591 );  
592 );  
593 );  
594 );  
595 );  
596 );  
597 );  
598 );  
599 );  
600 );  
601 );  
602 );  
603 );  
604 );  
605 );  
606 );  
607 );  
608 );  
609 );  
610 );  
611 );  
612 );  
613 );  
614 );  
615 );  
616 );  
617 );  
618 );  
619 );  
620 );  
621 );  
622 );  
623 );  
624 );  
625 );  
626 );  
627 );  
628 );  
629 );  
630 );  
631 );  
632 );  
633 );  
634 );  
635 );  
636 );  
637 );  
638 );  
639 );  
640 );  
641 );  
642 );  
643 );  
644 );  
645 );  
646 );  
647 );  
648 );  
649 );  
650 );  
651 );  
652 );  
653 );  
654 );  
655 );  
656 );  
657 );  
658 );  
659 );  
660 );  
661 );  
662 );  
663 );  
664 );  
665 );  
666 );  
667 );  
668 );  
669 );  
670 );  
671 );  
672 );  
673 );  
674 );  
675 );  
676 );  
677 );  
678 );  
679 );  
680 );  
681 );  
682 );  
683 );  
684 );  
685 );  
686 );  
687 );  
688 );  
689 );  
690 );  
691 );  
692 );  
693 );  
694 );  
695 );  
696 );  
697 );  
698 );  
699 );  
700 );  
701 );  
702 );  
703 );  
704 );  
705 );  
706 );  
707 );  
708 );  
709 );  
710 );  
711 );  
712 );  
713 );  
714 );  
715 );  
716 );  
717 );  
718 );  
719 );  
720 );  
721 );  
722 );  
723 );  
724 );  
725 );  
726 );  
727 );  
728 );  
729 );  
730 );  
731 );  
732 );  
733 );  
734 );  
735 );  
736 );  
737 );  
738 );  
739 );  
740 );  
741 );  
742 );  
743 );  
744 );  
745 );  
746 );  
747 );  
748 );  
749 );  
750 );  
751 );  
752 );  
753 );  
754 );  
755 );  
756 );  
757 );  
758 );  
759 );  
760 );  
761 );  
762 );  
763 );  
764 );  
765 );  
766 );  
767 );  
768 );  
769 );  
770 );  
771 );  
772 );  
773 );  
774 );  
775 );  
776 );  
777 );  
778 );  
779 );  
780 );  
781 );  
782 );  
783 );  
784 );  
785 );  
786 );  
787 );  
788 );  
789 );  
790 );  
791 );  
792 );  
793 );  
794 );  
795 );  
796 );  
797 );  
798 );  
799 );  
800 );  
801 );  
802 );  
803 );  
804 );  
805 );  
806 );  
807 );  
808 );  
809 );  
810 );  
811 );  
812 );  
813 );  
814 );  
815 );  
816 );  
817 );  
818 );  
819 );  
820 );  
821 );  
822 );  
823 );  
824 );  
825 );  
826 );  
827 );  
828 );  
829 );  
830 );  
831 );  
832 );  
833 );  
834 );  
835 );  
836 );  
837 );  
838 );  
839 );  
840 );  
841 );  
842 );  
843 );  
844 );  
845 );  
846 );  
847 );  
848 );  
849 );  
850 );  
851 );  
852 );  
853 );  
854 );  
855 );  
856 );  
857 );  
858 );  
859 );  
860 );  
861 );  
862 );  
863 );  
864 );  
865 );  
866 );  
867 );  
868 );  
869 );  
870 );  
871 );  
872 );  
873 );  
874 );  
875 );  
876 );  
877 );  
878 );  
879 );  
880 );  
881 );  
882 );  
883 );  
884 );  
885 );  
886 );  
887 );  
888 );  
889 );  
890 );  
891 );  
892 );  
893 );  
894 );  
895 );  
896 );  
897 );  
898 );  
899 );  
900 );  
901 );  
902 );  
903 );  
904 );  
905 );  
906 );  
907 );  
908 );  
909 );  
910 );  
911 );  
912 );  
913 );  
914 );  
915 );  
916 );  
917 );  
918 );  
919 );  
920 );  
921 );  
922 );  
923 );  
924 );  
925 );  
926 );  
927 );  
928 );  
929 );  
930 );  
931 );  
932 );  
933 );  
934 );  
935 );  
936 );  
937 );  
938 );  
939 );  
940 );  
941 );  
942 );  
943 );  
944 );  
945 );  
946 );  
947 );  
948 );  
949 );  
950 );  
951 );  
952 );  
953 );  
954 );  
955 );  
956 );  
957 );  
958 );  
959 );  
960 );  
961 );  
962 );  
963 );  
964 );  
965 );  
966 );  
967 );  
968 );  
969 );  
970 );  
971 );  
972 );  
973 );  
974 );  
975 );  
976 );  
977 );  
978 );  
979 );  
980 );  
981 );  
982 );  
983 );  
984 );  
985 );  
986 );  
987 );  
988 );  
989 );  
990 );  
991 );  
992 );  
993 );  
994 );  
995 );  
996 );  
997 );  
998 );  
999 );  
999 );
```

```

194        });
195    },
196    child: const Icon(Icons.arrow_back),
197  ),
198  const SizedBox(width: 10),
199  ElevatedButton(
200    onPressed: () {
201      setState(() {
202        final newX = handX + 4.0;
203        if (!checkCollision(newX, handY, handZ)) {
204          handX = newX;
205          if (isGrabbed) earthX = handX +
206            grabbedOffsetX;
207        }
208      });
209    },
210    child: const Icon(Icons.arrow_forward),
211  ],
212  ],
213  ElevatedButton(
214    onPressed: () {
215      setState(() {
216        final newY = handY - 4.0;
217        if (!checkCollision(handX, newY, handZ)) {
218          handY = newY;
219          if (isGrabbed) earthY = handY +
220            grabbedOffsetY;
221        }
222      });
223    },
224    child: const Icon(Icons.arrow_downward),
225  ],
226  ],
227  const SizedBox(width: 30),
228  Column(
229    children: [
230      const Text("Z", style: TextStyle(fontSize: 12,
231      fontWeight: FontWeight.bold)),
232      const SizedBox(height: 5),
233      ElevatedButton(
234        onPressed: () {
235          setState(() {

```

```

236             if (!checkCollision(handX, handY, newZ)) {
237                 handZ = newZ;
238                 if (isGrabbed) earthZ = handZ +
239                     grabbedOffsetZ;
240                     }
241                     });
242                     },
243                     style: ElevatedButton.styleFrom(padding: const
244                     EdgeInsets.symmetric(horizontal: 20, vertical: 15)),
245                     child: const Text("    ", style: TextStyle(fontSize:
246                     16)),
247                     ),
248                     const SizedBox(height: 5),
249                     ElevatedButton(
250                     onPressed: () {
251                     setState(() {
252                     final newZ = handZ - 4.0;
253                     if (!checkCollision(handX, handY, newZ)) {
254                     handZ = newZ;
255                     if (isGrabbed) earthZ = handZ +
256                     grabbedOffsetZ;
257                     }
258                     }
259                     );
260                     },
261                     ],
262                     ),
263                     ],
264                     const SizedBox(height: 15),
265
266                     // GRAB/RELEASE
267                     const Text("                ", style:
268                     TextStyle(fontWeight: FontWeight.bold)),
269                     const SizedBox(height: 10),
270                     Row(
271                     mainAxisAlignment: MainAxisAlignment.center,
272                     children: [
273                     ElevatedButton(

```

```

274     setState(() {
275         indexAngle = middleAngle = ringAngle = pinkyAngle =
276         12.0;
277         if (canGrab() && !isGrabbed) {
278             isGrabbed = true;
279             grabbedOffsetX = earthX - handX;
280             grabbedOffsetY = earthY - handY;
281             grabbedOffsetZ = earthZ - handZ;
282         }
283     },
284     style: ElevatedButton.styleFrom(
285         backgroundColor: Colors.red,
286         padding: const EdgeInsets.symmetric(horizontal: 30,
287         vertical: 15),
288         ),
289         child: const Text("GRAB", style: TextStyle(fontSize:
290         18, fontWeight: FontWeight.bold)),
291         ),
292         const SizedBox(width: 20),
293         ElevatedButton(
294             onPressed: () {
295                 setState(() {
296                     indexAngle = middleAngle = ringAngle = pinkyAngle =
297                     0.0;
298                     isGrabbed = false;
299                 });
300             },
301             style: ElevatedButton.styleFrom(
302                 backgroundColor: Colors.green,
303                 padding: const EdgeInsets.symmetric(horizontal: 30,
304                 vertical: 15),
305                 ),
306                 child: const Text("RELEASE", style: TextStyle(fontSize:
307                 18, fontWeight: FontWeight.bold)),
308                 ),
309                 ],
310                 ),
311
312         const SizedBox(height: 15),
313
314         // Slider(
315         value: indexAngle, min: 0, max: 12, divisions: 13,
316         label: (180 - 10 * indexAngle.round()).toString(),

```

```

313     onChanged: (v) => setState(() => indexAngle = v),
314   ),
315   Slider(
316     value: middleAngle, min: 0, max: 12, divisions: 13,
317     label: (180 - 10 * middleAngle.round()).toString(),
318     onChanged: (v) => setState(() => middleAngle = v),
319   ),
320   Slider(
321     value: ringAngle, min: 0, max: 12, divisions: 13,
322     label: (180 - 10 * ringAngle.round()).toString(),
323     onChanged: (v) => setState(() => ringAngle = v),
324   ),
325   Slider(
326     value: pinkyAngle, min: 0, max: 12, divisions: 13,
327     label: (180 - 10 * pinkyAngle.round()).toString(),
328     onChanged: (v) => setState(() => pinkyAngle = v),
329   ),
330 ],
331 ),
332 )
333 ];
334 } else{
335   children = <Widget>[
336   const Padding(
337     padding: EdgeInsets.all(8.0),
338     child: Text("Failed to load"),
339   )
340 ];
341 }
342 return Expanded(
343   child: Column(
344     mainAxisAlignment: MainAxisAlignment.center
345     ,
346     children: children,
347   ),
348 );
349 ],
350 ),
351 ),
352 );
353 );
354 }
355 }
356

```

```
357 Future<List<Mesh3D>> _generatePoints() async {
358     return [
359         Mesh3D(await ObjParser().loadFromResources("assets/hand/hand.obj")),
360         Mesh3D(await ObjParser().loadFromResources("assets/hand/index.obj")),
361         Mesh3D(await ObjParser().loadFromResources("assets/hand/middle.obj")),
362         Mesh3D(await ObjParser().loadFromResources("assets/hand/ring.obj")),
363         Mesh3D(await ObjParser().loadFromResources("assets/hand/pinky.obj")),
364         Mesh3D(await ObjParser().loadFromResources("assets/hand/skull.obj")),
365     ];
366 }
```

Результат запуска представлен на рисунке 1.

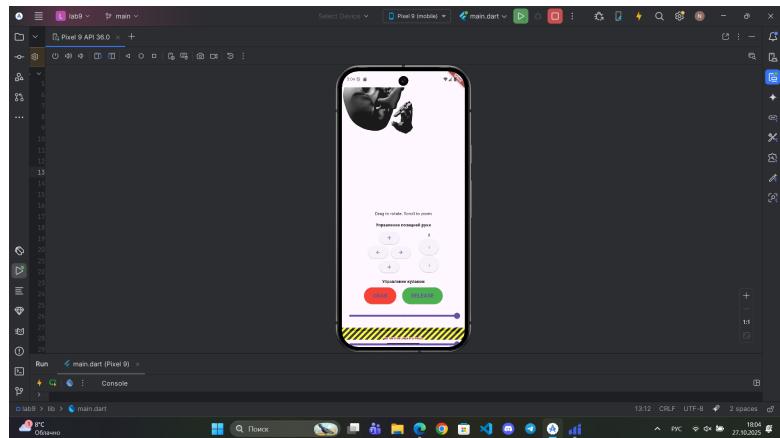


Рис. 1 — Результат