**实现物体绕不同轴旋转 并调用外部函数**

Posted on 2013年04月28日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 128 次

第一个和第二脚本赋给目标物体；第三个脚本赋给任何一个物体作为测试物体使用。代码目的是方便外部调用和函数重用。注意isRotating参数的使用。

第一个文件，声明枚举类型，分别为均匀变化和加速变化

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | **using** UnityEngine; |
| 02 |  |
| 03 | **using** System.Collections; |
| 04 |  |
| 05 | **public** **enum** CTRotationType |
| 06 |  |
| 07 | { |
| 08 |  |
| 09 | Uniform, |
| 10 |  |
| 11 | AccelerateUniformly |
| 12 |  |
| 13 | } |
| 14 |  |

第二个文件：主函数，实现围绕轴变化的两个函数，分别为均匀变化和加速变化

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 001 | **using** UnityEngine; |
| 002 |  |
| 003 | **using** System.Collections; |
| 004 |  |
| 005 | **public** **class** CTRotation : MonoBehaviour { |
| 006 |  |
| 007 | *// Use this for initialization* |
| 008 |  |
| 009 | **void** Start () { |
| 010 |  |
| 011 | } |
| 012 |  |
| 013 | *// Update is called once per frame* |
| 014 |  |
| 015 | **void** Update () { |
| 016 |  |
| 017 | **if** (isRotating) |
| 018 |  |
| 019 | { |
| 020 |  |
| 021 | executeRotate(); |
| 022 |  |
| 023 | } |
| 024 |  |
| 025 | } |
| 026 |  |
| 027 | **bool** isRotating = **false**; |
| 028 |  |
| 029 | Quaternion definedRotation = new Quaternion(0, 0, 0,0); |
| 030 |  |
| 031 | Vector3 rotateVector = new Vector3(1,0,0); |
| 032 |  |
| 033 | **float** rotateVelocity = 0; |
| 034 |  |
| 035 | **float** accelerateDuration = 0; |
| 036 |  |
| 037 | **float** leftDuration = 0; |
| 038 |  |
| 039 | **float** rotateDuration = 0; |
| 040 |  |
| 041 | **int** rotateAxis = 0; |
| 042 |  |
| 043 | **float** angleRange = 0; |
| 044 |  |
| 045 | **float** deltaRotate = 0;*//0;* |
| 046 |  |
| 047 | *// acceleration when it is in the accelerating process.* |
| 048 |  |
| 049 | **float** rotateAcceleration = 0; |
| 050 |  |
| 051 | CTRotationType rotateType; |
| 052 |  |
| 053 | *//int RotateType = 0;* |
| 054 |  |
| 055 | **private** **void** initRotateArgument( **float** \_initAngleRange, **int** \_initRotateAxis, **float** \_initRotateDuration) |
| 056 |  |
| 057 | { |
| 058 |  |
| 059 | rotateAxis = \_initRotateAxis; |
| 060 |  |
| 061 | rotateDuration = \_initRotateDuration; |
| 062 |  |
| 063 | leftDuration = \_initRotateDuration; |
| 064 |  |
| 065 | angleRange = \_initAngleRange; |
| 066 |  |
| 067 | rotateType = CTRotationType.Uniform; |
| 068 |  |
| 069 | } |
| 070 |  |
| 071 | **public** **void** RotateTo(**float** \_angleRange, **int** \_axis, **float** \_duration) |
| 072 |  |
| 073 | { |
| 074 |  |
| 075 | print(“**in** the rotateto”); |
| 076 |  |
| 077 | isRotating = **false**; |
| 078 |  |
| 079 | rotateType = CTRotationType.Uniform; |
| 080 |  |
| 081 | *//RotateType = 0;* |
| 082 |  |
| 083 | initRotateArgument(\_angleRange, \_axis, \_duration); |
| 084 |  |
| 085 | **switch**(rotateAxis) |
| 086 |  |
| 087 | { |
| 088 |  |
| 089 | **case** 0: *//rotate around X axis* |
| 090 |  |
| 091 | { |
| 092 |  |
| 093 | rotateVector = Vector3.right; |
| 094 |  |
| 095 | **break**; |
| 096 |  |
| 097 | } |
| 098 |  |
| 099 | **case** 1:*//rotate around Y axis* |
| 100 |  |
| 101 | { |
| 102 |  |
| 103 | rotateVector = Vector3.up; |
| 104 |  |
| 105 | **break**; |
| 106 |  |
| 107 | } |
| 108 |  |
| 109 | **case** 2:*//rotate around Z axis* |
| 110 |  |
| 111 | { |
| 112 |  |
| 113 | rotateVector = Vector3.forward; |
| 114 |  |
| 115 | **break**; |
| 116 |  |
| 117 | } |
| 118 |  |
| 119 | **default**: |
| 120 |  |
| 121 | **break**; |
| 122 |  |
| 123 | } |
| 124 |  |
| 125 | deltaRotate = angleRange/rotateDuration; |
| 126 |  |
| 127 | isRotating = **true**; |
| 128 |  |
| 129 | } |
| 130 |  |
| 131 | **public** **void** RotateTo(**float** \_angleRange, **int** \_axis, **float** \_duration, **float** \_accelerateDuration) |
| 132 |  |
| 133 | { |
| 134 |  |
| 135 | isRotating = **false**; |
| 136 |  |
| 137 | rotateType = CTRotationType.AccelerateUniformly; |
| 138 |  |
| 139 | *//RotateType = 1;* |
| 140 |  |
| 141 | rotateAcceleration = 1/((rotateDuration - accelerateDuration)\*accelerateDuration); |
| 142 |  |
| 143 | initRotateArgument(\_angleRange, \_axis, \_duration); |
| 144 |  |
| 145 | **switch**(rotateAxis) |
| 146 |  |
| 147 | { |
| 148 |  |
| 149 | **case** 0: *//rotate around X axis* |
| 150 |  |
| 151 | { |
| 152 |  |
| 153 | rotateVector = Vector3.right; |
| 154 |  |
| 155 | **break**; |
| 156 |  |
| 157 | } |
| 158 |  |
| 159 | **case** 1:*//rotate around Y axis* |
| 160 |  |
| 161 | { |
| 162 |  |
| 163 | rotateVector = Vector3.up; |
| 164 |  |
| 165 | **break**; |
| 166 |  |
| 167 | } |
| 168 |  |
| 169 | **case** 2:*//rotate around Z axis* |
| 170 |  |
| 171 | { |
| 172 |  |
| 173 | rotateVector = Vector3.forward; |
| 174 |  |
| 175 | **break**; |
| 176 |  |
| 177 | } |
| 178 |  |
| 179 | **default**: |
| 180 |  |
| 181 | **break**; |
| 182 |  |
| 183 | } |
| 184 |  |
| 185 | accelerateDuration = \_accelerateDuration; |
| 186 |  |
| 187 | *// deltaRotate = angleRange/(\_duration - \_accelerateDuration\*2);* |
| 188 |  |
| 189 | isRotating = **true**; |
| 190 |  |
| 191 | } |
| 192 |  |
| 193 | **void** executeRotate() |
| 194 |  |
| 195 | { |
| 196 |  |
| 197 | **switch** (rotateType) |
| 198 |  |
| 199 | { |
| 200 |  |
| 201 | *//case 0://CTMoveType.Uniform:* |
| 202 |  |
| 203 | **case** CTRotationType.Uniform: |
| 204 |  |
| 205 | uniformRotate(); |
| 206 |  |
| 207 | **break**; |
| 208 |  |
| 209 | *//case 1://CTMoveType.AccelerateUniformly:* |
| 210 |  |
| 211 | **case** CTRotationType.AccelerateUniformly: |
| 212 |  |
| 213 | accelerateRotate(); |
| 214 |  |
| 215 | **break**; |
| 216 |  |
| 217 | } |
| 218 |  |
| 219 | leftDuration -= Time.deltaTime; |
| 220 |  |
| 221 | */\* if (leftDuration <= 0)* |
| 222 |  |
| 223 | *{* |
| 224 |  |
| 225 | *transform.position = targetPosition;* |
| 226 |  |
| 227 | *isMoving = false;* |
| 228 |  |
| 229 | *}\*/* |
| 230 |  |
| 231 | } |
| 232 |  |
| 233 | **private** **void** accelerateRotate() |
| 234 |  |
| 235 | { |
| 236 |  |
| 237 | print(leftDuration); |
| 238 |  |
| 239 | **if** (leftDuration > (rotateDuration - accelerateDuration)) |
| 240 |  |
| 241 | { |
| 242 |  |
| 243 | rotateVelocity = (**float**)((angleRange\*(rotateDuration - leftDuration))\*rotateAcceleration); |
| 244 |  |
| 245 | *// transform.Rotate(rotateVelocity \* Time.deltaTime\*rotateVector, Space.World);* |
| 246 |  |
| 247 | transform.Rotate(rotateVelocity \* rotateVector\*Time.deltaTime, Space.World); |
| 248 |  |
| 249 | } |
| 250 |  |
| 251 | **else** **if** (leftDuration > accelerateDuration) |
| 252 |  |
| 253 | { |
| 254 |  |
| 255 | rotateVelocity = (**float**)((angleRange\*accelerateDuration)\*rotateAcceleration); |
| 256 |  |
| 257 | transform.Rotate(rotateVelocity\*rotateVector\*Time.deltaTime, Space.World); |
| 258 |  |
| 259 | } |
| 260 |  |
| 261 | **else** **if** (leftDuration > 0) |
| 262 |  |
| 263 | { |
| 264 |  |
| 265 | rotateVelocity= (**float**)((angleRange\*leftDuration)\*rotateAcceleration); |
| 266 |  |
| 267 | transform.Rotate(rotateVelocity\*rotateVector\*Time.deltaTime, Space.World); |
| 268 |  |
| 269 | } |
| 270 |  |
| 271 | **else** |
| 272 |  |
| 273 | isRotating = **false**; |
| 274 |  |
| 275 | } |
| 276 |  |
| 277 | **private** **void** uniformRotate() |
| 278 |  |
| 279 | { |
| 280 |  |
| 281 | print(leftDuration); |
| 282 |  |
| 283 | *//if(leftDuration)* |
| 284 |  |
| 285 | **if**(leftDuration > 0) |
| 286 |  |
| 287 | { |
| 288 |  |
| 289 | transform.Rotate(rotateVector\*deltaRotate\*Time.deltaTime, Space.World); |
| 290 |  |
| 291 | *//transform.Rotate(rotateVector \* Time.deltaTime\*deltaRotate, Space.World);* |
| 292 |  |
| 293 | } |
| 294 |  |
| 295 | **else** |
| 296 |  |
| 297 | isRotating = **false**; |
| 298 |  |
| 299 | } |
| 300 |  |
| 301 | } |

第三个文件，测试脚本

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | **using** UnityEngine; |
| 02 |  |
| 03 | **using** System.Collections; |
| 04 |  |
| 05 | **public** **class** TestRotationScript : MonoBehaviour { |
| 06 |  |
| 07 | *// Use this for initialization* |
| 08 |  |
| 09 | **void** Start () { |
| 10 |  |
| 11 | } |
| 12 |  |
| 13 | *// Update is called once per frame* |
| 14 |  |
| 15 | **void** Update () { |
| 16 |  |
| 17 | } |
| 18 |  |
| 19 | **void** OnGUI () |
| 20 |  |
| 21 | { |
| 22 |  |
| 23 | CTRotation ttscript; |
| 24 |  |
| 25 | CTChangeAlpha colorScript; |
| 26 |  |
| 27 | GameObject testObject = GameObject.Find(“TestCube”); |
| 28 |  |
| 29 | *//Component testObjectScript = testObject.GetComponent(“CRotation”);* |
| 30 |  |
| 31 | ttscript = (CTRotation)testObject.GetComponent(“CTRotation”); |
| 32 |  |
| 33 | colorScript = (CTChangeAlpha)testObject.GetComponent(“CTChangeAlpha”); |
| 34 |  |
| 35 | **if** (GUI.Button (new Rect (20,40,80,20), “UniRotate”)) { |
| 36 |  |
| 37 | ttscript.RotateTo(3600f, 2, 2f); |
| 38 |  |
| 39 | } |
| 40 |  |
| 41 | **if**(GUI.Button(new Rect(20,60,80,20),“AccRotate”)){ |
| 42 |  |
| 43 | ttscript.RotateTo(3600f, 2, 2f, 0.5f); |
| 44 |  |
| 45 | } |
| 46 |  |
| 47 | **if**(GUI.Button(new Rect(20,80,80,20),“Color”)){ |
| 48 |  |
| 49 | colorScript.ColorTo(2,5.0f); |
| 50 |  |
| 51 | } |
| 52 |  |
| 53 | } |
| 54 |  |
| 55 | } |