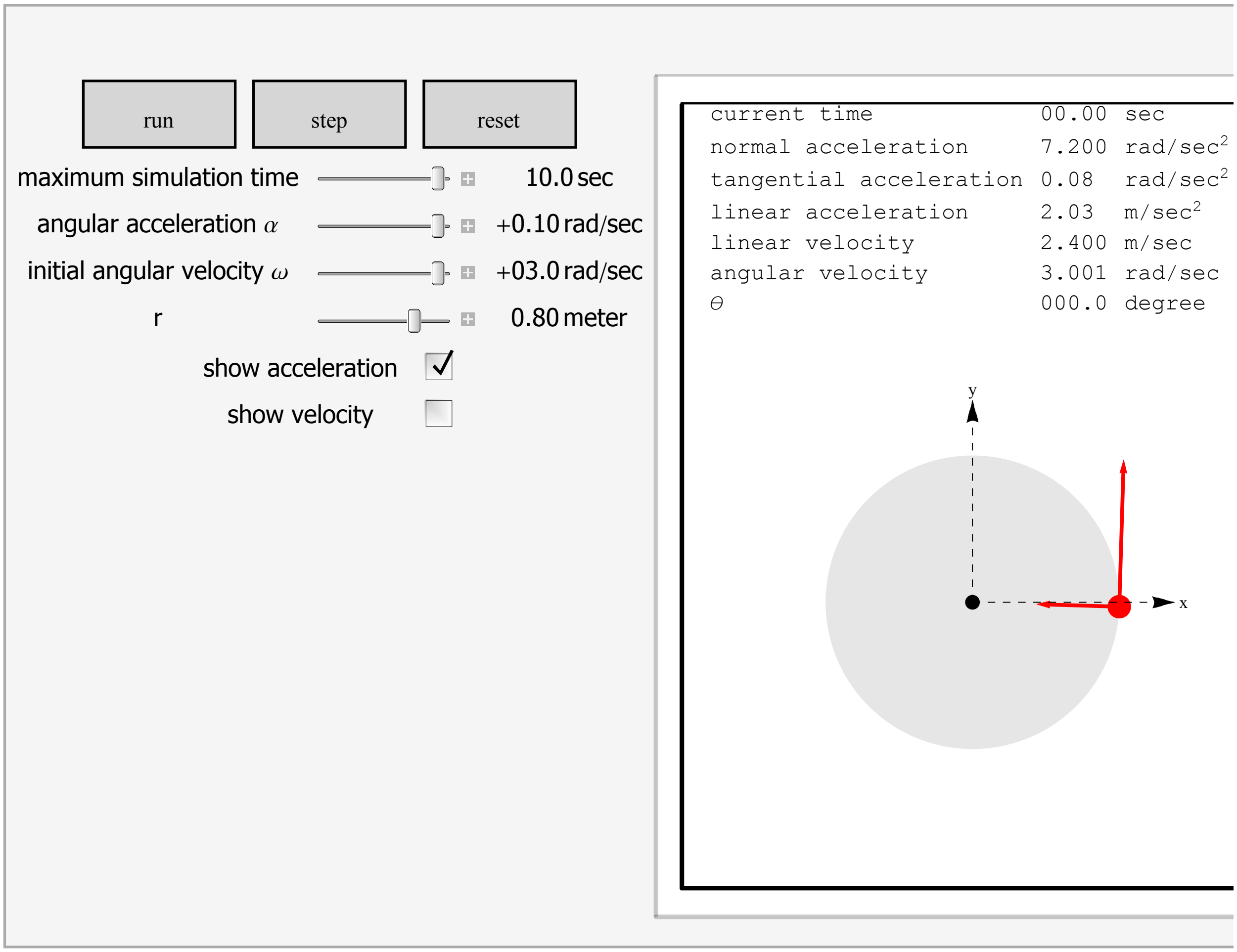
```
(*By Nasser M. Abbasi. Show constant angular velocity and acceleration motion
 1/4/2014*)

Manipulate[
 tick;
 Module[{x, y, linearAcc, maxOmega = 3 + 0.1 * maxSimulationTime,
   maxR = 1, maxAlpha = 0.1, at, an, v, normaAcc, tanAcc, debug = False},

  If[debug, Print["calling makeStep ", state]];
  If[debug, Print["stat is ", state]];
  If[debug, Print["BEFORE, currentTime= ", currentTime, "\ncurrentTheta= ", currentTheta,
    "\ncurrentOmega= ", currentOmega, "\nalpha= ", alpha, "\ndelT= ", delT, "\nr= ",
    r, "\nmaxOmega= ", maxOmega, "\nmaxR= ", maxR, "\nmaxAlpha= ", maxAlpha]];

  {currentSpeed, currentTheta, currentOmega, linearAcc, normaAcc, tanAcc, at, an, v, x, y} =
   makeStep[currentTheta, currentOmega, alpha, delT, r, maxOmega, maxR, maxAlpha];
  currentTime += delT;
  If[debug, Print["\n\nAFTER, currentTime= ", currentTime, "\ncurrentTheta= ",
    currentTheta, "\ncurrentOmega= ", currentOmega, "\nalpha= ", alpha, "\ndelT= ",
    delT, "\nr= ", r, "\nmaxOmega= ", maxOmega, "\nmaxR= ", maxR, "\nmaxAlpha= ",
    maxAlpha, "\nat= ", at, "\nan= ", an, "\nx= ", x, "\ny= ", y, "\nv= ", v]];


  g = Grid[{
     {
      Grid[{
        {"current time", padIt2[currentTime, {4, 2}] , "sec"},
        {"normal acceleration", padIt2[Norm@normaAcc, {4, 3}] , "rad/sec^2"},
        {"tangential acceleration", padIt2[Norm@tanAcc, {3, 2}] , "rad/sec^2"},
        {"linear acceleration ", padIt2[linearAcc, {3, 2}], "m/sec^2"},
        {"linear velocity ", padIt2[currentSpeed, {4, 3}], "m/sec"},
        {"angular velocity ", padIt2[currentOmega, {4, 3}], "rad/sec"},
        {"θ ", padIt2[currentTheta * 180 / Pi, {4, 1}], "degree"}
       }, Alignment → Left
      ]
     },
     {Graphics[
       {
        {RGBColor[{.9, .9, .9}], Disk[{0, 0}, r]},
        {Red, Disk[{x, y}, 0.08 * r]},
        Disk[{0, 0}, 0.05 * r],
        If[showVelocity , {Arrowheads → Small, Arrow[v]}],
        If[showAcc,
         {{Thick, Red, Arrowheads → Small, Arrow[an]},
          {Thick, Red, Arrowheads → Small, Arrow[at]}
         }
        ],
        (*Text[v,{1.2x,1.2y},{0,0}]*)
        (*{Thin,Dashed,Line[{{0,0},{x,y}}]},*)
        {Dashed, Arrow[{{0, 0}, {1.1, 0}}]},
        {Dashed, Arrow[{{0, 0}, {0, 1.1}}]},
       Text["x", {1.15, 0}, {0, 0}],
       Text["y", {0, 1.15}, {0, 0}]
      },
      PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}},
      ImagePadding → 5, ImageSize → 300, AspectRatio → 1
     ]
    }
   }, Spacings → {.5, .1}, Frame → True, Alignment →  Center
  ];

 Which[
  state == "running" || state == "step",
  (
   If[currentTime ≤  maxSimulationTime,
    If[state == "running",
     tick = Not[tick]
    ],
    currentTheta = -omega * delT;
    currentOmega = omega;
    currentTime = -delT
   ]
  )
  ,
  state == "reset",
  (
   currentTheta = -omega * delT;
   currentOmega = omega;
   currentTime = -delT;
   state = "step";
   tick = Not[tick]
  )
 ];

 g
],

Grid[{
  {Grid[{
     {
      Button[Text[Style["run", 12]],
       state = "running"; tick = Not[tick], ImageSize → {80, 35}],
      Button[Text[Style["step", 12]], state = "step"; tick = Not[tick],
       ImageSize → {80, 35}],
      Button[Text[Style["reset", 12]], state = "reset";
       tick = Not[tick], ImageSize → {80, 35}]
     }
    }]
  },
  {
   Grid[{
     {
      "maximum simulation time", Manipulator[Dynamic[maxSimulationTime,
        {maxSimulationTime = #; state = "reset"; tick = Not[tick]} &],
       {0.1, 10, 0.01}, ImageSize → Tiny, ContinuousAction → True],
      Row[{Dynamic@padIt2[maxSimulationTime, {3, 1}], Spacer[2], "sec"}], SpanFromLeft
     },
     {
      "angular acceleration α", Manipulator[Dynamic[alpha, {alpha = #} &],
       {-0.1, 0.1, 0.01}, ImageSize → Tiny, ContinuousAction → True],
      Row[{Dynamic@padIt1[alpha, {3, 2}], Spacer[2], "rad/sec"}], SpanFromLeft
     },
     {
      "initial angular velocity ω",
      Manipulator[Dynamic[omega, {omega = #; currentOmega = omega; tick = Not[tick]} &],
       {-3, 3, 0.1}, ImageSize → Tiny, ContinuousAction → True],
      Row[{Dynamic@padIt1[omega, {3, 1}], Spacer[2], "rad/sec"}], SpanFromLeft
     },
     {
      "r", Manipulator[Dynamic[r, {r = #; tick = Not[tick]} &],
       {0.1, 1, 0.1}, ImageSize → Tiny, ContinuousAction → True],
      Dynamic@Row[{padIt2[r, {3, 2}], Spacer[2], "meter"}], SpanFromLeft
     }
    }
   ]
  },

  {
   Grid[{
     {"show acceleration ", Checkbox[Dynamic[showAcc, {showAcc = #; tick = Not[tick]} &]]},
     {"show velocity ",
      Checkbox[Dynamic[showVelocity, {showVelocity = #; tick = Not[tick]} &]]}
    }]
  }
 }
]
,
{{tick, True}, None},
{{state, "step"}, None},
{{omega, 1.5}, None},
{{currentOmega, 1.5}, None},
{{currentTheta, 0}, None},
{{currentSpeed, 0}, None},
{{maxSimulationTime, 1}, None},
{{alpha, 0}, None},
{{r, .8}, None},
{{currentTime, 0}, None},
{{delT, 0.01}, None},

{{g, 0}, None},
{{showAcc, False}, None},
{{showVelocity, True}, None},
SynchronousUpdating → True,
ControlPlacement → Left,
TrackedSymbols :→ {tick},
Initialization :→
 (
  (*definitions used for parameter checking*)
  integerStrictPositive = (IntegerQ[#] && # > 0 &);
  integerPositive = (IntegerQ[#] && # ≥ 0 &);
  numericStrictPositive = (Element[#, Reals] && # > 0 &);
  numericPositive = (Element[#, Reals] && # ≥ 0 &);
  numericStrictNegative = (Element[#, Reals] && # < 0 &);
  numericNegative = (Element[#, Reals] && # ≤ 0 &);
  bool = (Element[#, Booleans] &);
  numeric = (Element[#, Reals] &);
  integer = (Element[#, Integers] &);
  (*--------------------------------------------*)
  padIt1[v_?numeric, f_List] := AccountingForm[v,
    f, NumberSigns → {"-", "+"}, NumberPadding → {"0", "0"}, SignPadding → True];
  (*--------------------------------------------*)
  padIt1[v_?numeric, f_Integer] := AccountingForm[Chop[v],
    f, NumberSigns → {"-", "+"}, NumberPadding → {"0", "0"}, SignPadding → True];
  (*--------------------------------------------*)
  padIt2[v_?numeric, f_List] := AccountingForm[v,
    f, NumberSigns → {"", ""}, NumberPadding → {"0", "0"}, SignPadding → True];
  (*--------------------------------------------*)
  padIt2[v_?numeric, f_Integer] := AccountingForm[Chop[v],
    f, NumberSigns → {"", ""}, NumberPadding → {"0", "0"}, SignPadding → True];
  (*--------------------------------------------*)

  makeStep[currentAngle_, currentOmega_, alpha_, delT_, r_, maxOmega_, maxR_, maxAlpha_] :=
   Module[{$currentAngle = currentAngle, $currentSpeed, $currentOmega = currentOmega,
     x, y, v0, v, v0Normalized, normaAcc, normaAccNormalized, an,
     tanAcc, tanAccNormalized, at, linearAcc, debug = False},

    x = r Cos[$currentAngle];
    y = r Sin[$currentAngle];

    $currentSpeed = {-$currentOmega y, $currentOmega x};
    v0Normalized = {-$currentOmega y, $currentOmega x} / (maxOmega maxR);
    v = {{x, y}, {x, y} + v0Normalized};

    normaAcc = {-$currentOmega^2 x, -$currentOmega^2 y};
    normaAccNormalized = normaAcc / (maxOmega^2 maxR);
    If[debug, Print["normaAcc=", normaAcc,
      "\nnormaAccNormalized=", normaAccNormalized, "\n(maxOmega^2 maxR)=",
      (maxOmega^2 maxR), "\nNorm[normaAcc]= ", Norm[normaAcc]]];
    an = {{x, y}, {x, y} + normaAccNormalized};

    tanAcc = {-alpha y, alpha x};
    tanAccNormalized = tanAcc / (maxAlpha maxR);
    at = {{x, y}, {x, y} + tanAccNormalized};
    linearAcc = Norm[at + an];
    $currentAngle += $currentOmega * delT + 1 / 2 alpha * delT^2;
    $currentOmega += alpha * delT;
```

```
      {Norm@$currentSpeed, Mod[$currentAngle, 2 Pi],
       $currentOmega, linearAcc, normaAcc, tanAcc, at, an, v, x, y}


     ]
   )

]
```



maximum simulation time 10.0 sec

angular acceleration α +0.10 rad/sec

initial angular velocity ω +03.0 rad/sec

r 0.80 meter

show acceleration ☑

show velocity □

current time 00.00 sec

normal acceleration 7.200 rad/sec^2

tangential acceleration 0.08 rad/sec^2

linear acceleration 2.03 m/sec^2

linear velocity 2.400 m/sec

angular velocity 3.001 rad/sec

θ 000.0 degree