Esercizio su OpenGL

Di seguito si riporta un programma che effettua una simulazione video del moto uniforme unidimensionale di una particella.
Modificare il programma in modo che venga rappresentato un moto parabolico e/o un moto circolare uniforme. In entrambi i casi è importante modificare opportunamente:

- la legge oraria, riguardando le funzioni **`updateFrame()`** e **`display();`**
- le condizioni iniziali del moto, impostate nel **`main( )`** ;
- le condizioni per terminare l'animazione, descritte in **`timerFunction(int timerID).`**

Per compilare usare: **`gcc -Wall -pedantic es1.c -o es1 -lGL -lm -lglut`**

CODICE

```c
#include <GL/glut.h>
#include <stdio.h>
#include <math.h>
#define PI 3.141592653

// ----------------------- variabili globali -------------------------------------

double radius = 0.03;  // raggio della pallina da disegnare
int frameNumber = 0;
int animating = 0;
double posx;
double posx0;
int nstep = 2000; // numero massimo di step per l'animazione, con 2000 dura un minuto.
int dt = 30;      // intervallo di tempo tra un frame e l'altro, in millisecondi.

// ----------------------- prototipi di funzioni ---------------------------------

void key(unsigned char ch, int x, int y); // Descrive come gestire i comandi da keyboard
void display();                           // Funzione che descrive i pixel da stampare a video
void timerFunction(int timerID);     // Funzione che controlla l'avanzamento dell'animazione
void updateFrame();                  // Funzione che descrive come aggiornare l'animazione

// ---------------------- programma main con inizializzazione --------------------

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE);
    glutInitWindowSize(500,500);
    glutInitWindowPosition(100,100);
    glutCreateWindow("OpenGL - Basic Draw");
    glutDisplayFunc(display);
    glutKeyboardFunc(key);
    posx0 = - 0.9;
    updateFrame();
    glutMainLoop();
    return 0;
}
```

```
// ----------------------- funzioni per la parte grafica -------------------------

void key(unsigned char ch, int x, int y){
    if ( ch == ' ' && animating==0 ) {
        animating = 1;
        glutTimerFunc(dt, timerFunction, 0); }
}

void display() {
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_POLYGON);
    for(double i = 0; i < 2*PI; i += PI/20) {//<- Aumenta 20 se vuoi una pallina più liscia
        glVertex3f(cos(i) * radius + posx, sin(i) * radius, 0.0);}
    glEnd();
    glutSwapBuffers();
}

void timerFunction(int timerID) {
    if (frameNumber>nstep) {
       animating=0;
       return;}
    if (posx>0.9||posx<-0.9) {
      animating=0;
      return;}
    if (animating == 1) {
        updateFrame();
        glutTimerFunc(dt, timerFunction, 0);
        glutPostRedisplay();}
}

void updateFrame() {
   posx = frameNumber*dt/10000.0 + posx0;
   frameNumber++;
}
```