

تمامی دستورات کامنت گذاری شده تا در حد مطلوب قابل فهم باشد .

مدل T فرض شده :

```
point A[9] = { { -0.125f, -0.5f }, { -0.125f, 0.0f }, { -0.25f, 0.0f }, { -0.25f, 0.25f }, { 0.0f, 0.25f }, { 0.25f, 0.25f }, { 0.25f, 0.0f }, { 0.125f, 0.0f }, { 0.125f, -0.5f } }; // T Model
```

مدل خانه فرض شده :

```
point B[9] = { { -0.125f, 0 }, { -0.125f, -0.5f }, { -0.25f, -0.5f }, { -0.25f, 0.25f }, { 0.0f, 0.75f }, { 0.25f, 0.25f }, { 0.25f, -0.5f }, { 0.125f, -0.5f }, { 0.125f, 0 } }; // Home Model
```

مدل 9 ضلعی منتظم برای بخش شکل دلخواه :

```
point C[9] = { { 0.0f, 1.0f }, { 0.643f, 0.766f }, { 0.985f, 0.174f }, { 0.866f, -0.5f }, { 0.342f, -0.940f }, { -0.342f, -0.940f }, { -0.866f, -0.5f }, { -0.985f, 0.174f }, { -0.643f, 0.766f } }; // Regular 9-Sided Model
```

که البته مدل 9 ضلعی موقع نمایش در عدد $\frac{1}{2}$ ضرب می شود تا شکل از کادر بیرون زده نشود .

تابع display برای نمایش استفاده می شود و با استفاده از قابلیت idle هر بار توسط تابع AnimateScene دوباره صدا زده می شود .

آرایه T در AnimateScene با توجه به state ای که توسط تابع Keyboard و صفحه کلید تعیین شده است مقدار دهی می شود و سپس با تابع display نمایش داده می شود .

درون یابی به نحو صحیح توسط فرمول زیر انجام شده است .

```
t++;
if (t > frame)
t = frame;

for (int i = 0; i < 9; i++)
{
    T[i].x = A[i].x + (t / frame)*(B[i].x - A[i].x);
    T[i].y = A[i].y + (t / frame)*(B[i].y - A[i].y);
}
```

نحوه استفاده از کد :

پس از اجرای موفقیت آمیز در ابتدا شکل T رسم خواهد شد با زدن دکمه c شکل خانه ترسیم می شود با زدن دکمه t شکل T مجددا رسم می شود .

با زدن دکمه d شکل از T به خانه تبدیل می شود و با زدن دکمه p شکل از خانه به 9 ضلعی منتظم تبدیل می شود .

