Chaos Game

Description: Start with an equilateral triangle with vertices 1, 2 and 3. Pick an arbitrary point \( P_0 \) in the plane, then choose one of the triangle vertices at random and let \( P_1 \) be the point halfway between \( P_0 \) and the chosen vertex; then again randomly choose the vertex and let \( P_2 \) be the point halfway between \( P_1 \) and chosen vertex. Continuing this way yields an infinite sequence \( \{P_n\} \).

To program the Chaos Game we assume that our triangle is

\[
\text{tr} = \text{Line}[\{(0, 0), (1/2, \text{Sqrt}[3]/2), (1, 0), (0, 0)\}];
\]

We define three functions, which gives the halfway from a given vertex to the point \( P_i \):

\[
\begin{align*}
\text{f1}[x_] & := x/2 \\
\text{f2}[x_] & := (x + (1, 0))/2 \\
\text{f3}[x_] & := (x + (1/2, \text{Sqrt}[3]/2))/2
\end{align*}
\]

On each iteration we pick one of these functions at random:

\[
\text{f}[x_] := \{\text{f1}, \text{f2}, \text{f3}\}[[\text{Random[Integer, \{1, 3\}]}]][x]
\]

and iterate \( f \) starting with the arbitrary point:

\[
\text{NestList}[f, \{0.1, 0.5\}, 10]
\]

To order to produce a picture, the above pairs should be transformed to the primitive \text{Point}. 
Point /@ %

Point[{{0.1, 0.5}}, Point[{{0.55, 0.25}}, Point[{{0.275, 0.125}},
Point[{{0.1375, 0.0625}}, Point[{{0.06875, 0.03125}},
Point[{{0.284375, 0.448638}}, Point[{{0.392188, 0.657332}},
Point[{{0.696094, 0.328666}}, Point[{{0.348047, 0.164333}},
Point[{{0.424023, 0.515179}}, Point[{{0.462012, 0.690602}}]

Show[Graphics[{{tr, %}}]]

Finally we combine all said above into a function

Game[start_, n_] := Show[Graphics[Map[Point, NestList[f, start, n]]]]

Game[{{0.1, 0.5}}, 10^4]
Change the starting point

\begin{verbatim}
Game[{1., 0.5}, 10^4]
\end{verbatim}

- Graphics -

What we see here that the Chaos Game converges to the Sierpinski triangle. Actually, if we ignore a few first points, our triangle is not distinguishable from the Sierpinski triangle.

Now we implement the deterministic algorithm. Instead of random choosing, we will take all possible choices of vertices.

\begin{verbatim}
Clear[f];
f[x_] :=
Join[
  x,
  Map[# + {1, 0} &, x],
  Map[# + {1/2, Sqrt[3]/2} &, x]]/2

f[{(0.1, 0.2)}]

{{0.05, 0.1}, {0.55, 0.1}, {0.3, 0.533013}}
\end{verbatim}

Now we iterate this function a couple of times

\begin{verbatim}
Nest[f, {{0.1, 0.2}}, 2]

{{0.025, 0.05}, {0.275, 0.05}, {0.15, 0.266506},
 {0.525, 0.05}, {0.775, 0.05}, {0.65, 0.266506},
 {0.275, 0.483013}, {0.525, 0.483013}, {0.4, 0.699519}}
\end{verbatim}

Observe, that a number of possible choices is growing exponentially: \(3^n\).

Here is a final function
\[\text{GameD[start\_\_}, n\_] := \text{Show[Graphics[Map[Point, Nest[f, \{start\_\_, n\}_]]]}\]

We draw a few pictures

\[\text{GameD[\{0.1, 0.5\}, 4]}\]

\[\text{- Graphics -}\]

\[\text{GameD[\{0.1, 0.5\}, 6]}\]

\[\text{- Graphics -}\]
The proof that our chaos game indeed converges to the Sierpinski triangle is not trivial. You have to prove that the distance (in the Hausdorff metric) between those two sets approaches zero. I refer you to the book *M. Barnsley, Fractals Everywhere, Academic Press, 1988* for details.

**Geometry of Binomial Coefficients**

Binomial coefficients are coefficients by $x$ in the following expansions:

\[
\text{Expand}[\binom{1}{1} + x^2] = 1 + 2x + x^2
\]

\[
\text{Expand}[\binom{1}{1} + x^4] = 1 + 4x + 6x^2 + 4x^3 + x^4
\]

In *Mathematica* they are represented by `Binomial`.
They form a triangle (called Pascal's triangle):

```
TableForm[Table[Binomial[i, k], {i, 0, 8}, {k, 0, i}],
TableAlignments -> Center]
```

What picture can be made out of these numbers? Consider the binomial coefficients by reducing modulo 2.
ListPlot3D[Table[Mod[Binomial[i, k], 2], {i, 0, 2^6}, {k, 0, 2^6}]]

Do you see anything? Let us put this picture upside down:

ListPlot3D[1 - Table[Mod[Binomial[i, k], 2], {i, 0, 2^6}, {k, 0, 2^6}]]
Aha! Using the graphic primitive `Raster` we can see the structure more clearly:

```plaintext
				tab = 1 -
				Table[Mod[Binomial[i, j], 2], {i, 0, 2^7}, {j, 0, 2^7}];
				Show[Graphics[Raster[tab], AspectRatio -> 1, FrameTicks -> None]]
```

This picture describes the geometrical pattern of zeroes and ones obtained by reducing modulo two each element of Pascal's triangle formed from binomial coefficients. Ones are indicated by black squares and zeroes are by white (blank) squares.

What the pattern do you see? Look at the following two outputs:

```plaintext
TableForm[StringReplace[ToString[#, 
      {"\[rightarrow]" \[rightarrow] ", " \[rightarrow] ", " \[rightarrow] " }]] & @
Table[Binomial[i, k], {i, 0, 6}, {k, 0, i}],
TableAlignments -> Center]
```

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
```
The number of odd binomial coefficients in the $n$-th row of Pascal’s triangle is $2^{W(n)}$, where $W(n)$ gives the number of occurrences of the digit 1 in the binary representation of the integer $n$.

**Julia Set**

Consider a rational function $R(z)$. The “filled-in” Julia set is the set of points $z$ which is bounded after $R(z)$ is repeatedly applied. There are two types of Julia sets: connected sets (for example, Mandelbrot set) and Cantor sets (Fatou dust).

```
Clear[JuliaSet];
JuliaSet[c_., {x1_, x2_}, {y1_, y2_}, opts___] :=
DensityPlot[-Length[
    FixedPointList[#^2 + c &, x + I y, 60, SameTest -> (Abs[#2] > 3. &)]],
{x, x1, x2}, {y, y1, y2}, opts, PlotPoints -> 225, Mesh -> False,
Frame -> False, AspectRatio -> Automatic]
```
• Douady’s Rabbit Fractal

\[
\text{JuliaSet}[-0.123 + 0.745 \, \text{i},
\{-1.5, 1.5\}, \{-1.5, 1.5\}, \text{ColorFunction} \rightarrow \text{Hue}]
\]
- 11-cycle Dragon

\[ \text{JuliaSet}[0.32 + 0.043 \, I, \{-1.5, 1.5\}, \{-1.5, 1.5\}] \]
- Siegel Disk Fractal

\[\text{JuliaSet}(-0.31 - 0.587i, \{(-1.5, 1.5), (-1.5, 1.5)\})\]
- Dendrite Fractal

\[
\text{JuliaSet}[1. \text{I}, \{-1.5, 1.5\}, \{-1.5, 1.5\}, \text{ColorFunction} \rightarrow \text{Hue}]
\]

- San Marco Fractal

\[
\text{JuliaSet}[-1., \{-2, 2\}, \{-1, 1\}]
\]
The Mandelbar Set

The Mandelbar Set is a fractal set analogous to the Mandelbrot set (as its generalization to a higher power or with the variable $z$ replaced by its complex conjugate)

```
JuliaSet[-3/4., {(-2, 2), (-1, 1)}]
```

```
DensityPlot[-Length[FixedPointList[#^4 + x + I y &, x + I y, 60, SameTest -> (Abs[#2] > 3. &)]],
{x, -2.25, 2.25}, {y, -1.5, 1.5}, PlotPoints -> 225,
Mesh -> False, Frame -> False, AspectRatio -> Automatic]
```
DensityPlot[-Length[FixedPointList[
    Conjugate[#]^4 + x + I y &], x + I y, 60, SameTest -> (Abs[#2] > 3. &)]],
{x, -2.25, 2.25}, {y, -1.5, 1.5}, PlotPoints -> 225,
Mesh -> False, Frame -> False, AspectRatio -> Automatic]
Julia Sets of Other Functions

\[
\text{DensityPlot}\left[\text{Length}\left[\text{FixedPointList}\left[\text{Conjugate}[\#^3 - 0.53 - 0.4 I \&, \\
x + I y, 60, \text{SameTest} \rightarrow (\text{Abs}[\#^2] > 3. \&)]\right]\right], \{x, -1.5, 1.5\}, \\
\{y, -1.5, 1.5\}, \text{PlotPoints} \rightarrow 225, \text{Mesh} \rightarrow \text{False}, \text{Frame} \rightarrow \text{False}, \\
\text{AspectRatio} \rightarrow \text{Automatic}, \text{ColorFunction} \rightarrow \text{Hue}\right]
\]