
チューリング・モデル

Code

```
step[] :=  
  (step1[];  
   step2[];  
   step3[];  
   step4[];  
   step5[];  
   step6[];  
  );
```

```
step1[] :=  
  v1 = Table[  
    If[v[[i]] ≥ 1, IntegerPart[v[[i]] (1 - d) - e],  
    0],  
    {i, 1, width}]
```

```
step2[] :=  
  u1 = Table[  
    If[u[[i]] == 0,  
      If[Random[Real, {0, 1}] < p,  
        1,  
        0],  
    u[[i]]],  
    {i, 1, width}]
```

```
step3[] :=  
  v2 = Table[  
    If[u1[[i]] == 1,  
      v1[[i]] + w1,  
      v1[[i]]],  
    {i, 1, width}]
```

```
index[i_] := Mod[i + width, width, 1]
```

```
nu[i_] := Total[Table[u[[index[i + j]]], {j, -ru, ru}]]
```

```

step4[] :=
u2 = Table[
  If[u1[[i]] == 0 && nu@i > Round[m0 + m1 v2[[i]]],
    1,
    u1[[i]],
  {i, 1, width}]

```

```

meanV2[i_] :=
Total[Table[v2[[index[i + j]]], {j, -rv, rv}]] / (2 rv + 1)

```

```

step5[] :=
v = Table[Round@meanV2[i], {i, 1, width}]

```

```

step6[] :=
u = Table[
  If[v[[i]] ≥ w2,
    0,
    u2[[i]],
  {i, 1, width}]

```

```

initialize[] := (
  u = Table[
    If[Random[Real, {0, 1}] < initProb, 1, 0], {width}];
  v = Table[0, {width}];
)

```

```

makePattern[] := (
  initialize[];
  pattern = {};
  AppendTo[pattern, u];
  Do[
    step[];
    AppendTo[pattern, u],
    {i, 0, history - 1}];
  ListDensityPlot[(1 - #) & /@ pattern,
    Mesh → False, FrameLabel → {"", time}];
)

```

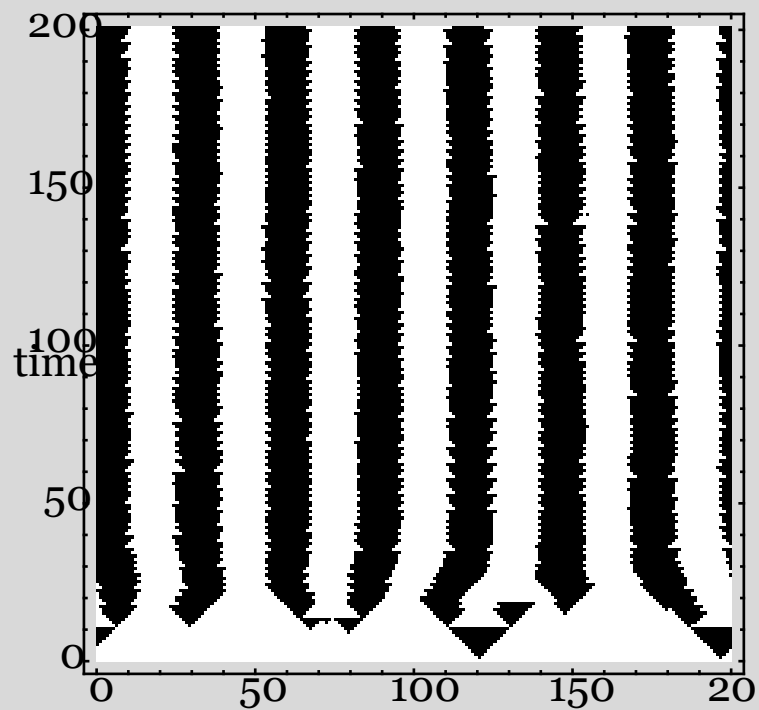
Result

```
width = 200;  
history = 200;
```

■ Pattern A

```
d = 0; e = 1; initProb = 0;  
ru = 1;  
rv = 17;  
w1 = 1;  
w2 = 1;  
m0 = m1 = 0;  
p = 0.002;
```

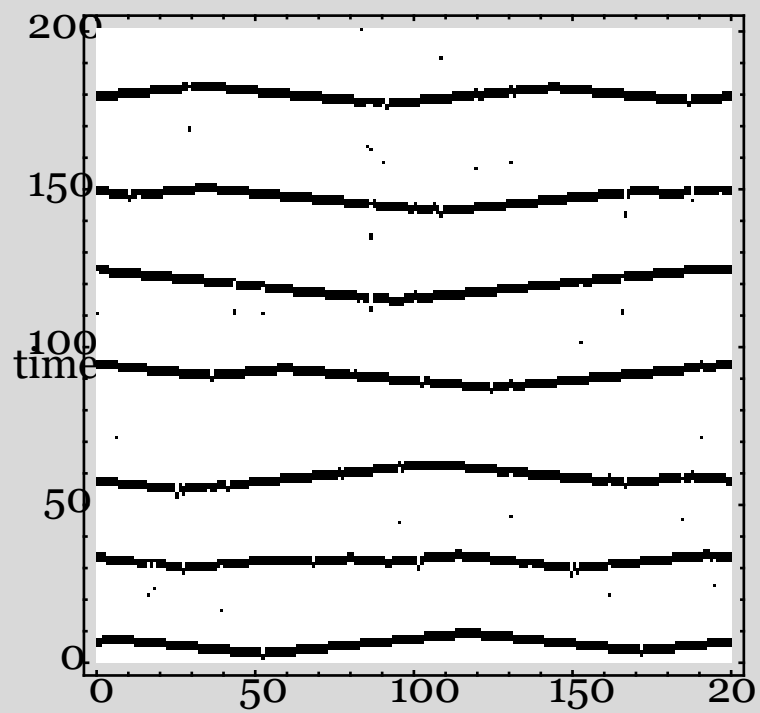
```
makePattern[]
```



■ Pattern B

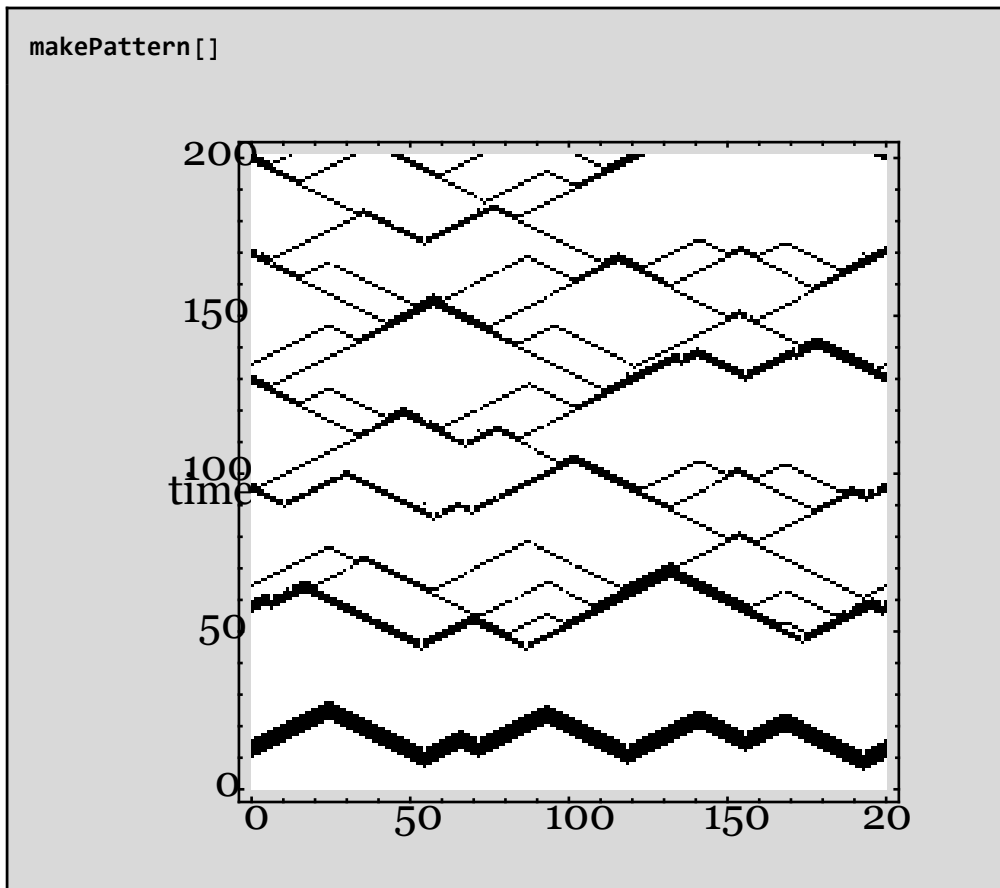
```
d = 0; e = 1; initProb = 0;  
ru = 10;  
rv = 0;  
w1 = 8;  
w2 = 21;  
m0 = 0;  
m1 = 1;  
p = 0.002;
```

makePattern[]



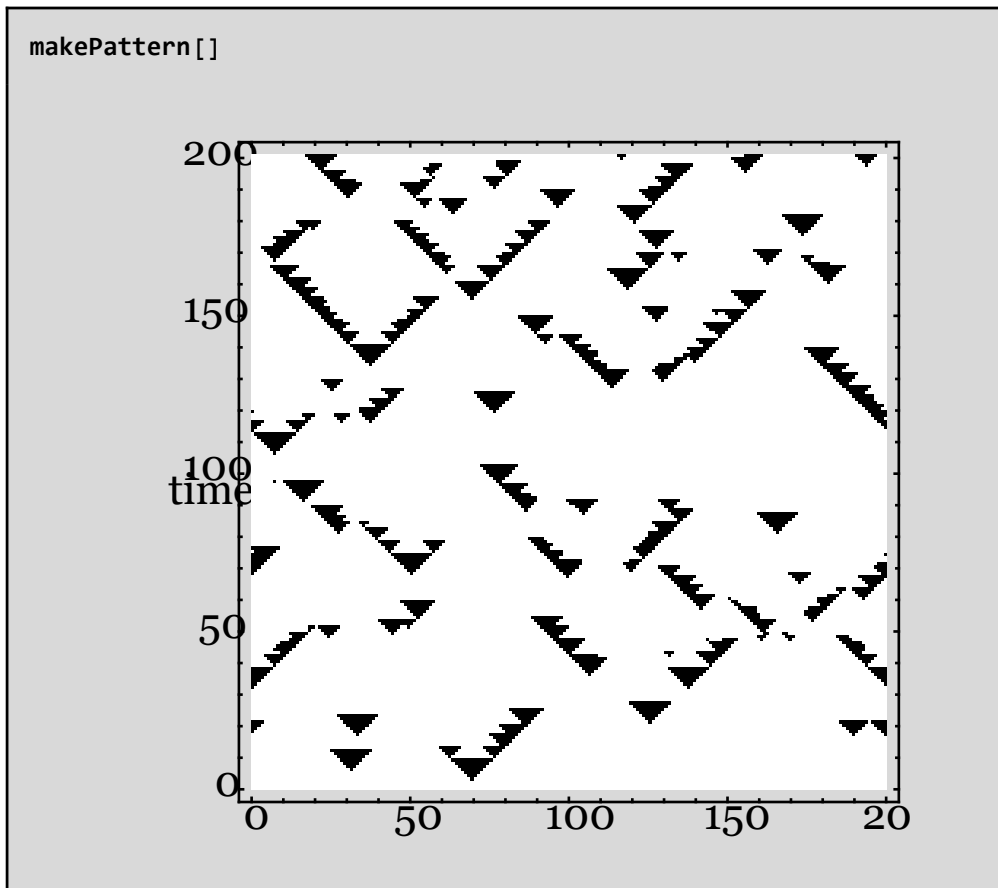
■ Pattern C

```
d = 0; e = 1; initProb = 0;  
ru = 2;  
rv = 0;  
w1 = 10;  
w2 = 48;  
m0 = m1 = 0;  
p = 0.002;
```



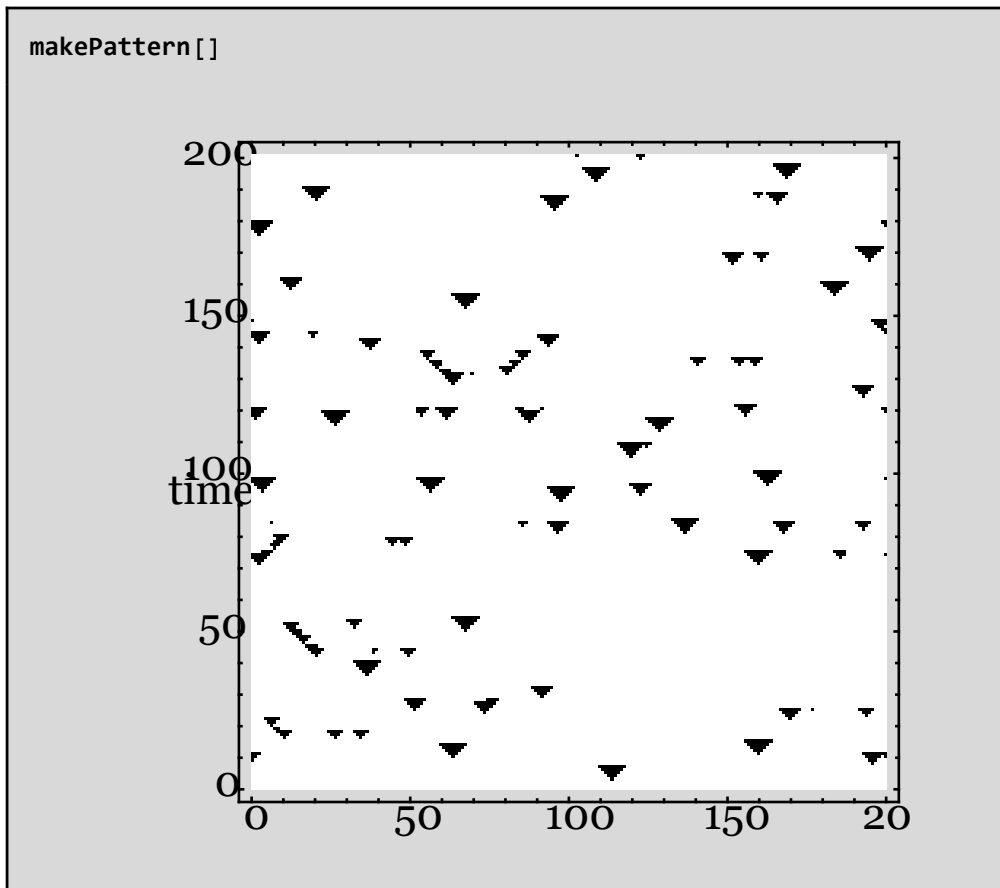
■ Pattern D

```
d = 0; e = 1; initProb = 0;  
ru = 1;  
rv = 16;  
w1 = 8;  
w2 = 6;  
m0 = m1 = 0;  
p = 0.002;
```



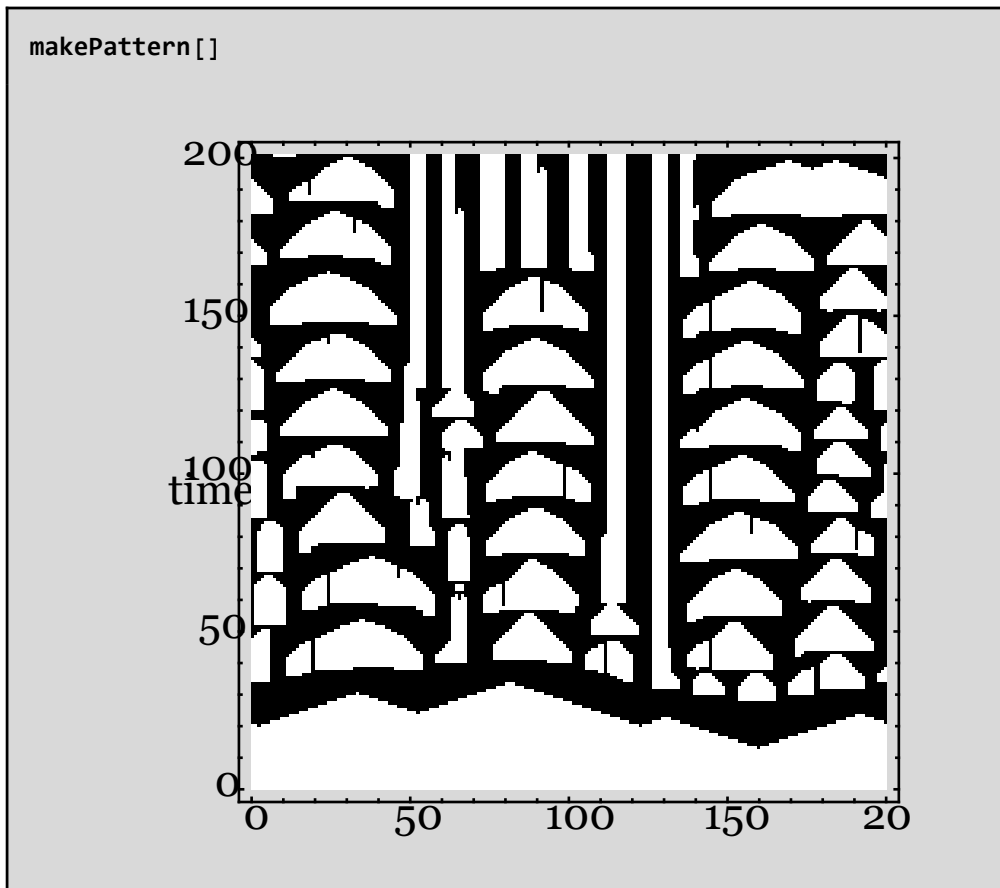
■ Pattern E

```
d = 0; e = 1; initProb = 0;  
ru = 1;  
rv = 17;  
w1 = 16;  
w2 = 6;  
m0 = m1 = 0;  
p = 0.002;
```



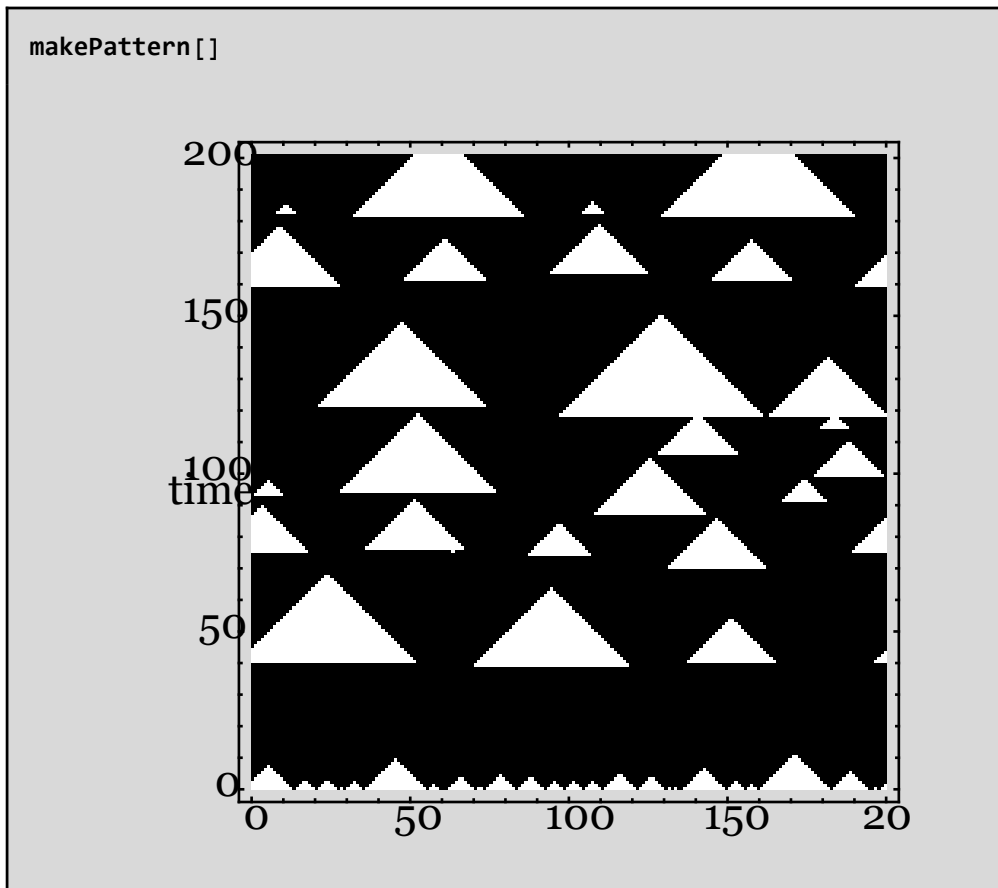
■ Pattern F

```
d = 0; e = 1; initProb = 0;  
ru = 3;  
rv = 8;  
w1 = 2;  
w2 = 11;  
m0 = 0;  
m1 = 0.3;  
p = 0.001;
```



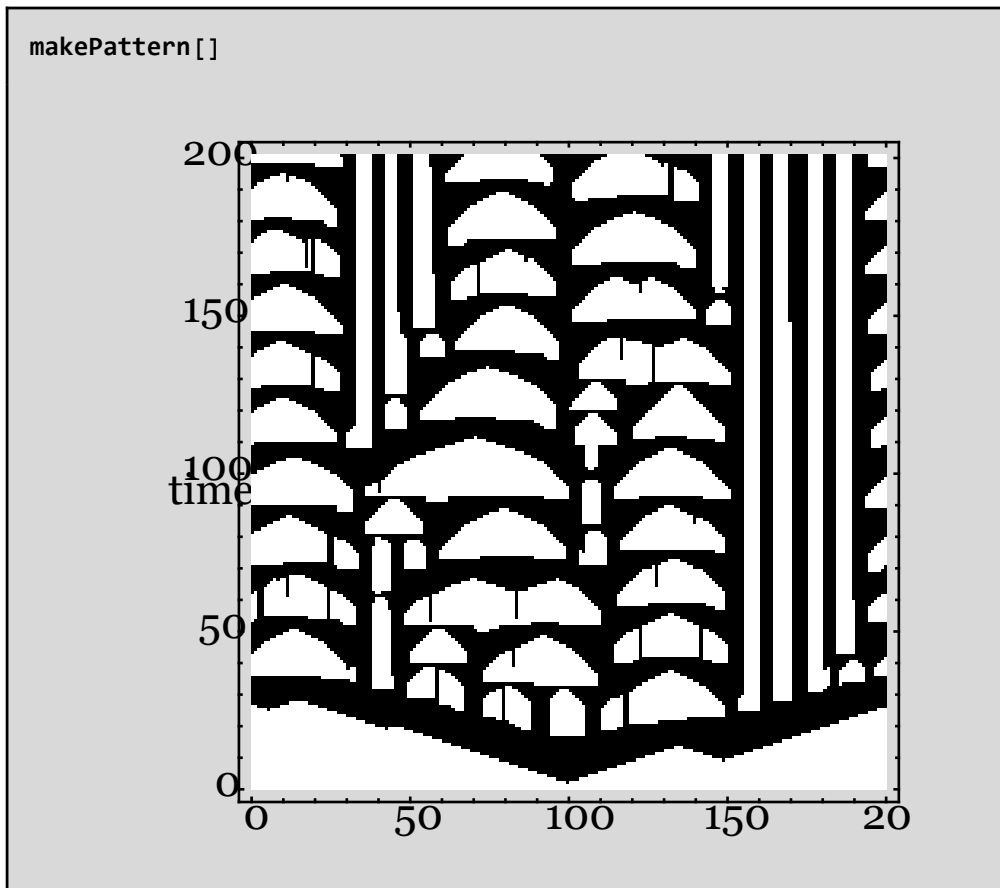
■ Pattern G

```
ru = 1;  
rv = 23;  
w1 = 4;  
w2 = 61;  
m0 = m1 = 0;  
p = 0;  
d = 0.05;  
e = 0;  
initProb = 0.1;
```

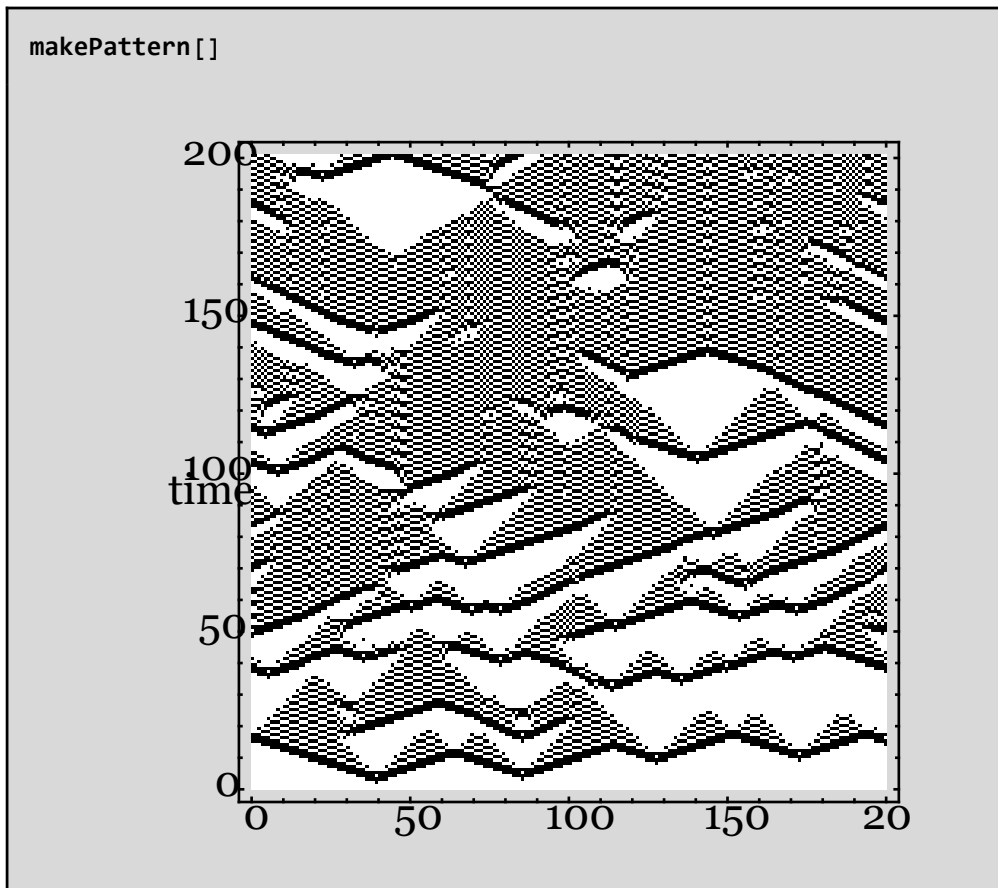
■ Pattern H

```
d = 0; e = 1; initProb = 0;  
ru = 3;  
rv = 8;  
w1 = 2;  
w2 = 11;  
m0 = 0;  
m1 = 0.3;  
p = 0.001;
```



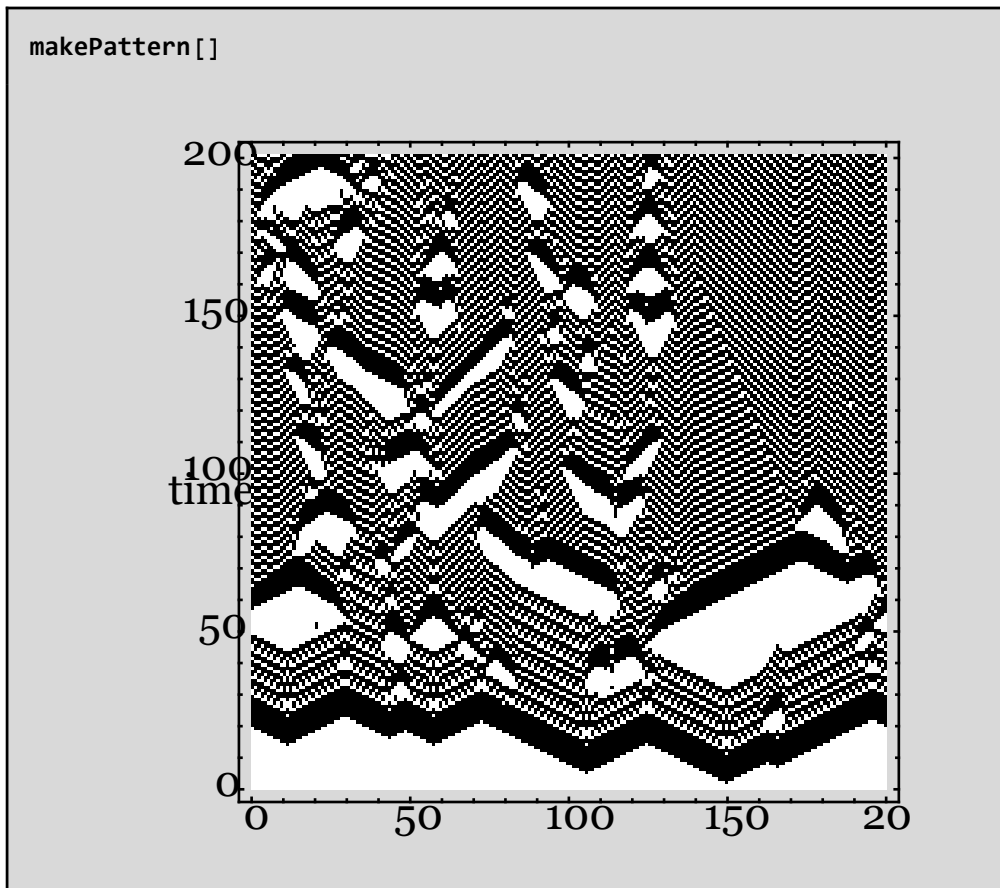
■ Pattern I

```
initProb = 0;  
ru = 3;  
rv = 0;  
w1 = 5;  
w2 = 12;  
m0 = 0;  
m1 = 0.22;  
p = 0.004;  
d = 0.19;  
e = 0.0;
```



■ Pattern J

```
initProb = 0;  
ru = 2;  
rv = 0;  
w1 = 6;  
w2 = 35;  
m0 = 0;  
m1 = 0.05;  
p = 0.002;  
d = 0.1;  
e = 0.0;
```



■ Pattern K

```
d = 0; e = 1; initProb = 0;  
ru = 1;  
rv = 2;  
w1 = 5;  
w2 = 10;  
m0 = 0;  
m1 = 0.3;  
p = 0.002;
```

