

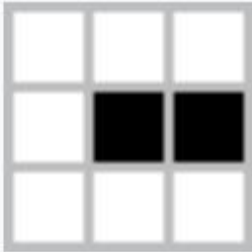



3Dセルオートマトンの実装

この実験で何をやったか

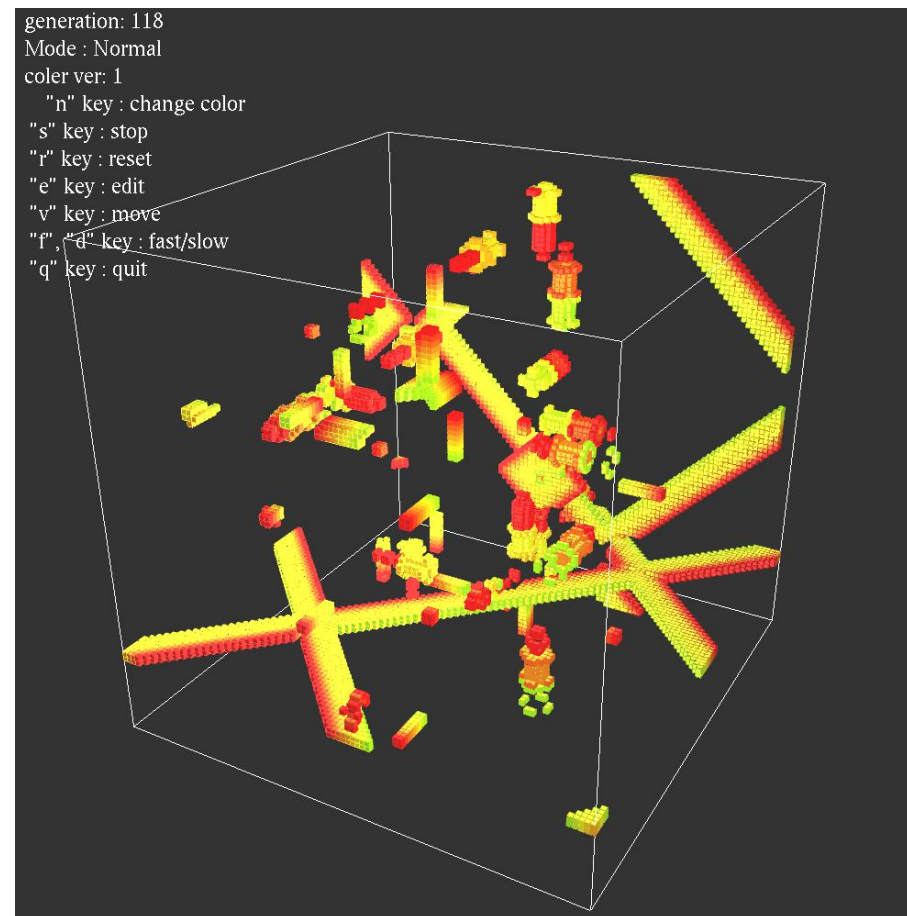
- オートマトンの一種であるConwayのライフゲームの拡張

ライフゲームの基本ルール

誕生	生存（維持）	死（過疎）	死（過密）
			

Conwayのライフゲームと何が違うのか

- 3次元空間への拡張
- 複数状態を保持
- 寿命の概念の導入



なぜこれをやろうと思ったか

- 当初何をしようとしていたか
 - 3DにしたらopenGLの良さが出せそう。
- 実装してみたけどなんか見栄えがよくない...
 - セルの着色
 - 何に対して? → 状態
 - 「ライフゲームっぽい」セルの発生を視覚的に感じたい。
- じゃあ状態数を拡張してみよう
 - ルールの変更
 - 編集(デバッグ)モードを追加

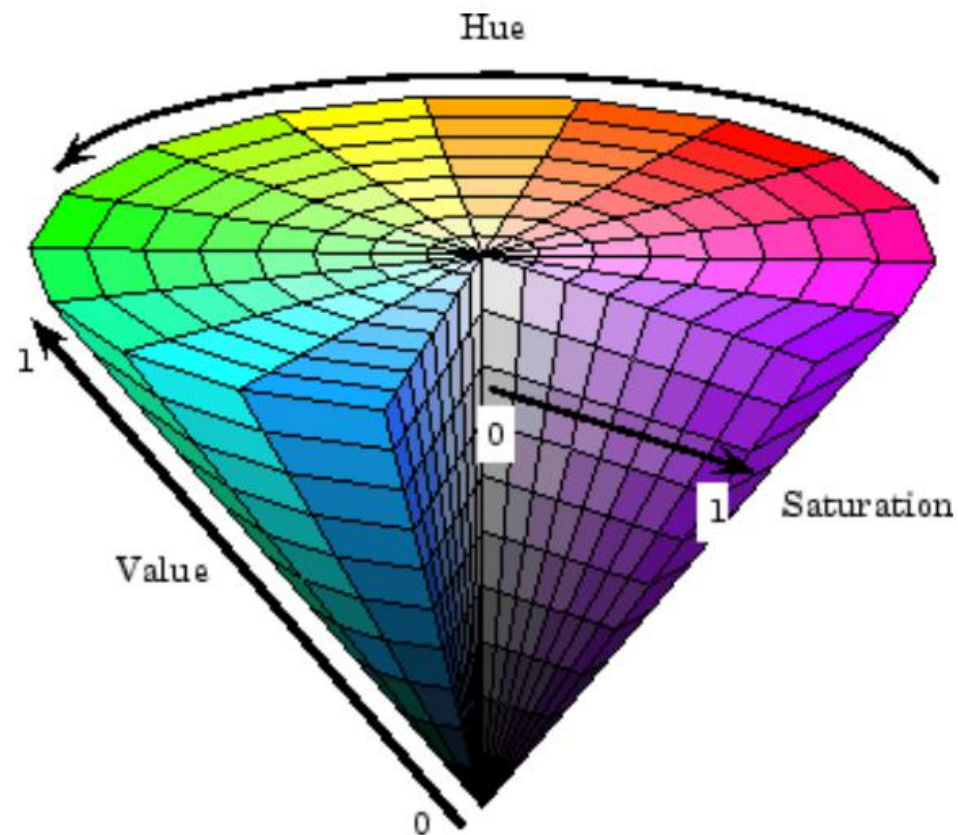
技術的な工夫・苦労した点

- グラデーションの作成
- 対話性の向上
- 描画の高速化
- (ライフゲームのルール策定)

グラデーションの作成

- HSV色空間
- Hueを 15° ずつ変化
- HSV → RGB

HSV 色空間の図



<https://jp.mathworks.com/help/images/convert-from-hsv-to-rgb-color-space.html>

対話性の向上

- 使いそうな機能は一通り追加
 - 停止・再生
 - セルの追加・消去
 - セルの更新を加速・減速
 - セルのリセット
 - オートローテーションモード
 - etc...

描画の高速化

- 必要ないセルを描画しない
- ノイマン近傍のセルが全て生きている場合

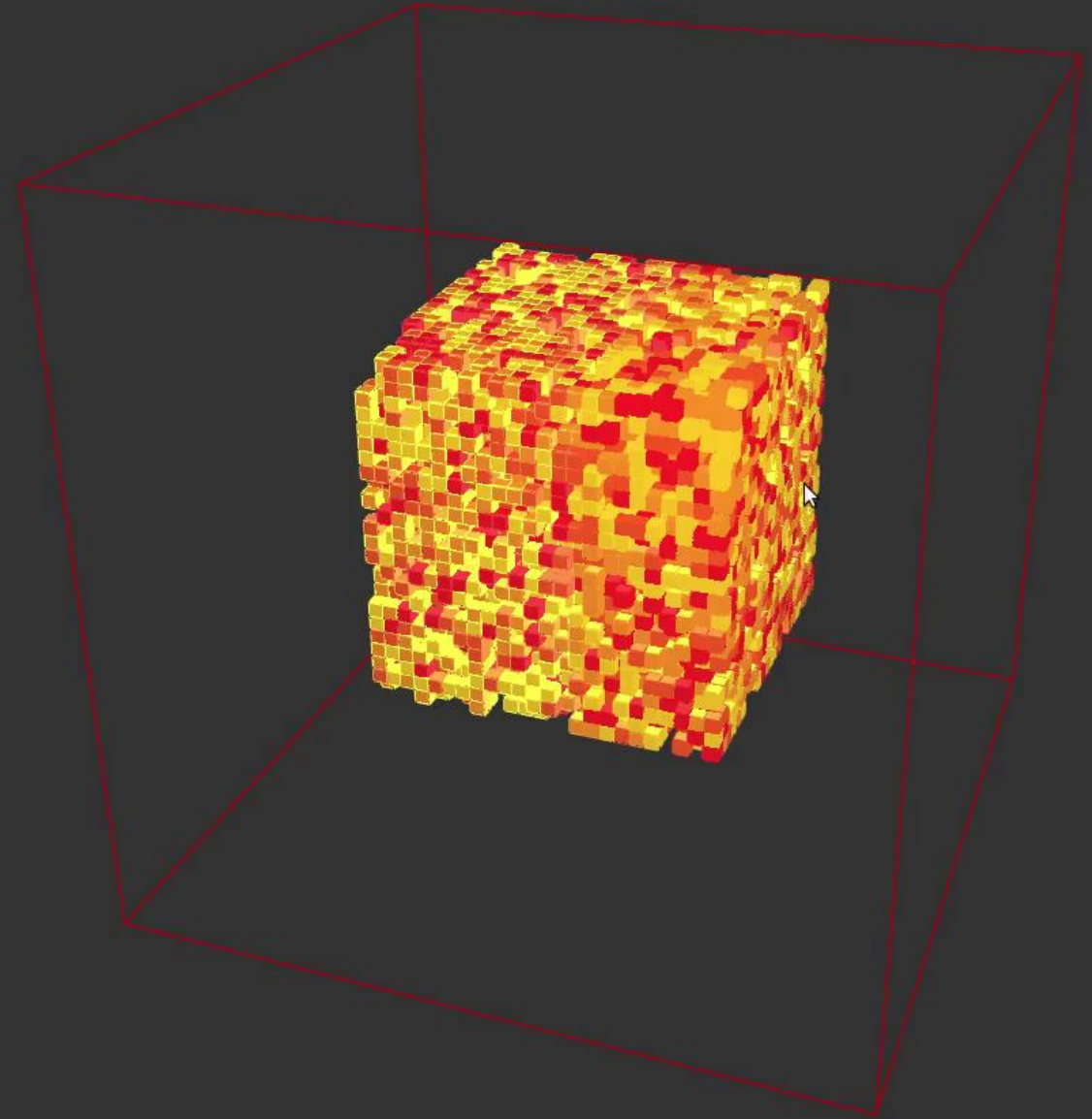
(ライフゲームのルール策定)

- OpenGLで苦労した点ではないが...
- 正直一番大変だった。
 - 大半をここに費やした。



Survive : 4~26
Birth : 4
Number of States : 5

generation: 0
Mode : Stop
"s" key : start
"r" key : reset



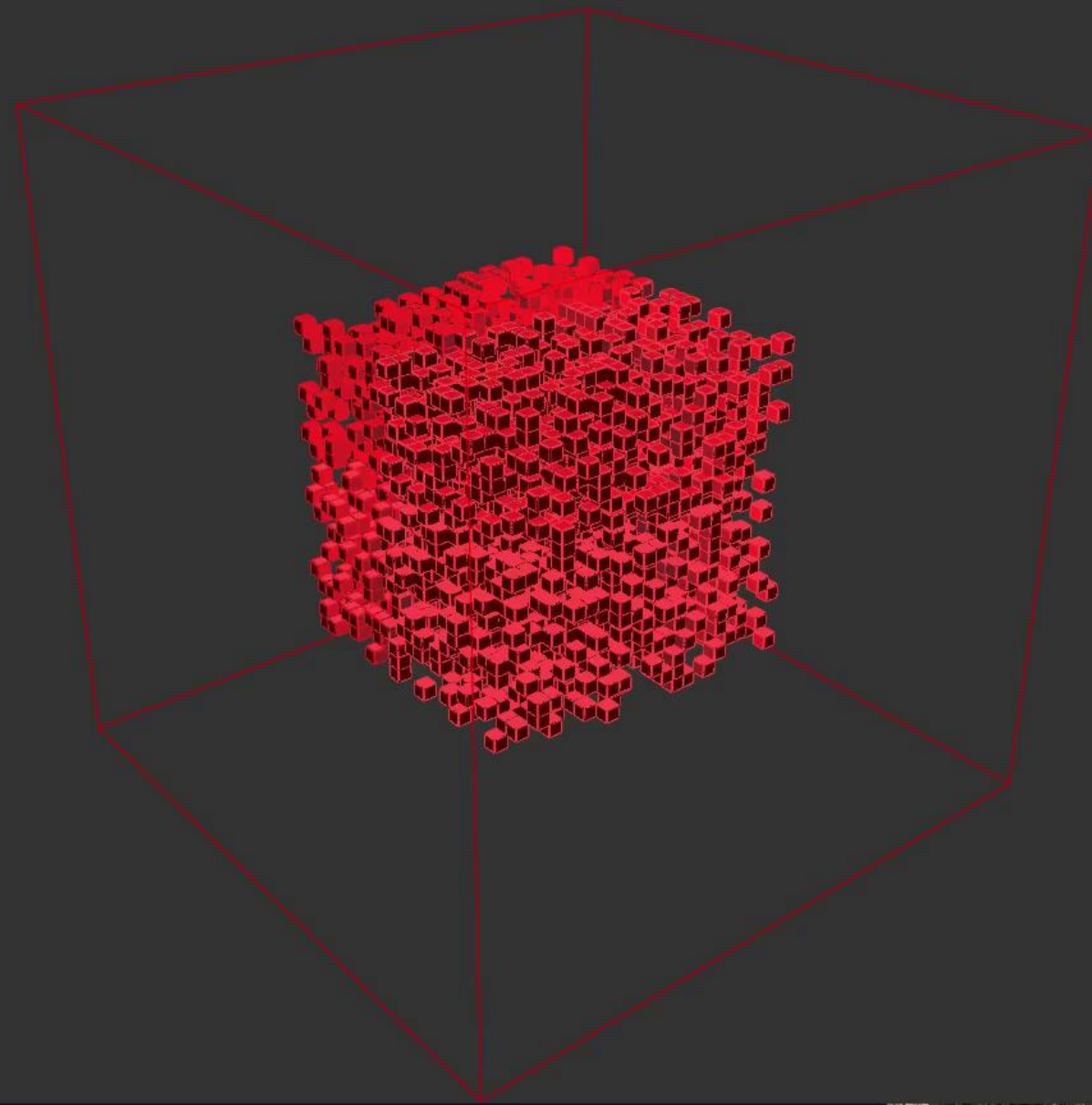
Survive : 4~26

Birth : 4

Number of States : 10

- ・ 初期値によってすべて決まる

test2
generation: 0
Mode : Stop
"s" key : start
"r" key : reset

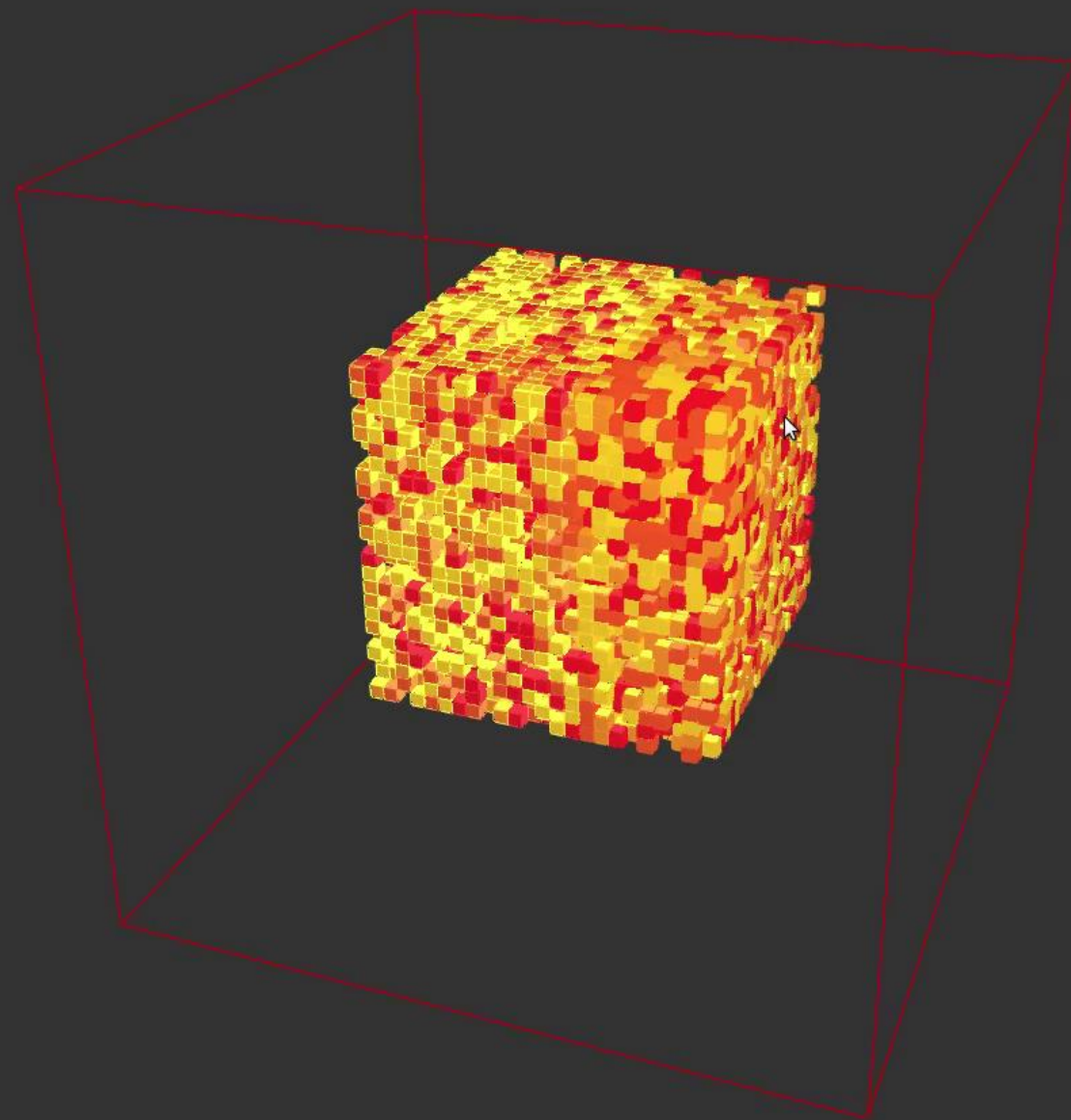


Survive : 4~26

Birth : 4, 8

Number of States : 5

test2
generation: 0
Mode : Stop
"s" key : start
"r" key : reset



Survive : 4~26

Birth : 4, 6, 7, 12, 13

Number of States : 10

generation: 31

Mode : Move

"s" key : stop

"r" key : reset

"e" key : edit

"v" key : Not move

"f", "d" key : fast/slow

"q" key : quit

