Quantum Game of Life

Key Idea:
Conway's game of life is a "cellular automaton", which uses simple rules to simulate complex behaviours.
It is based on a grid in which each cell can either be dead or alive.
The game then evolves in cycles using the following rules:

- Any live cell with two or three live neighbours survives.
- Any dead cell with three live neighbours becomes a live cell.
- All other live cells die in the next generation. Similarly, all other dead cells stay dead.

The quantum version:
In the quantum version, cells can be dead and alive at the same time. This is represented using grayscale.
Due to this, there may be multiple outcomes possible after an evolution step, leading to another state superposition, again represented in grayscale.

The quantum side of the game

Each cell has a state $|\Psi> = \begin{pmatrix} a \\ b \end{pmatrix}$ with $a^2 + b^2 = 1$

If $|\Psi> = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, the cell is alive.

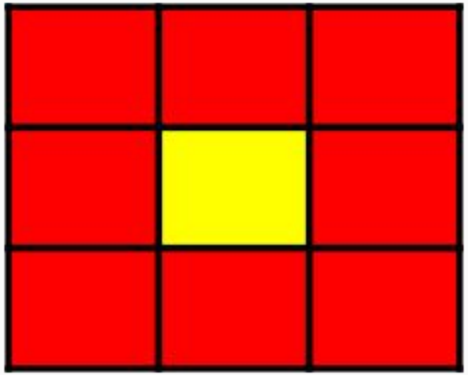
If $|\Psi> = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, the cell is dead.

For any other value, the cell is in a superposition of being alive and dead.

Explanation based on this paper:
https://arxiv.org/pdf/1902.07835.pdf

The quantum side of the game

Each cell (*yellow*) is surrounded by 8 cells (*red*), which are its “neighborhood”:



A “neighborhood liveness” parameter *A* is defined by adding all the “a” values of the surrounding 8 cells:

$$A = \sum_{ik=1}^{8} a_k \, ,$$

Explanation based on this paper: https://arxiv.org/pdf/1902.07835.pdf

The quantum side of the game

The game goes through cycles. After each cycle, the following operations are applied to each cell, based on its "neighborhood liveness" parameter A:

A	$\hat{G}$
$0 \leq A \leq 1$	$\hat{D}$
$1 < A \leq 2$	$(\sqrt{2}+1)(2-A)\hat{D} + (A-1)\hat{S}$
$2 < A \leq 3$	$(\sqrt{2}+1)(3-A)\hat{S} + (A-2)\hat{B}$
$3 < A \leq 4$	$(\sqrt{2}+1)(4-A)\hat{B} + (A-3)\hat{D}$
$A \geq 4$	$\hat{D}$

using these "birth", "death" and "survival" matrices:

$$\hat{B} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \quad \hat{D} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} \quad \hat{S} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Explanation based on this paper:
https://arxiv.org/pdf/1902.07835.pdf

Gameplay

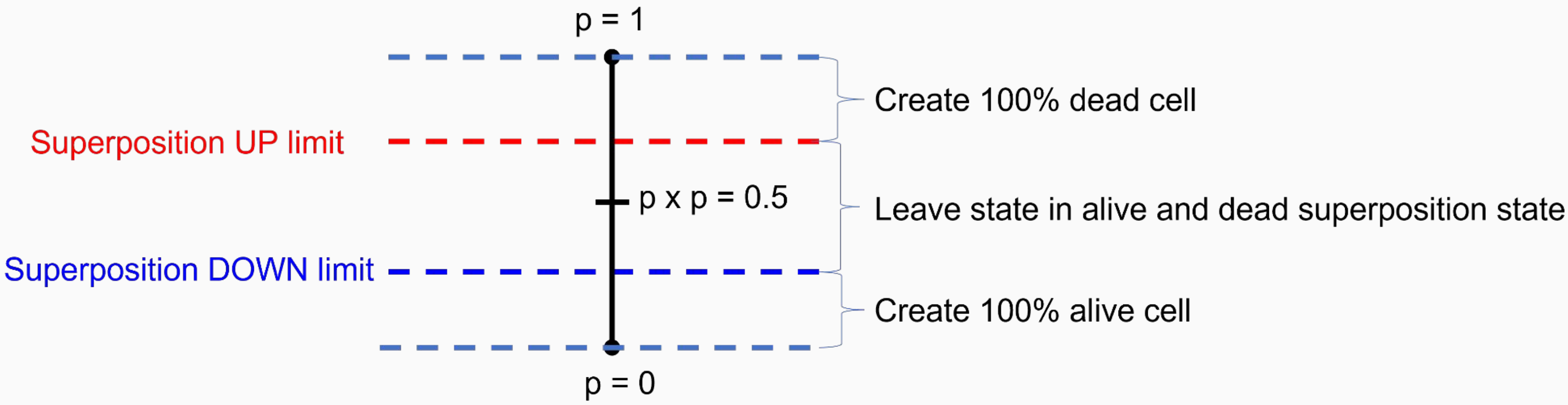
When you click on any cell in the classical space, its state will be toggled, i.e. if there is no cell (dead cell), a cell is created and if there is a cell (alive cell), it gets killed.

At the same time, you create a new cell in the quantum space. But its alive or dead state will depend on the superposition limits.

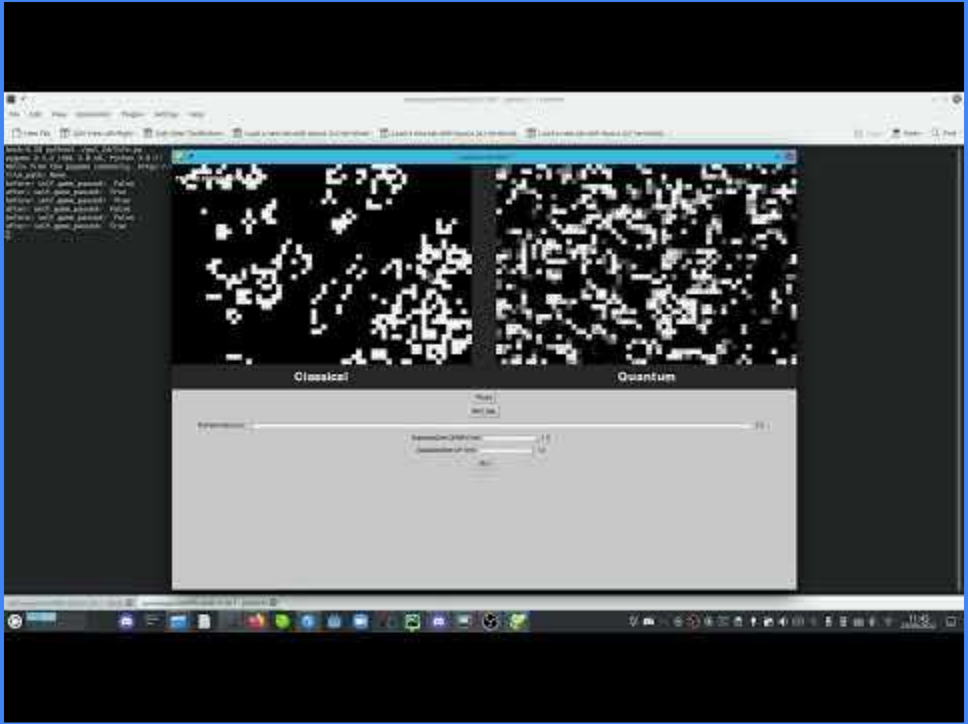
Gameplay

When a new quantum cell is created:

1) Pick a random value **p** between 0 and 1.
2) If **p x p** > (UP limit), create a dead cell.
3) If **p x p** < (DOWN limit), create a live cell.
4) Else, create a cell that has a probability **p x p** of being dead and that is otherwise alive.



Gameplay Video



Credits

Team: Mike Taverne, Aayush Kucheria

Libraries: Pygame, ThorPy

IDEs: PyCharm

Code/game hosting: Github, Replit

External:

(1) Qiskit Camp - Hackaton Madrid 2019 - Quantum Game of Life,
(2) Faux, David, Mayank Shah, and Christopher Knapp. "The semi-quantum game of life." arXiv preprint arXiv:1902.07835 (2019).