# 112 Term Project 2024 Fall

## Pattern Generator Tool based on Hand Gesture

@Author: Chia Hui Yen, huiyenc@andrew.cmu.edu

1. Project description

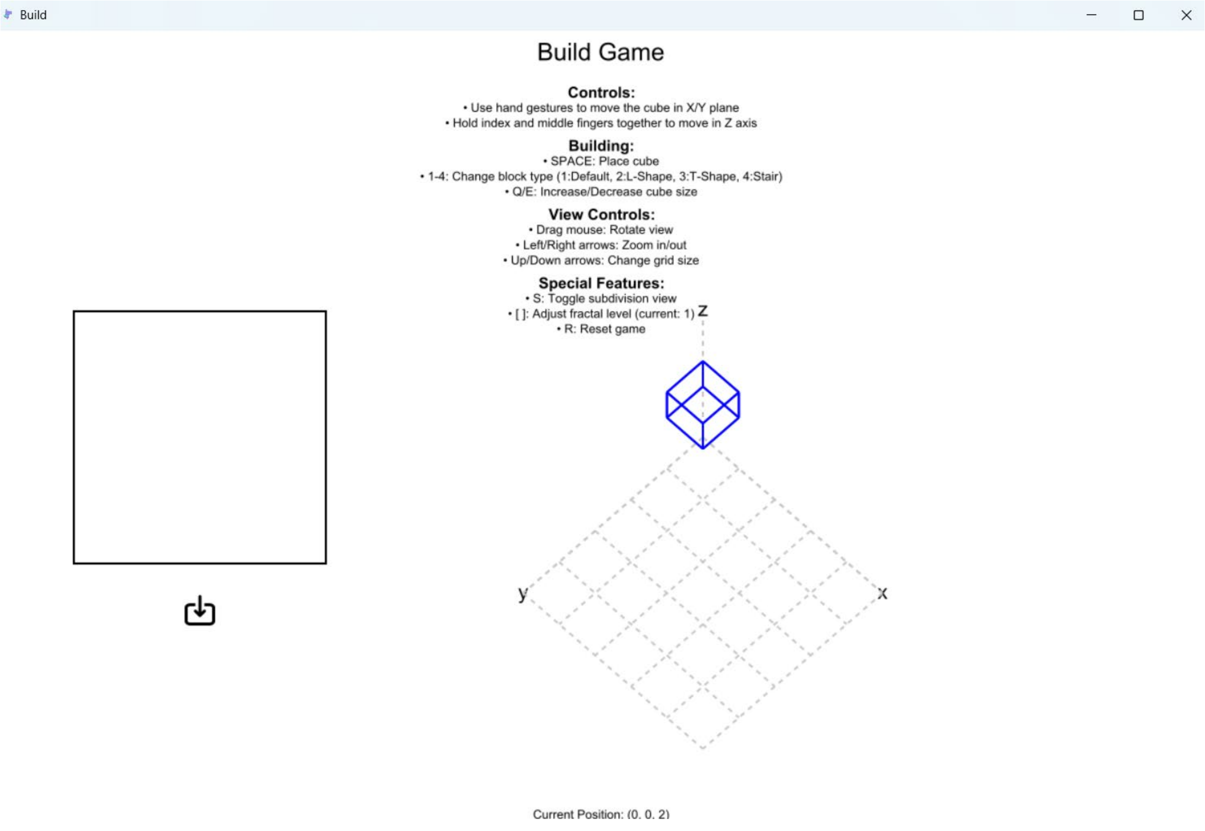
This project is about exploring interactive pattern generation tool that using hand gesture controls, creating pattern in 2d and 3d.

At its core, it offers two main modes: a Build mode for creating and manipulating 3D cube patterns in space, and a Draw mode for free-form 2D pattern creation that can be transformed into 3D structures. Users can seamlessly switch between mouse/keyboard controls and hand gestures, resize cells from 1x1x1 up to 3x3x3, and apply subdivisions to create intricate geometric patterns.

This project applied computer vision through OpenCV and MediaPipe to enable intuitive hand gesture controls - users can manipulate objects in 3D space using simple finger movements, with one finger controlling X-Y plane movement and two fingers controlling Z-axis depth. Besides, the pattern generation method included several cell types through inheritance of class. Another highlight of the mode is it also incorporates catmulk-clark subdivision algorithms that can create smoother surface from edgy initial cube, while maintaining interactive performance. In background of running the appearance, the cube could “merge” together to generate less shape.

The application also supports importing reference images and includes comprehensive undo/redo capabilities to ensure a smooth creative workflow.

For the 2D draw mode, it features a grid-based interface. It’s grid-based drawing app, drawing by mouse drag and hand gesture. With a customized algorithm getNeighbourAvg(), it’s able to times the cell-size by power of 2 and created more detailed pattern by the initiare pattern.



Build Game

Controls:

- Use hand gestures to move the cube in X/Y plane
- Hold index and middle fingers together to move in Z axis

Building:

- SPACE: Place cube
- 1-4: Change block type (1:Default, 2:L-Shape, 3:T-Shape, 4:Stair)
- Q/E: Increase/Decrease cube size

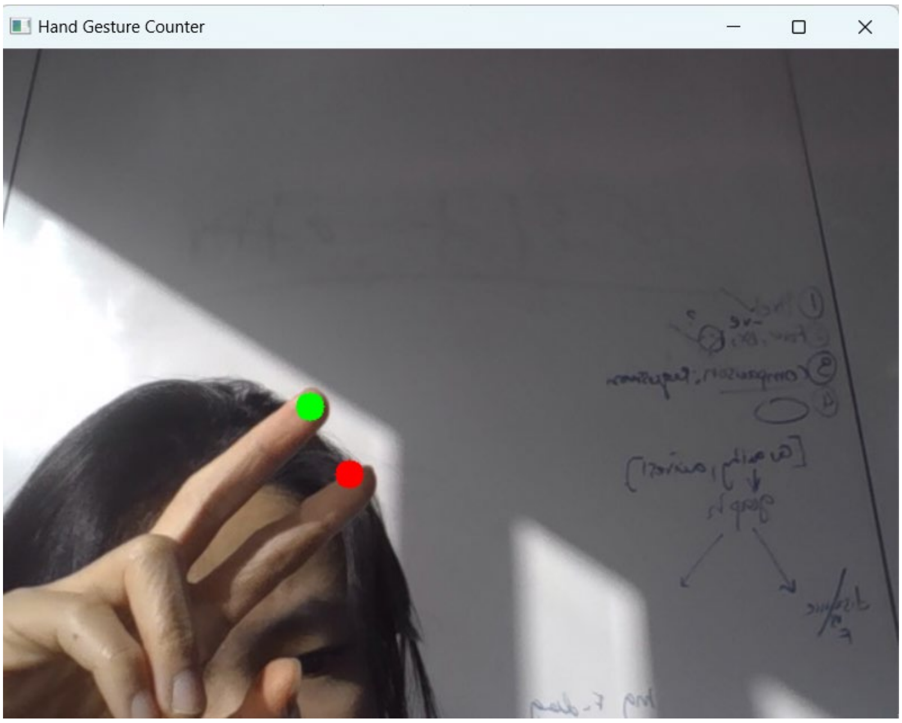
View Controls:

- Drag mouse: Rotate view
- Left/Right arrows: Zoom in/out
- Up/Down arrows: Change grid size

Special Features:

- S: Toggle subdivision view
- [ ]: Adjust fractal level (current: 1)
- R: Reset game

Current Position: (0, 0, 2)



Ideal project for reference:

### Features:

#### Menu Page (MenuPage.py)

- Interactive start menu with animated title

- Multiple mode selection options:
  - Build Mode: Create 3D patterns
  - Draw Mode: Free-form drawing with hand gestures
- Help section with instructions
- Settings configuration

#### Build Mode (Build.py)

- 3D cube pattern generation and manipulation
- Interactive rotation controls via mouse drag
- Pattern subdivision and fractal generation
- Real-time 3D visualization with perspective adjustments
- Edge and vertex manipulation
- Hand gesture controls:
  - One finger tip to move in X-Y plane
  - Two finger tips to move in Z plane
- Cell resizing capabilities (1x1x1 up to 3x3x3)
- Grid scaling
- Space key to fix cell position
- R key to reset view

#### Draw Mode (Draw.py)

- Grid-based drawing interface
- Hand gesture detection for drawing
- Pattern subdivision capabilities with adjustable levels
- Two drawing methods:
  - Mouse-based drawing
  - Hand gesture controls
- Real-time preview
- Pattern reset functionality
- 3D view toggle with space key
- Multiple cell types with inheritance
- Cell placement and removal
- Image import for reference
- Subdivision algorithm for both single and complex cells

- Snap-to-grid functionality

2. Similar Projects

a. Mesh View in 3d, learning 3d to 2d projection

https://github.com/tcabezon/15112-hnx.py.git

b. 112 assignment: week7-tetris.py
Using app.board, a 2d list to store the placement of insert cell, using

===

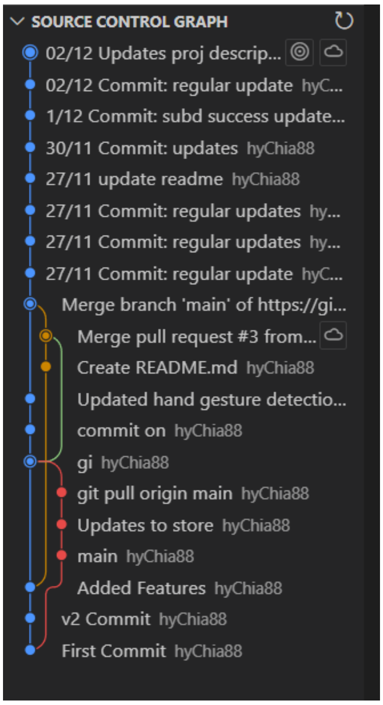
Version Control / Backup Plan

The project uses with GitHub for backup and collaboration:

Remote: https://github.com/huiyenc/112-term-project-2024-fall

- Remote repository on GitHub serves as backup, regular commits tracking feature additions and bug fixes.

- Commit history provides rollback capability if needed



```
02/12 Updates proj descrip...
02/12 Commit: regular update hyC...
1/12 Commit: subd success update...
30/11 Commit: updates hyChia88
27/11 update readme hyChia88
27/11 Commit: regular updates hy...
27/11 Commit: regular updates hy...
27/11 Commit: regular update hyC...
Merge branch 'main' of https://gi...
Merge pull request #3 from...
Create README.md hyChia88
Updated hand gesture detectio...
commit on hyChia88
gi hyChia88
git pull origin main hyChia88
Updates to store hyChia88
main hyChia88
Added Features hyChia88
v2 Commit hyChia88
First Commit hyChia88
```

Local:

- Local backups saved in my own laptop, regularly pushed to remote

===

1. Tech List

### Libraries and Technologies Used

- Python

- CMU Graphics Library - For 2D graphics rendering and user interface

- OpenCV (cv2) - For webcam capture and image processing

- MediaPipe - For hand gesture detection and tracking

- Tkinter - For file dialog

### Installation

Run the application: run MenuPage.py , it will automatically run Build.py or Draw.py depending on the mode you choose. (not done this part yet)

==
=====

2. Key Algorithms

### 3D projection

- Rotation matrices for 3D transformations

- Perspective projection calculations

- Vertex and edge manipulation

### grid base and subdivide Generation

- Grid-based pattern system

- Cell inheritance hierarchy

- Catmulk clark subd

### Computer Vision

- Real-time hand tracking

- Gesture recognition

- Coordinate mapping

### User Interface

- Interactive grid system

- Mouse and keyboard controls

- Hand gesture controls

- Mode switching

### Data Structures

- 2D and 3D arrays for grid representation

- Object-oriented cell system

- Inheritance-based cell types

- Matrix transformations

==

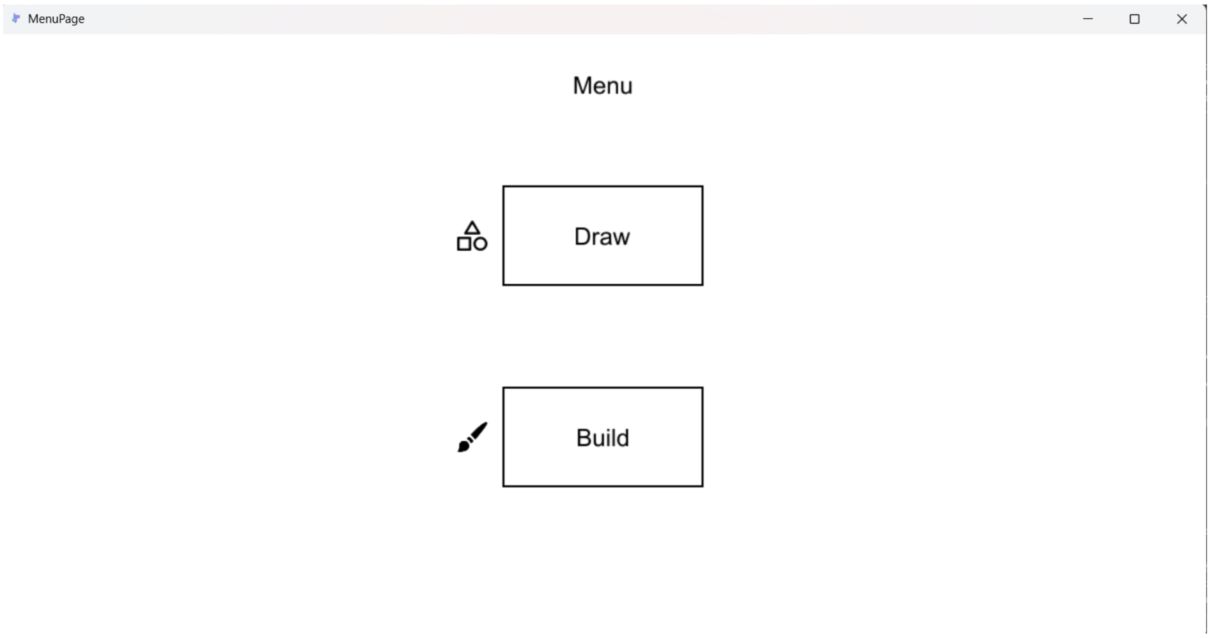
3. Storyboard

### Opening Menu

1. Two main options displayed:

   - Build Mode

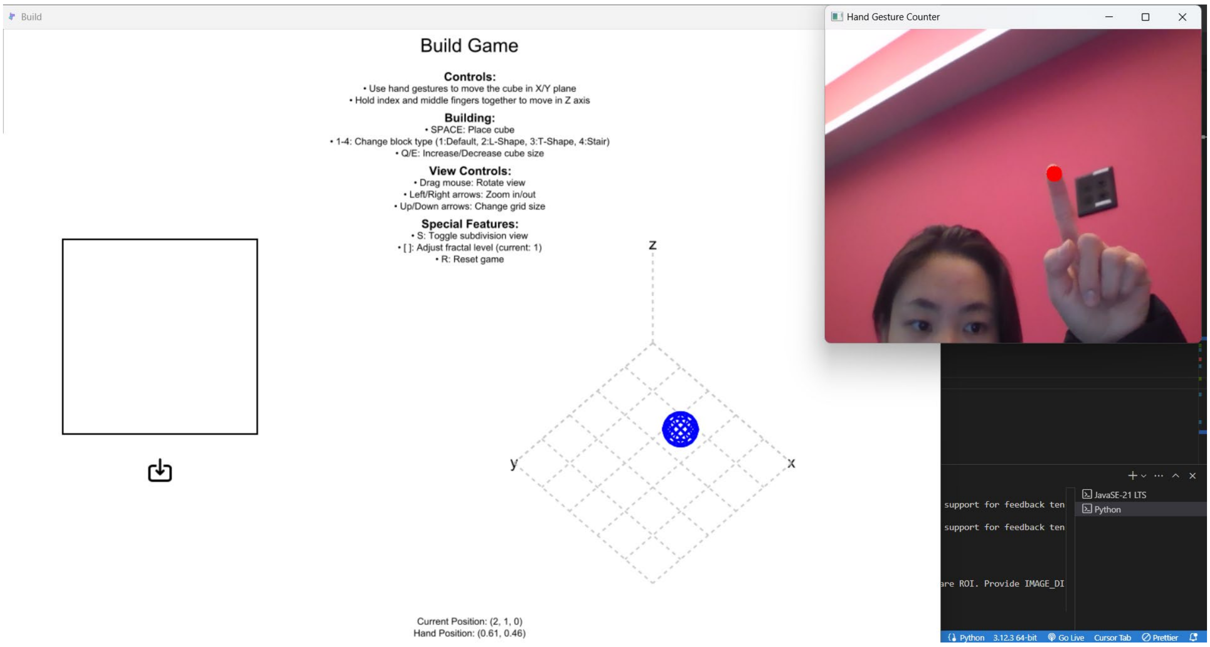
   - Draw Mode



2. Help and settings buttons visible [*future*]

### Build Mode Flow

1. User enters 3D building environment



Build Game

Controls:
• Use hand gestures to move the cube in X/Y plane
• Hold index and middle fingers together to move in Z axis

Building:
• SPACE: Place cube
• 1-4: Change block type (1:Default, 2:L-Shape, 3:T-Shape, 4:Stair)
• Q/E: Increase/Decrease cube size

View Controls:
• Drag mouse: Rotate view
• Left/Right arrows: Zoom in/out
• Up/Down arrows: Change grid size

Special Features:
• S: Toggle subdivision view
• [ ]: Adjust fractal level (current: 1)
• R: Reset game

Current Position: (2, 1, 0)
Hand Position: (0.61, 0.46)

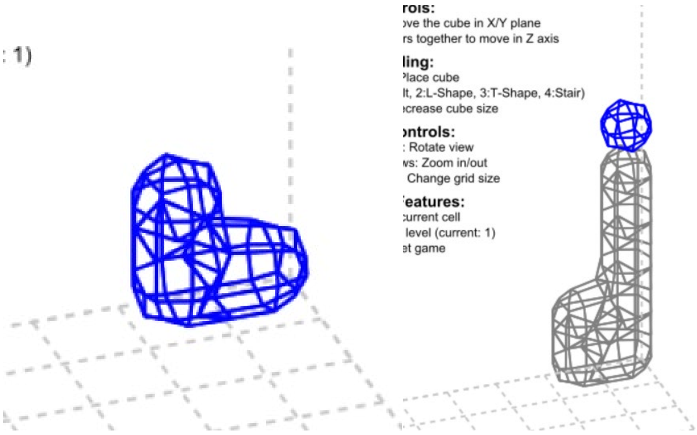
```
support for feedback ten

support for feedback ten


are ROI. Provide IMAGE_DI
```

2. 5x5 grid appears with coordinate system, with a init cell at (0,0,0)

3. User can build 3d pattern in the grid by place cell. There are several type of cell can choose to use and also at the side a rectangle to import image for reference. Cell beside each other will auto snap together. (it is worked by clean duplicated edge/pts function)



rols:
ove the cube in X/Y plane
rs together to move in Z axis

ling:
Place cube
lt, 2:L-Shape, 3:T-Shape, 4:Stair)
crease cube size

ontrols:
: Rotate view
ws: Zoom in/out
Change grid size

eatures:
current cell
level (current: 1)
et game

Detail user guide:

# Movement controls
Use hand gestures to move the cube in X/Y plane, Hold index and middle fingers together to move in Z axis.

# Building controls
'SPACE: Place cube
1-4: Change block type
Q/E: Increase/Decrease cube size
D: Delete current cell

# View controls
Drag mouse: Rotate view
Left/Right arrows: Zoom in/out

Up/Down arrows: Change grid size

# Special features
[ ]: Adjust subd level
R: Reset game

Save/load creations [*future*]

### Draw Mode Flow

1. User enters 2D drawing grid

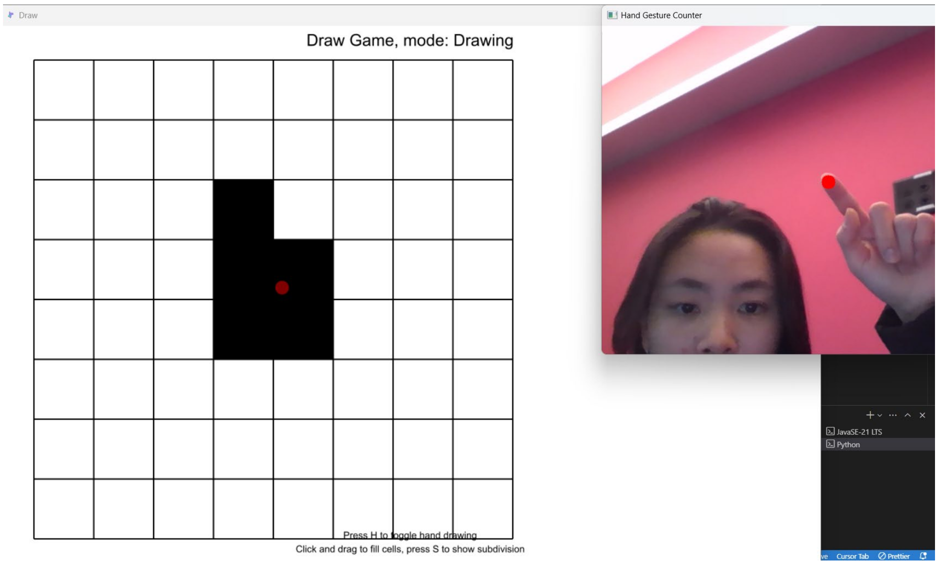
2. Tools appear:

   - Mouse drawing

   - Hand gesture controls

   - Subdivision options

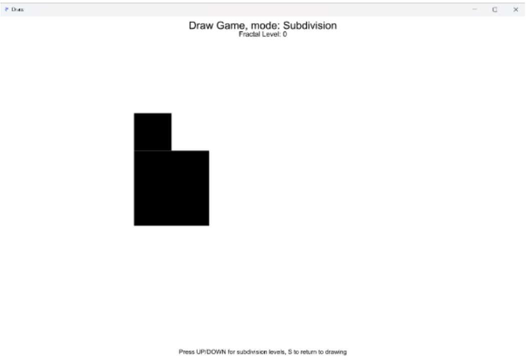
   - Pattern generation



Draw Game, mode: Drawing

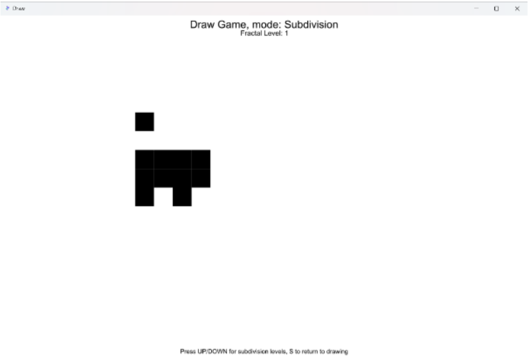
Press H to toggle hand drawing

Click and drag to fill cells, press S to show subdivision



Draw Game, mode: Subdivision
Fractal Level: 0

Press UP/DOWN for subdivision levels, S to return to drawing



Draw Game, mode: Subdivision
Fractal Level: 1

Press UP/DOWN for subdivision levels, S to return to drawing

3. User creates patterns by:

   - Drawing directly

   - Using hand gestures

- Applying subdivisions

4. Option to view in 3D [*future*]

### UX experience

- Save & Export result [*future*]

==

4. Notes

- Webcam access required for hand gesture features

- Recommended screen resolution: 1200x800

- Maximum subdivision level: 3 #(for now, not completed yet)

==

5. References & Citations

### 3D projection

concept: https://skannai.medium.com/projecting-3d-points-into-a-2d-screen-58db65609f24 and modified it to fit my needs

https://github.com/tcabezon/15112-hnx.py.git -> hnXfunction.py (line 74, twoDToIsometric(app,points), take the concept of proj 3d pts & but not using numpy, but not using cuz 3d rotation is different)

Formula from below: Taking Y-axis rotation x X-axis rotation

https://www.quora.com/How-do-you-convert-3D-coordinates-x-y-z-to-2D-coordinates-x-y

https://en.wikipedia.org/wiki/Rotation_matrix#General_3D_rotations (Basic 3d rotation -> General 3d rotation)

At the end, use 3d matrix rotation to rotate the 3d points, then project to 2d screen

### Hand gesture control

https://www.youtube.com/watch?v=v-ebX04SNYM

https://youtu.be/RRBXVu5UE-U?si=FTBWxNPHmmu-KmW6

### Catmull-Clark subdivision

https://en.wikipedia.org/wiki/Catmull%E2%80%93Clark_subdivision_surface

https://rosettacode.org/wiki/Catmull%E2%80%93Clark_subdivision_surface