Build by Motion

Pattern Generator Tool based on Hand Gesture, term project for 15112 CMU

Author: Chia Hui Yen, huiyenc@andrew.cmu.edu, Instructor/mentor: Austin & Lucy

1. Project description

This application explores the potential of using body gestures to build, draw, and create geometric patterns. It is an experiment—a brief demo or prototype—to test the experience of controlling building blocks with hand gestures. Currently, the gestures are limited to moving cells. In my ultimate vision, this project will operate within VR, XR, or MR environments, turning it into an immersive game or creative tool.

In the main build page, user can use hand gesture and shortcuts to control cell to build unique patterns, and use [/] to do subdivision on geometry.

Besides, to simplify the creation of complex patterns, I've implemented features that allow users to use basic actions or methods to visualize intricate designs. For example, users can import images to generate patterns, jump to 2d draw page draw their own designed cell pattern.

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.

Demo link:

https://drive.google.com/file/d/1qRN519mcnK0UUIrHIP6DIMGxLYHkUj4z/view?usp=sharing

- **Structure**

Menu Page (MenuPage.py)

Launch the app and hit **SPACE** (or use a key gesture in future iterations) to begin.

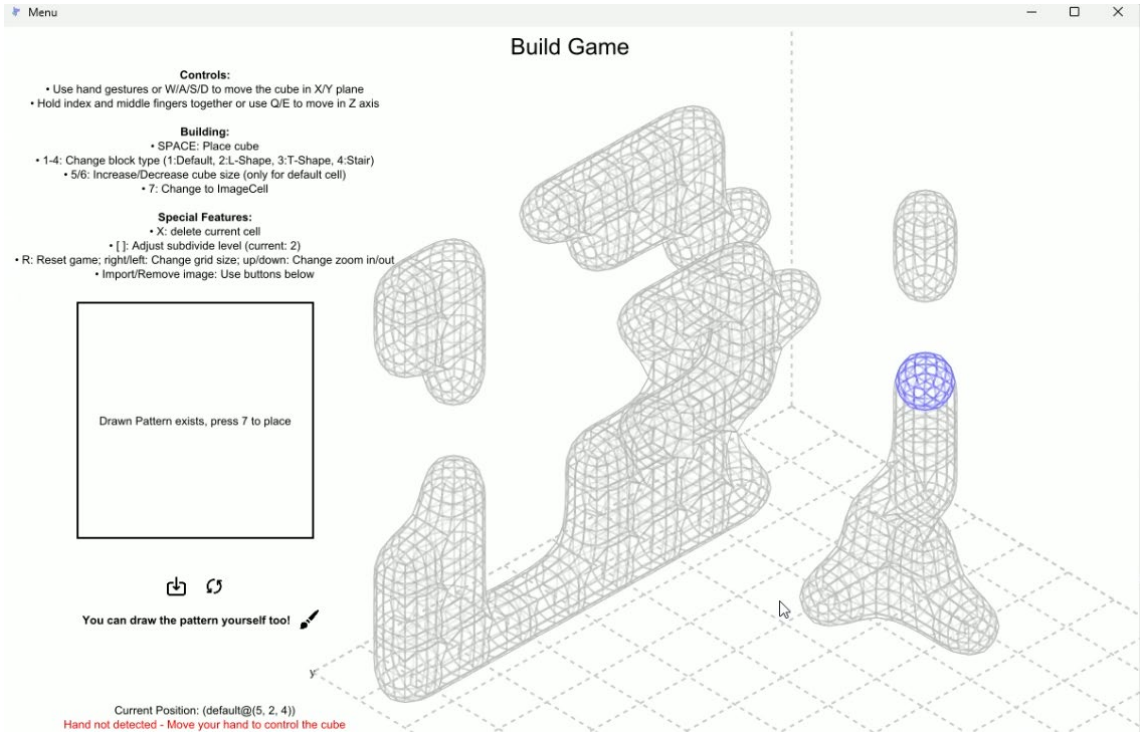
Build Page (Build.py)

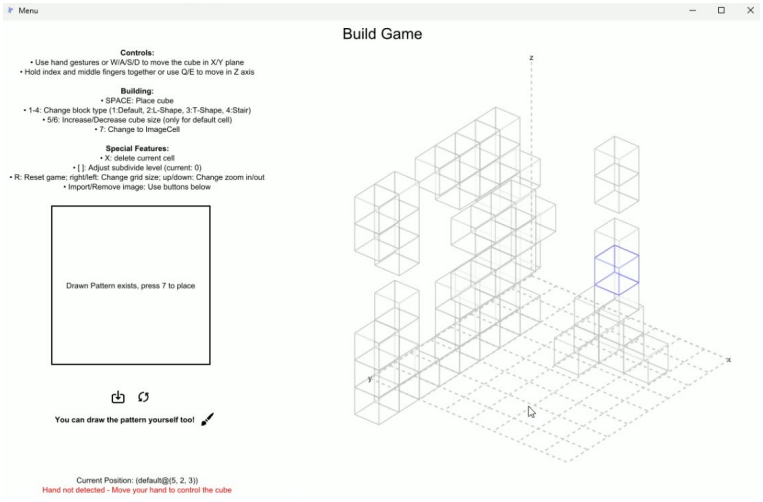
Main 3d building page. Able to jump to Draw page to draw custom cell pattern.

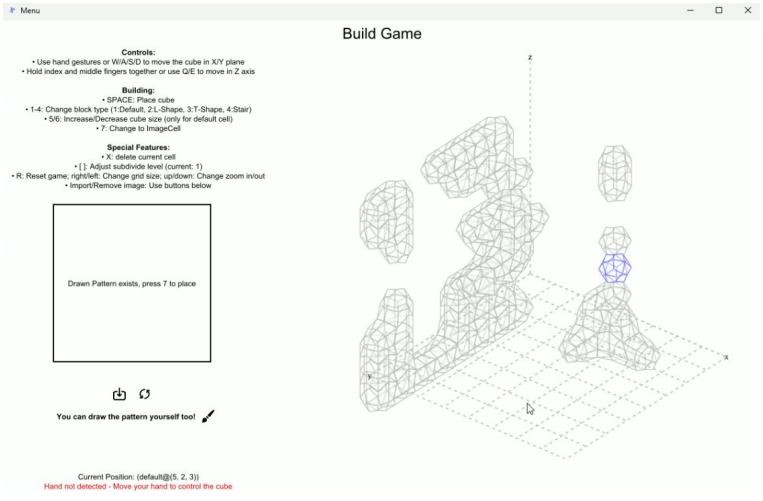
Draw Page (Draw.py)

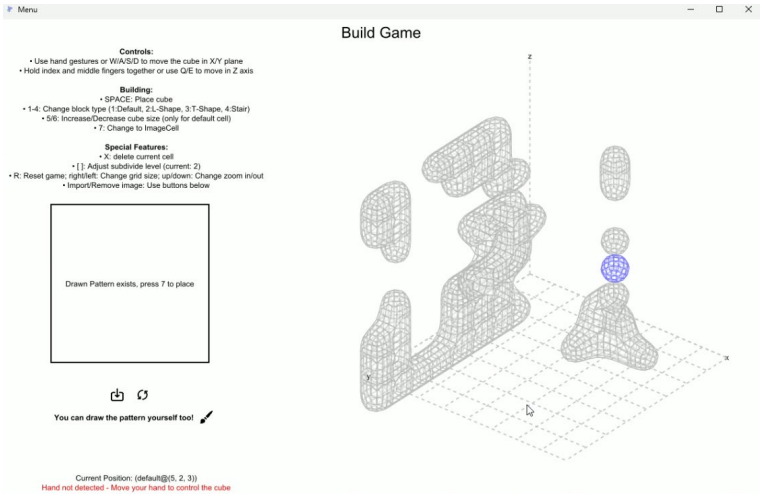
Grid-based drawing interface. Hit S & E to save and export cell pattern.

Some screenshot of app to show the use:


Menu
Build Game
Controls:
• Use hand gestures or W/A/S/D to move the cube in X/Y plane
• Hold index and middle fingers together or use Q/E to move in Z axis
Building:
• SPACE: Place cube
• 1-4: Change block type (1:Default, 2:L-Shape, 3:T-Shape, 4:Stair)
• 5/6: Increase/Decrease cube size (only for default cell)
• 7: Change to ImageCell
Special Features:
• X: delete current cell
• [ ]: Adjust subdivide level (current: 2)
• R: Reset game; right/left: Change grid size; up/down: Change zoom in/out
• Import/Remove image: Use buttons below
Drawn Pattern exists, press 7 to place
You can draw the pattern yourself too!
Current Position: (default@(5, 2, 4))
Hand not detected - Move your hand to control the cube


Menu
Build Game
Controls:
• Use hand gestures or W/A/S/D to move the cube in X/Y plane
• Hold index and middle fingers together or use Q/E to move in Z axis
Building:
• SPACE: Place cube
• 1-4: Change block type (1:Default, 2:L-Shape, 3:T-Shape, 4:Stair)
• 5/6: Increase/Decrease cube size (only for default cell)
• 7: Change to ImageCell
Special Features:
• X: delete current cell
• [ ]: Adjust subdivide level (current: 0)
• R: Reset game; right/left: Change grid size; up/down: Change zoom in/out
• Import/Remove image: Use buttons below
Drawn Pattern exists, press 7 to place
You can draw the pattern yourself too!
Current Position: (default@(5, 2, 3))
Hand not detected - Move your hand to control the cube


Menu
Build Game
Controls:
• Use hand gestures or W/A/S/D to move the cube in X/Y plane
• Hold index and middle fingers together or use Q/E to move in Z axis
Building:
• SPACE: Place cube
• 1-4: Change block type (1:Default, 2:L-Shape, 3:T-Shape, 4:Stair)
• 5/6: Increase/Decrease cube size (only for default cell)
• 7: Change to ImageCell
Special Features:
• X: delete current cell
• [ ]: Adjust subdivide level (current: 1)
• R: Reset game; right/left: Change grid size; up/down: Change zoom in/out
• Import/Remove image: Use buttons below
Drawn Pattern exists, press 7 to place
You can draw the pattern yourself too!
Current Position: (default@(5, 2, 3))
Hand not detected - Move your hand to control the cube



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
Grid based application. Using app.board, a 2d list to store the placement of insert cell.

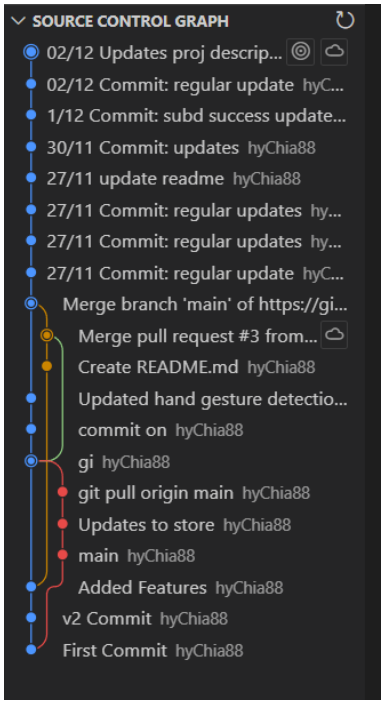
3. Version Control / Backup Plan

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



Local:

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

4. Tech List

Libraries and Technologies Used

- 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

Installation

Run Menu.py , it will jump to Build.py or Draw.py by button from Menu.py you choose.

5. 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 subdivision method

Computer Vision for gesture control
- Real-time hand tracking
- Gesture recognition

User Interface
- Interactive grid system
- Hand gesture controls & Mouse and keyboard controls

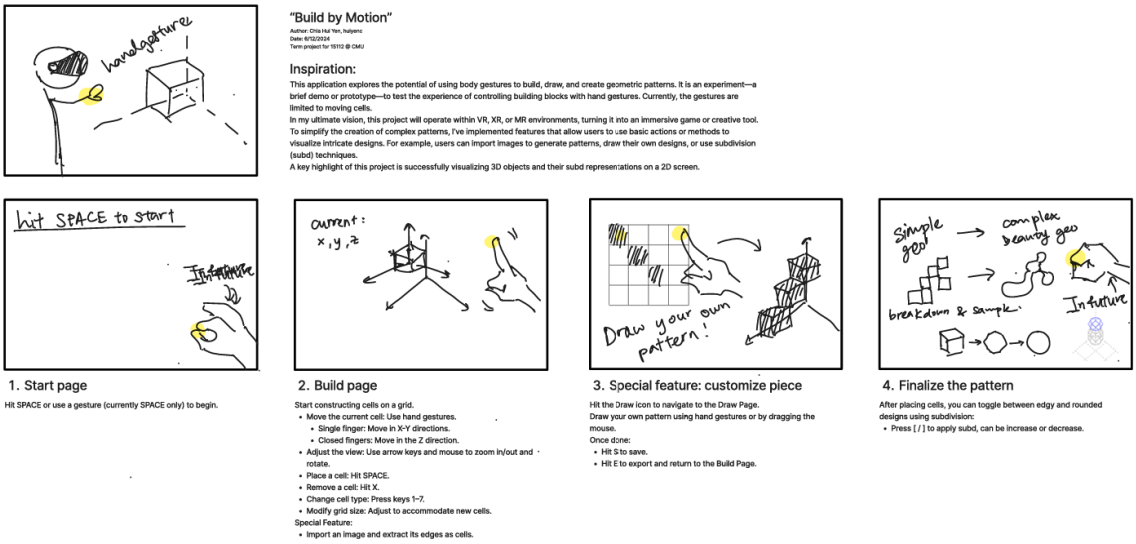
Data Structures
- 2D and 3D arrays for grid representation
- Object-oriented cell system
- Inheritance-based cell types
- Matrix transformations

6. Notes

- Webcam access required for hand gesture features
- Recommended screen resolution: 1200x750
- Maximum subdivision level: 0-2

7. Storyboard

Detail can be seen at external storyboard.pdf file



8. 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)

- OpenCV parse image

convert an RGB image to Boolean array method

https://answers.opencv.org/question/216848/how-to-convert-an-rgb-image-to-boolean-array/

https://stackoverflow.com/questions/70574748/what-settings-in-cv2-threshold-to-threshold-this-image

https://www.tutorialspoint.com/opencv-python-how-to-convert-a-colored-image-to-a-binary-image

- 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

- Use Gen AI for debugging