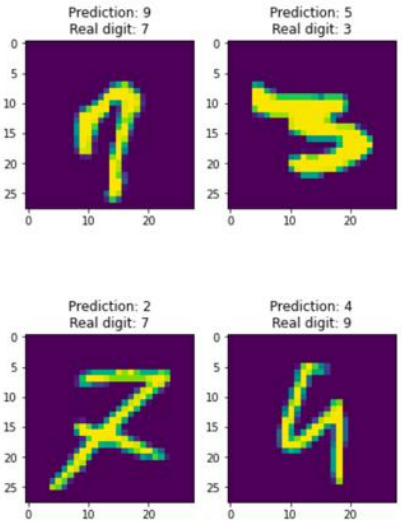
Computer Vision

CNN, Adversarial Attack, Deep Learning on 3D Data, Camera Model and Rigid Transformations

Works I have done in this project:

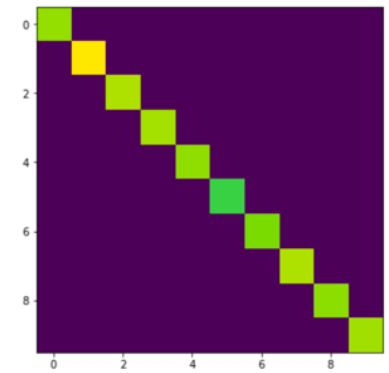
- Implement Convolutional Neural Network on Handwritten Digit Database (CNN)
  - Designed layers in the model to achieve 99% test accuracy
  - Plotted confusion matrix to measure the performance of the network
  - Visualized weights and kernels of the first layer of the CNN
- Adversarial Attack
  - Generated adversarial noise using the fast sign gradient method
  - Added noise to the input images (from handwritten digit database)
  - Plotted output images added with different level of noise by adversarial attack
- Deep Learning on 3D Data (dataset contains objects such as chair, car, lamp, table, etc.)
  - Built pointnet architecture to classify data to classes with at least 90% accuracy
  - Plotted critical points / salient points of data
- Camera Model and Rigid Transformations
  - Viewed objects from camera models with different focal length and optical axis



Plot digits that the network got wrong

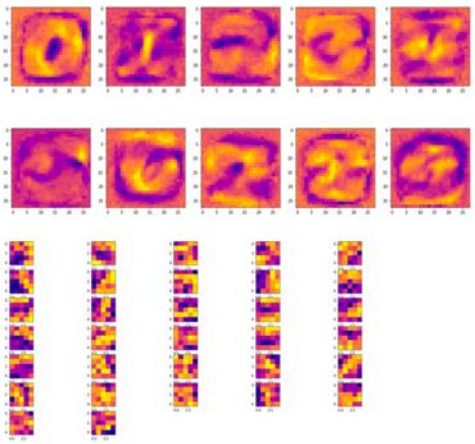
```
class CNN(nn.Module):
    def __init__(self, input_size, num_classes):
        """
        init convolution and activation layers
        Args:
            input_size: (1,28,28)
            num_classes: 10
        """
        super(CNN, self).__init__()

        ### YOUR CODE HERE
        self.layer1 = nn.Conv2d(1, 32, kernel_size=5, padding=1)
        self.layer2 = nn.Conv2d(32, 64, kernel_size=5)
        self.flat = nn.Flatten()
        self.fc1 = nn.Linear(1024, 10)
        ### END OF CODE


    def forward(self, x):
        """
        forward function describes how input tensor is transformed to output
tensor
        Args:
            x: (Nx1x28x28) tensor

        """
        ### YOUR CODE HERE
        # x = torch.flatten(x, 1)
        # print(x.shape)
        x = F.max_pool2d(F.relu(self.layer1(x)), 2)
        x = F.max_pool2d(F.relu(self.layer2(x)), 2)
        x = self.flat(x)
        # print(x.shape)
        x = self.fc1(x)
        ### END OF CODE
        return x
```

CNN architecture



```
2 0.000e+00 2.000e+00 1.000e+00 0.000e+00 2.000e+00 1.000e+00
0 3.000e+00 0.000e+00]
```

Confusion Matrix



Visualize weights and kernels

```
accuracies = []
examples = []
epsilons = [0, .05, .1, .15, .2, .25, .3]
# put model in eval mode
CNNTrainer.model.eval()

# update the helper class with a new batch size = 1
# for the dataloader. It doesnt change the model
# weights or any other parameter

opts = {
    'lr': 5e-4,
    'epochs': 5,
    'batch_size': 1 #this is the only change
}

CNNTrainer = TrainHelper(model = CNNTrainer.model,
                      train_set = train_dataset,
                      test_set = test_dataset,opts = opts)

device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")

# Run test for each epsilon
for eps in epsilons:
    acc, ex = eval_adversarial(CNNTrainer, device, eps)
    accuracies.append(acc)
    examples.append(ex)
```

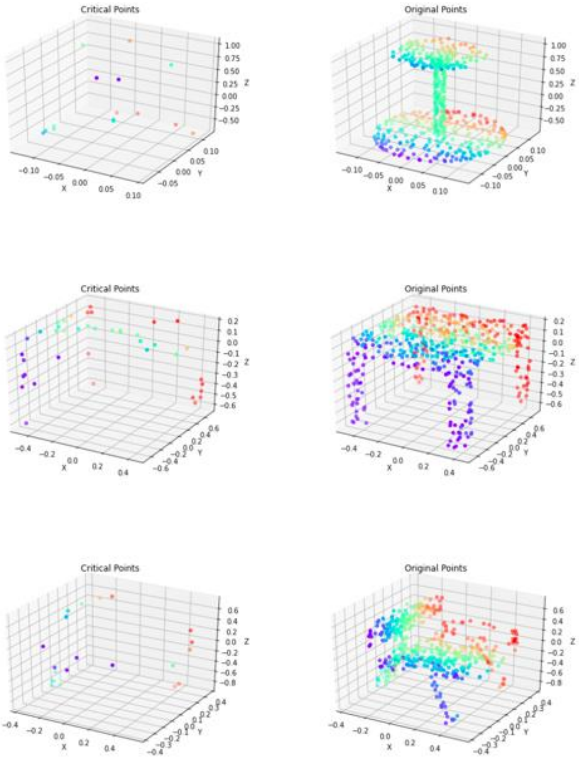
```
Epsilon: 0      Test Accuracy = 9919 / 10000 = 0.9919
Epsilon: 0.05   Test Accuracy = 9537 / 10000 = 0.9537
Epsilon: 0.1    Test Accuracy = 8328 / 10000 = 0.8328
Epsilon: 0.15   Test Accuracy = 5710 / 10000 = 0.571
Epsilon: 0.2    Test Accuracy = 3335 / 10000 = 0.3335
Epsilon: 0.25   Test Accuracy = 1896 / 10000 = 0.1896
Epsilon: 0.3    Test Accuracy = 1093 / 10000 = 0.1093
```

Test Accuracy with different level of adversarial attack

Output images with noise

```
    crit_point_dict = {}
    label_indices = [0,1,2,3,4]
    criterion = torch.nn.CrossEntropyLoss() # this allows you to not need the␣
↪TrainHelper Class

    for i, (data, labels) in enumerate(data_loader):
        # data.shape = [1, 500, 3]
        # labels.shape = [1]
        if labels.numpy() in label_indices:   # print one sample from each␣
↪class
            # create index in the dictionary and remove the class id from the␣
↪list of indices
            # we are looking for
            class_label = labels.numpy()[0]   # get class of current point␣
↪cloud
            label_indices.remove(class_label) # we don't print more samples␣
↪from that class
            crit_point_dict[class_label] = {}
            crit_point_dict[class_label]['indices'] = []
            crit_point_dict[class_label]['data'] = []

            # run sample through the network


            ### START CODE HERE ###
            data.requires_grad = True
            model.zero_grad()
            output = PtNet.forward(data)
            # ouput.shape = [1, 5]
            pred = output.max(1, keepdim=True)[1]
            loss = criterion(output, labels)
            loss.backward()
            print(loss)
            data_grad = data.grad.data
            # data_grad.shape = [1, 500, 3]
            # data_grad.squeeze(0).shape = [500, 3]
            data_squ = data_grad.squeeze(0)
            for i in range(data_squ.shape[0]):
              if data_squ[i][0].item() > 0 or data_squ[i][1].item() > 0 or␣
↪data_squ[i][2].item() > 0:
                crit_point_dict[class_label]['indices'].append(i)

            crit_point_dict[class_label]['data'] = data.squeeze(0).detach().
↪numpy()
```

Partial code of finding critical points in 3D data



Critical Points vs Original Points

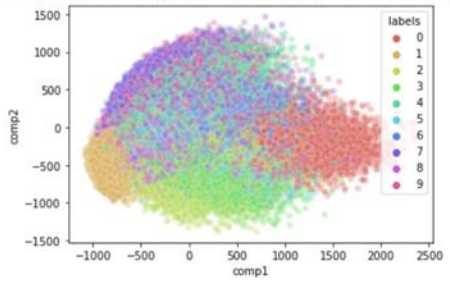
PCA, NN, CNN (Python with numpy, pytorch, matplotlib, pandas, sklearn)

Works I have done in this project:

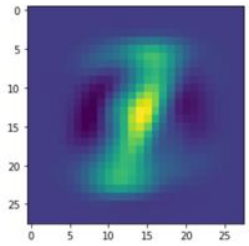
- Preprocess Dataset Using Principal Component Analysis (PCA)
  - Performed 2 different PCA methods: singular value decomposition and eigen decomposition
  - Projected data to different dimensions to see how well the digits would be represented after the implementation of PCA
- Neural Network for Regression
  - Built a MultiLayer Perceptron (MLP) for fitting a line to random data
  - Decided the right optimizer; chose an appropriate loss function for the learning task; and trained the model over the given data
- Corner Detection using Convolution Neural Network (CNN)
  - Developed a CNN model to detect corners in the collection of images and output the corresponding coordinate of the corner in the image
  - Built convolution and maxpool layers; pick non-linear activation function for each layer in the model

```
[8]  ### YOUR CODE HERE
     import seaborn as sns
     projected = np.matmul(data_centered, V.T[:,:2])
     df = pd.DataFrame(data = projected, columns=['comp1', 'comp2'])
     df['labels'] = pd.Series(labels)
     sns.scatterplot(
         x="comp1", y="comp2",
         hue="labels",
         palette=sns.color_palette("hls", 10),
         data=df,
         legend="full",
         alpha=0.3
     )
     ### END OF CODE
```

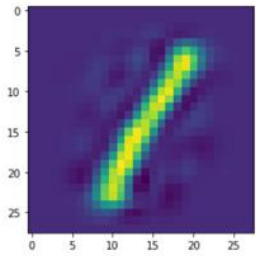
```
<matplotlib.axes._subplots.AxesSubplot at 0x7faef9eecf98>
```

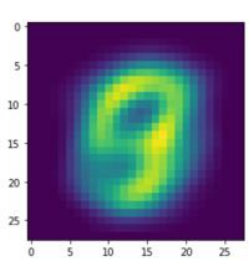


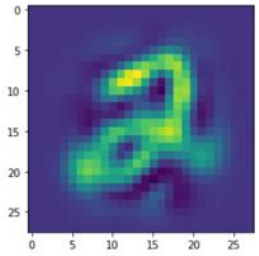
Data distribution in space spanned by 2 principal components



Data in 2 dimensions

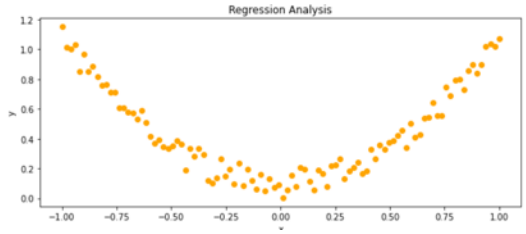




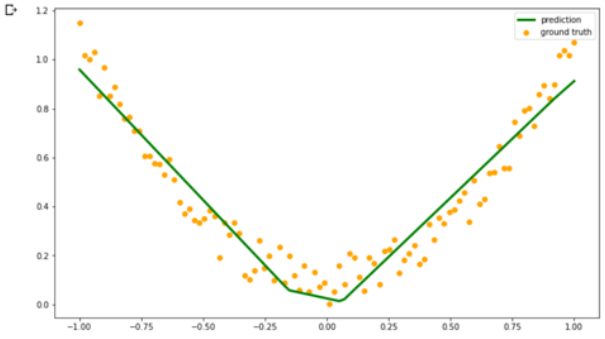


Data in 50 dimensions

```
torch.manual_seed(1)    # reproducible

x = torch.unsqueeze(torch.linspace(-1, 1, 100), dim=1)  # x data (tensor), shap
y = x.pow(2) + 0.2*torch.rand(x.size())                 # noisy y data (tensor)

# torch can only train on Variable, so convert them to Variable
x, y = Variable(x), Variable(y)

# view data
plt.figure(figsize=(10,4))
plt.scatter(x.data.numpy(), y.data.numpy(), color = "orange")
plt.title('Regression Analysis')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

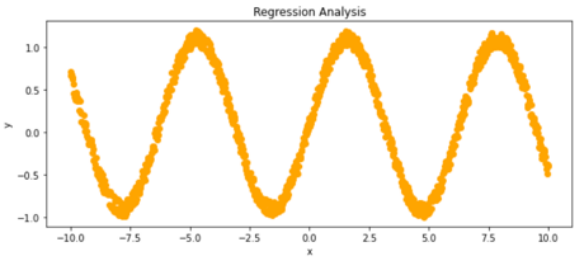



```
fig, ax = plt.subplots(figsize=(12,7))
ax.scatter(x.data.numpy(), y.data.numpy(), color = "orange", label='ground truth')
ax.plot(x.data.numpy(), prediction.data.numpy(), 'g-', lw=3, label='prediction')
plt.legend()
plt.show()
```

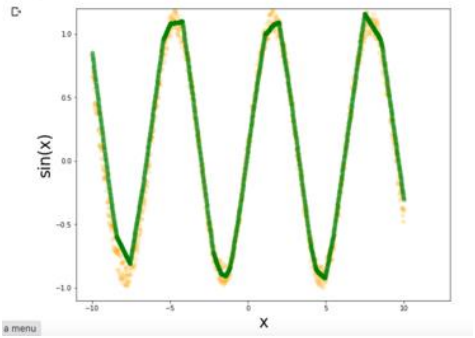


Neural Network for regression

```
[20]
    x = torch.unsqueeze(torch.linspace(-10, 10, 1000), dim=1)  # x data (tensor),
    y = torch.sin(x) + 0.2*torch.rand(x.size())                 # noisy y data (t

    x, y = Variable(x), Variable(y)

    plt.figure(figsize=(10,4))
    plt.scatter(x.data.numpy(), y.data.numpy(), color = "orange")
    plt.title('Regression Analysis')
    plt.xlabel('x')
    plt.ylabel('y')
    plt.show()
```

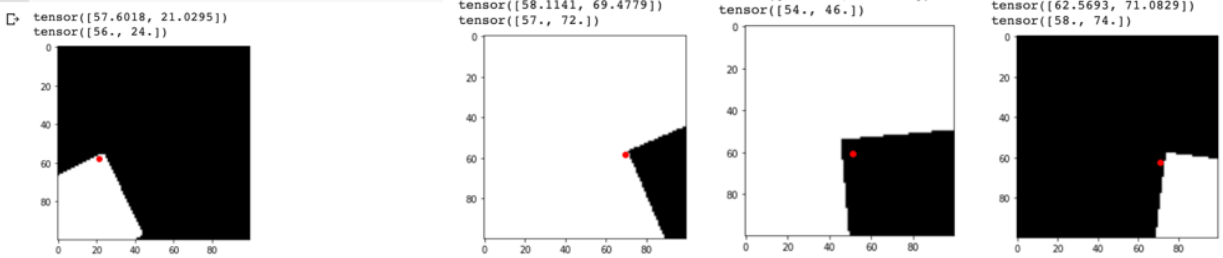



```
fig, ax = plt.subplots(figsize=(10,8))
# ax.set_title('Regression Analysis - model 3, Batches', fontsize=35)
ax.set_xlabel('x', fontsize=24)
ax.set_ylabel('sin(x)', fontsize=24)
ax.set_xlim(-11.0, 13.0)
ax.set_ylim(-1.1, 1.2)
ax.scatter(x.data.numpy(), y.data.numpy(), color = "orange", alpha=0.2)
prediction = net(x)     # input x and predict based on x
ax.scatter(x.data.numpy(), prediction.data.numpy(), color='green', alpha=0.5)
plt.show()
```



Implement MLP to fit the sine function

```
with torch.no_grad():
  for i,(x,y) in enumerate(testloader):
    x = torch.unsqueeze(x, 1)
    coords = net(x)

    for j in range(10):
      fig, ax = plt.subplots()
      ax.imshow(x[j][0], cmap=plt.cm.gray)
      pts = y[j]
      pts = coords[j] * 100
      print(pts)
      print(y[j])
      ax.plot(pts[1], pts[0], color='red', marker='o',
              linestyle='None', markersize=6)
      plt.show()

    break
```



Corner Detection

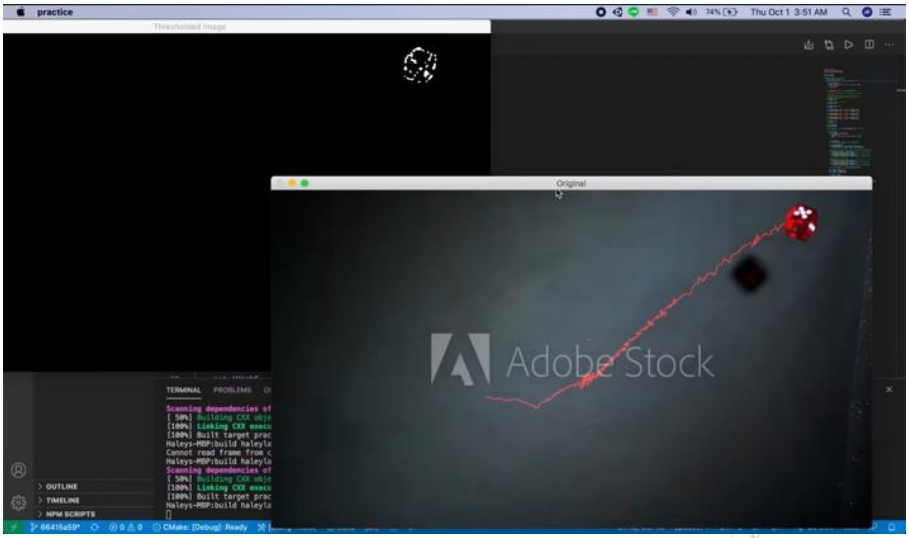
Object Tracking (OpenCV with C++)

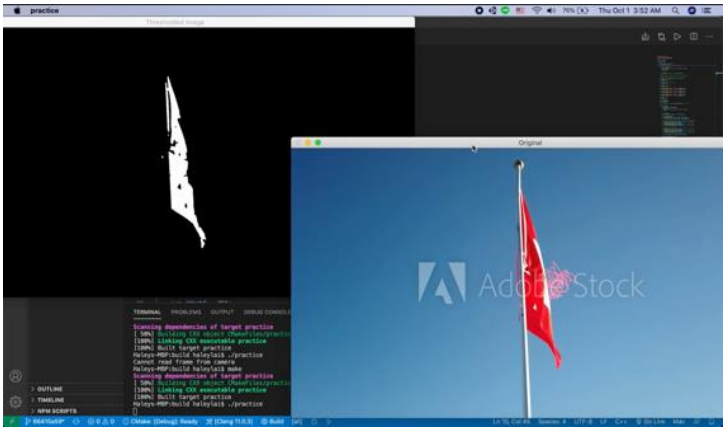
- A color and shape-based object detection program

In the following example, the program is targeting red objects specifically

What does the program do:

- Thresholded Image (left window in the figures):
  - By applying multiple image filters on each frame of the video, the program is able to segment out the main object in the video.
  - All unrelated backgrounds or objects will be filtered out during the image processing step (image filters)
- Original Image (right window in the figures):
  - Find the exact position of the target object in each frame
  - Draw a line along the trajectory of the object



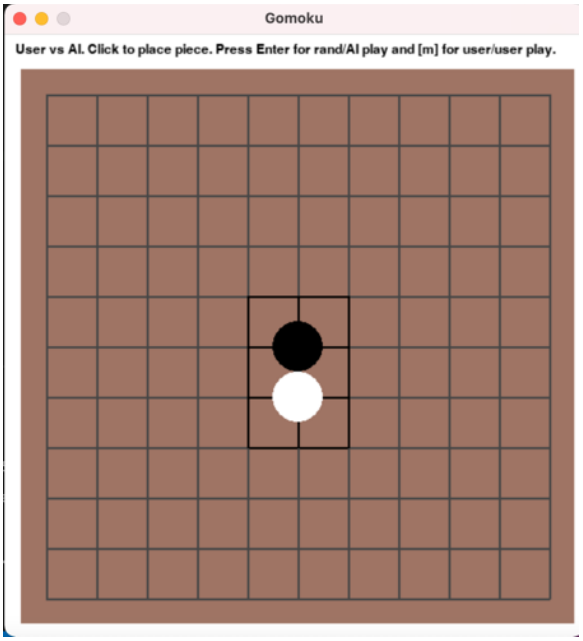



```
cv::Mat imgLines = cv::Mat::zeros(imgtemp.size(), CV_8UC3);

while (true) {
    cv::Mat image;
    bool success = cap.read(image);
    if (!success) {
        cout << "Cannot read frame from camera" << endl;
        break;
    }

    cv::Mat imgHSV;
    cv::cvtColor(image, imgHSV, cv::COLOR_BGR2HSV);

    cv::Mat imgThresholded;
    cv::inRange(imgHSV, cv::Scalar(iLowH, iLowS, iLowV),
        cv::Scalar(iHighH, iHighS, iHighV), imgThresholded);

    // elimate close objects which have the same color as our main object
    // Morphological opening: apply erosion, followed by the dilation
    cv::erode(imgThresholded, imgThresholded,
        cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(5, 5)));
    cv::dilate(imgThresholded, imgThresholded,
        cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(5, 5)));

    // elimate noises in the image
    // Morphological closing: apply dilation, followed by the erosion
    cv::dilate(imgThresholded, imgThresholded,
        cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(5, 5)));
    cv::erode(imgThresholded, imgThresholded,
        cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(5, 5)));

    cv::Moments oMoments = cv::moments(imgThresholded);
```

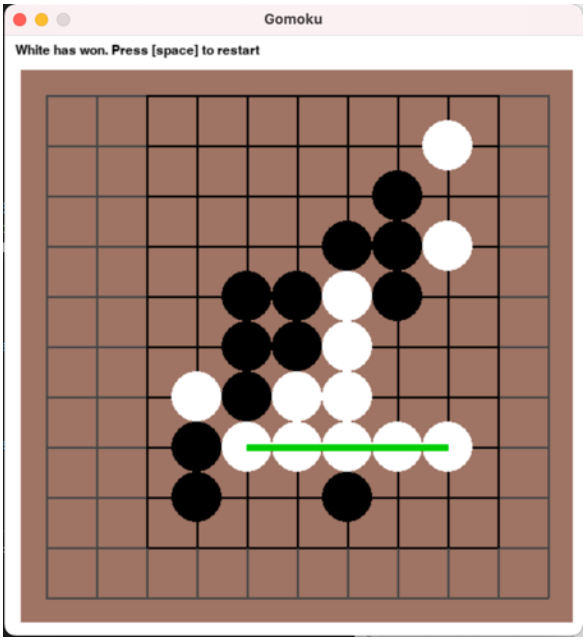
AI: Search and Reasoning

Gomoku AI (Python with pygame)

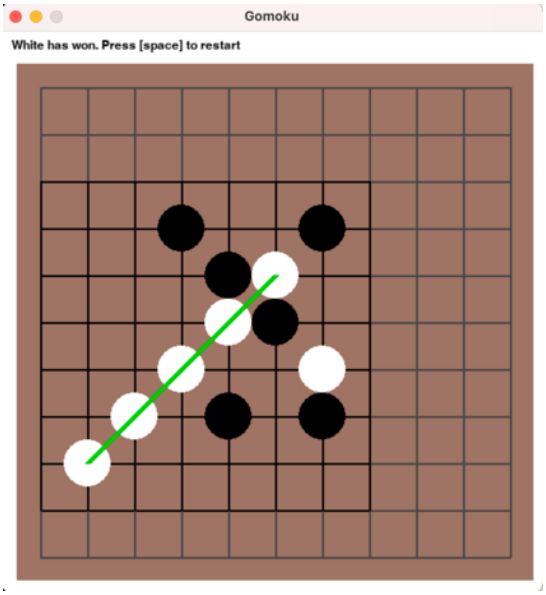
- Implement Monte Carlo Tree Search (MCTS) to play Gomoku



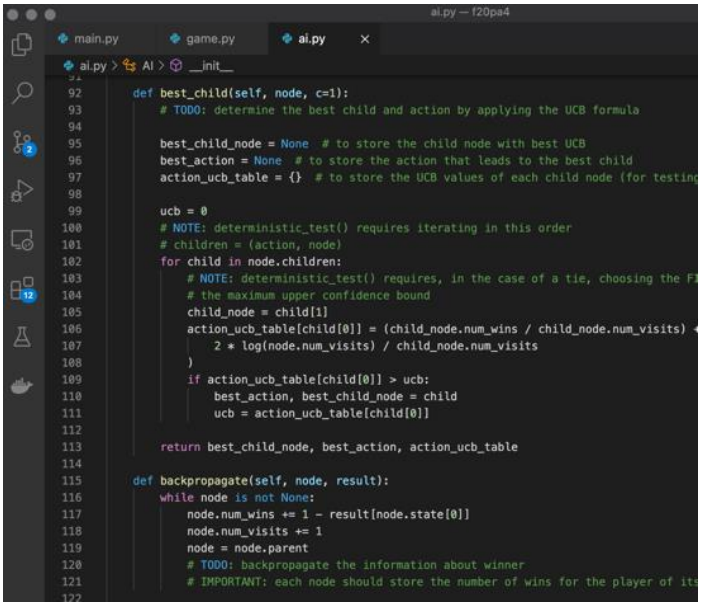
Start Board



Player vs AI (AI won)



AI vs AI (White won)

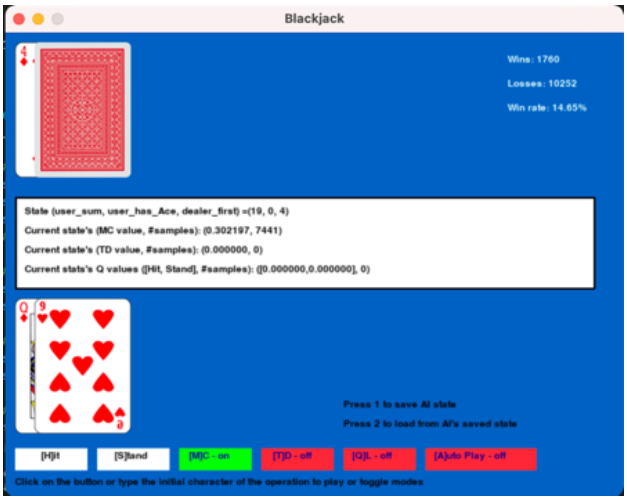


```
def best_child(self, node, c=1):
    # TODO: determine the best child and action by applying the UCB formula

    best_child_node = None  # to store the child node with best UCB
    best_action = None  # to store the action that leads to the best child
    action_ucb_table = {}  # to store the UCB values of each child node (for testing

    ucb = 0
    # NOTE: deterministic_test() requires iterating in this order
    # children = (action, node)
    for child in node.children:
        # NOTE: deterministic_test() requires, in the case of a tie, choosing the F
        # the maximum upper confidence bound
        child_node = child[1]
        action_ucb_table[child[0]] = (child_node.num_wins / child_node.num_visits)
            2 * log(node.num_visits) / child_node.num_visits
        )
        if action_ucb_table[child[0]] > ucb:
            best_action, best_child_node = child
            ucb = action_ucb_table[child[0]]

    return best_child_node, best_action, action_ucb_table

def backpropagate(self, node, result):
    while node is not None:
        node.num_wins += 1 - result[node.state[0]]
        node.num_visits += 1
        node = node.parent
        # TODO: backpropagate the information about winner
        # IMPORTANT: each node should store the number of wins for the player of its
```

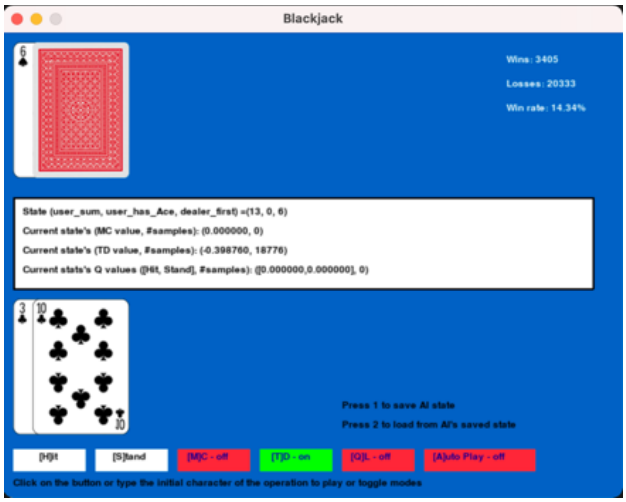
Partial code of AI (picking best action out of samples derived from simulations)

[Blackjack AI (Python with pygame)]

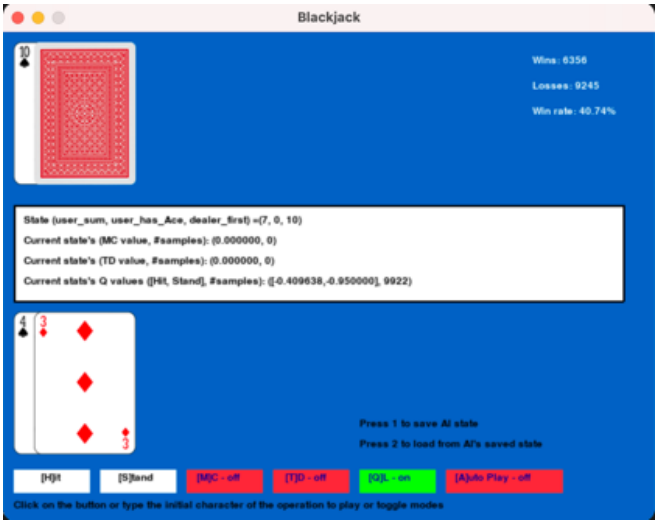
- Implement Monte Carlo policy evaluation, Temporal Difference policy evaluation, and Q-Learning to play Blackjack



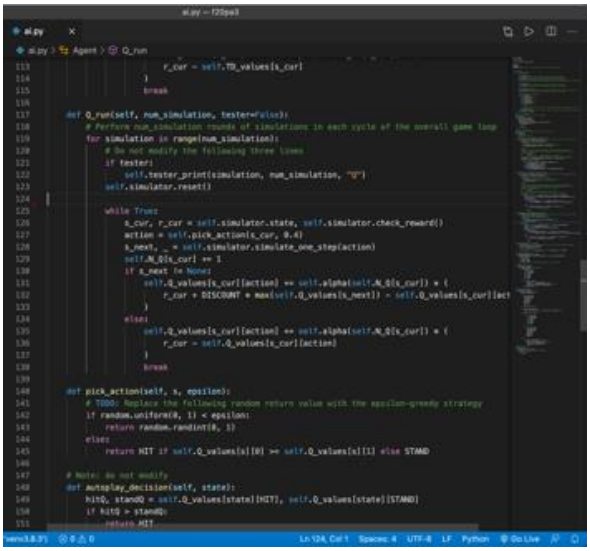
Screenshot of game running under Monte Carlo Policy



Game running under Temporal-Difference Policy



Game running under Q-learning algorithm



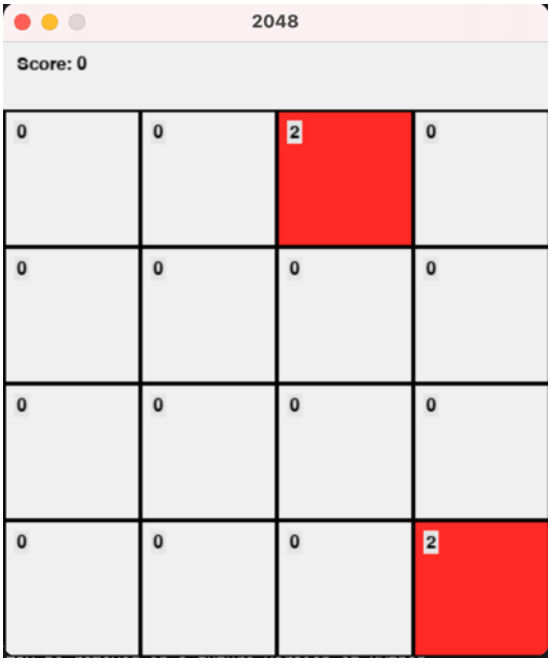
Partial code of Q-learning algorithm

[2048 AI (Python with pygame)]

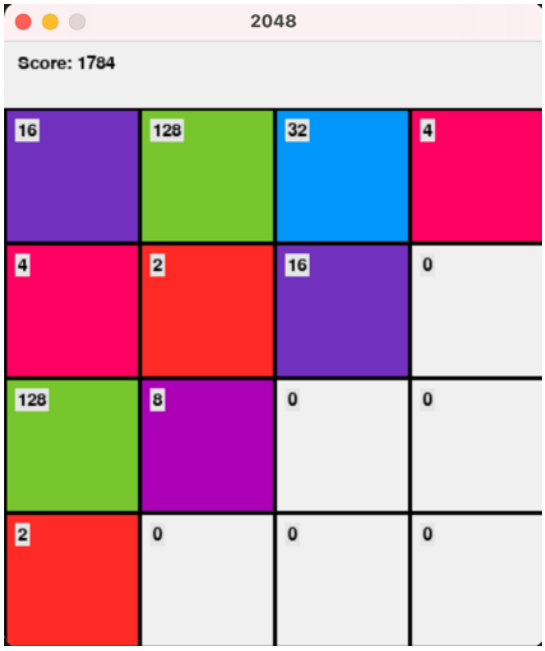
- Implement expectimax search method to play 2048

Description:

The program will simulate as many as possible game results based on actions that can be taken for the current game status, and it returns the best action that should be taken by the player

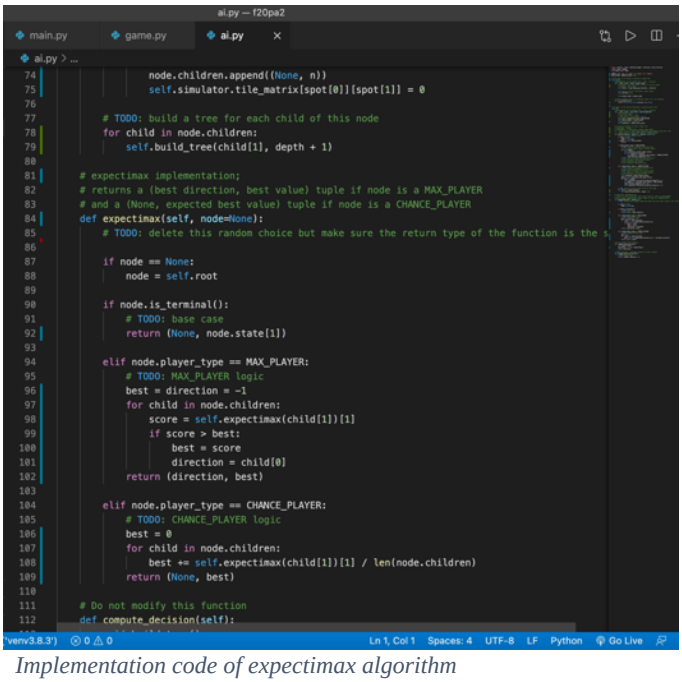


Start board



Middle game

End game



Implementation code of expectimax algorithm

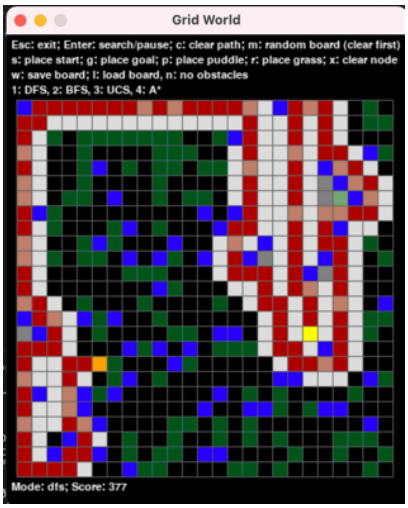
Grid World (Python with pygame)

- Find an existing path or the shortest path from the start (yellow node) to the goal (orange node) based on the pathfinding algorithm that is selected during the runtime

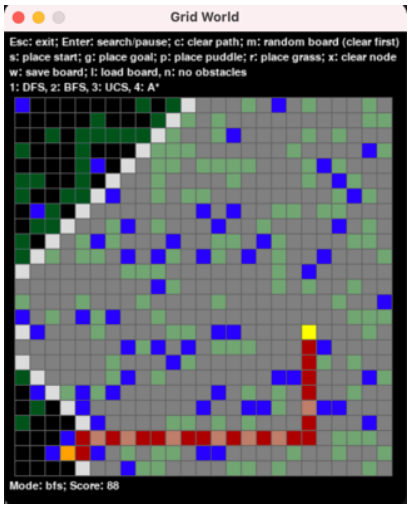
The program provides the following searching strategies:

- DFS
- BFS
- Uniform Cost Search (Dijkstra)
- A* Search using Manhattan Distance as the heuristic

Representation of colored grid in the figure:
Puddle (Blue grid): player cannot go pass it
Grass (Green grid): has 10 costs
Start (Yellow grid): start point
Goal (Orange grid): destination



DFS mode



BFS mode

Uniform Cost Search mode

A mode*

Partial code of A implementation*

2020 Summer Internship – Game Development

Zombie Runner Game (C# with Unity)

- A classic 2D runner game made by Unity

How to play:

- Hitting slide and jump buttons to dodge obstacles
- Filling Zombie Bite by absorbing spirits in the game, use the skill to defend attacks from Daoshi
- Survive to the end

Zombie Downstairs Game (C# with Unity)

- A classic 2D platform game made by Unity

How to play:

- Moving zombie to left and right to fall from platform to platform to not be hurt by the lightning trap from above
- Collecting enough spirits from human by attacking them to pass the game

Game Interface:
Movement: moving Zombie to left or right by hitting the left / right side of the screen

Health bar (top left): will reduce when hitting an obstacle. Game over when it reaches zero

Human spirits (top right): number of spirits the zombie has collected

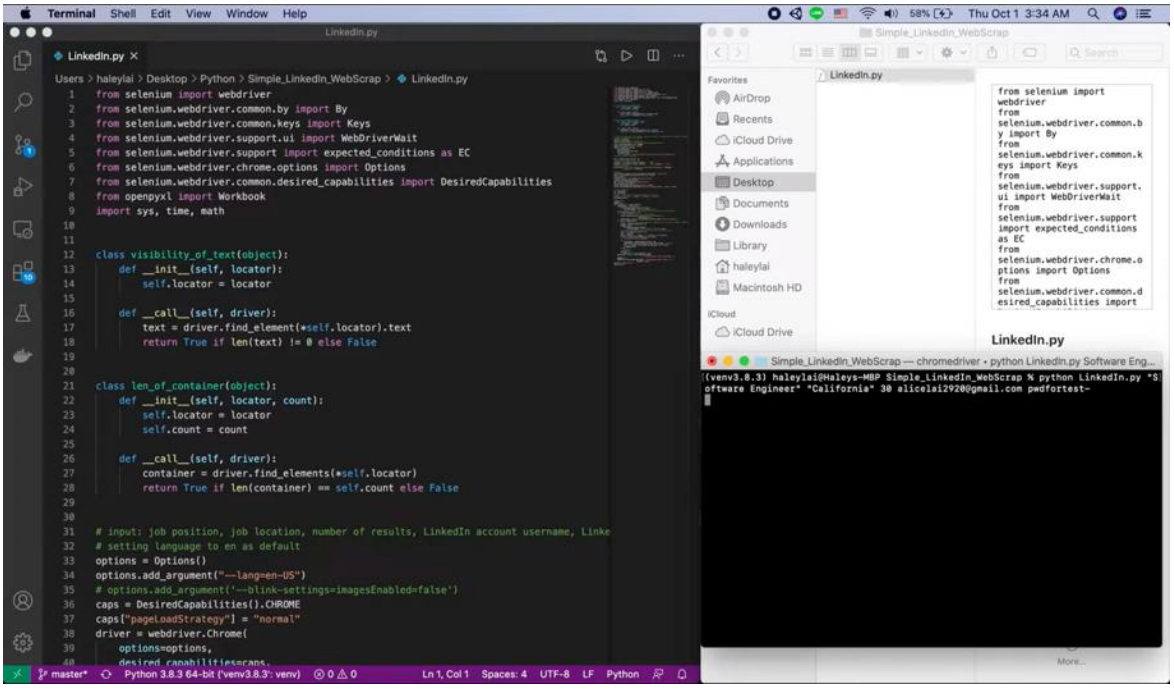
Zombie attacks human

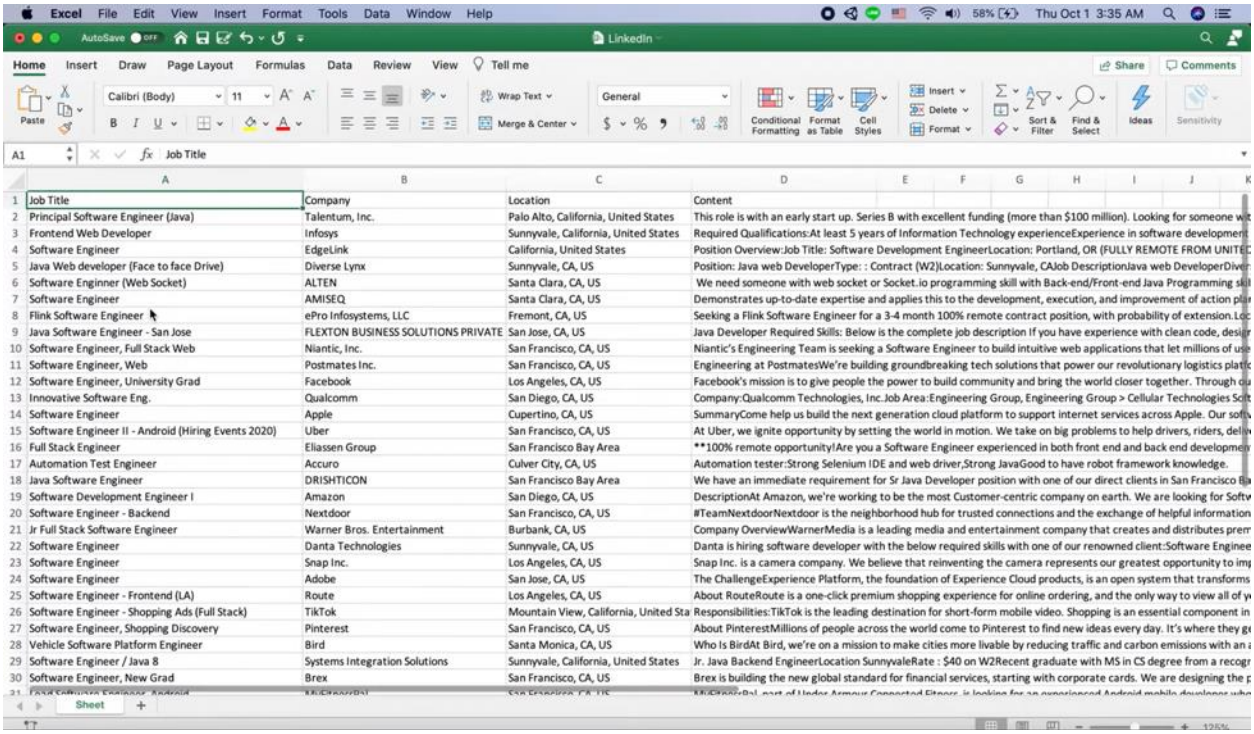
Zombie gets hurt by the trap

Web Scrapper

LinkedIn Web Scrap (Python, Selenium)

- Scraping job info from LinkedIn, including job title, company, location, and job content

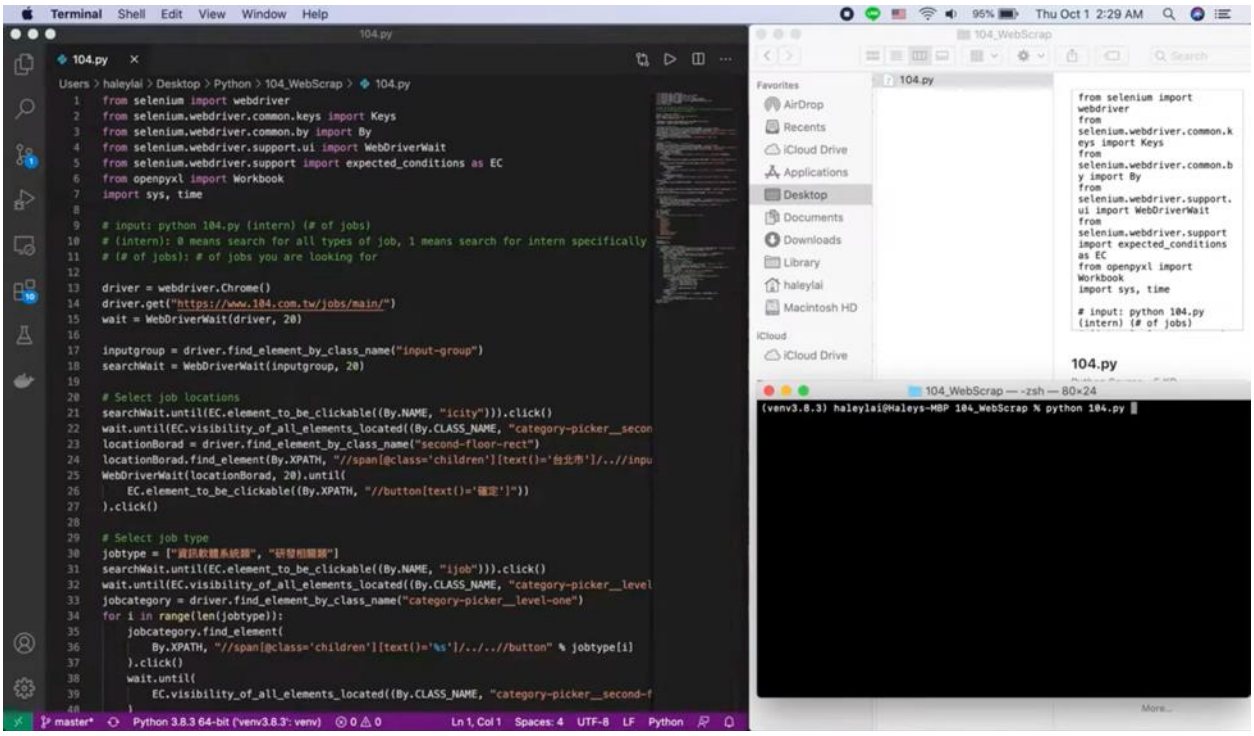


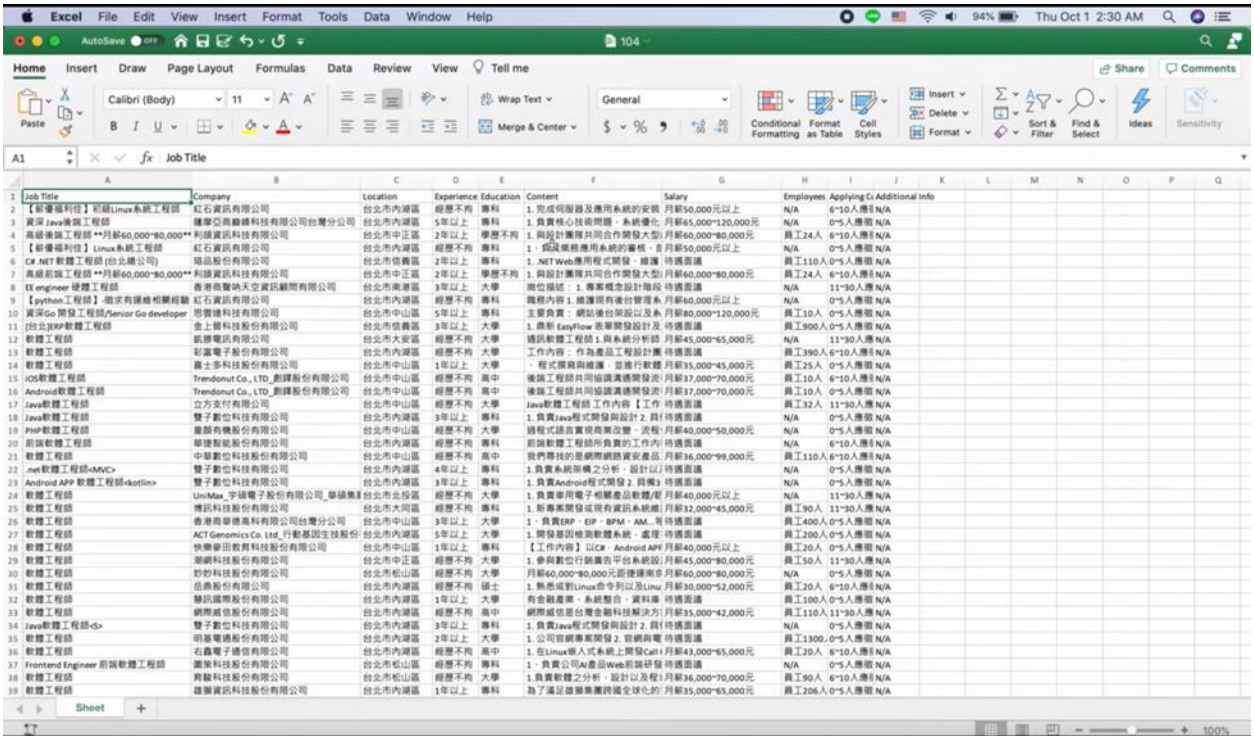


Job Title	Company	Location	Content
Principal Software Engineer (Java)	Talentum, Inc.	Palo Alto, California, United States	This role is with an early start up. Series B with excellent funding (more than $100 million). Looking for someone w
Frontend Web Developer	Infosys	Sunnyvale, California, United States	Required Qualifications:At least 5 years of Information Technology experienceExperience in software development
Software Engineer	EdgeLink	California, United States	Position Overview:Job Title: Software Development EngineerLocation: Portland, OR (FULLY REMOTE FROM UNITE
Java Web developer (Face to face Drive)	Diverse Lynx	Sunnyvale, CA, US	Position: Java web DeveloperType: : Contract (W2)Location: Sunnyvale, CAJob DescriptionJava web DeveloperDiver
Software Enginner (Web Socket)	ALTEN	Santa Clara, CA, US	We need someone with web socket or Socket.io programming skill with Back-end/Front-end Java Programming sk
Software Engineer	AMISEQ	Santa Clara, CA, US	Demonstrates up-to-date expertise and applies this to the development, execution, and improvement of action pla
Flink Software Engineer	ePro Infosystems, LLC	Fremont, CA, US	Seeking a Flink Software Engineer for a 3-4 month 100% remote contract position, with probability of extension.Loc
Java Software Engineer - San Jose	FLEXTON BUSINESS SOLUTIONS PRIVATE	San Jose, CA, US	Java Developer Required Skills: Below is the complete job description If you have experience with clean code, desi
Software Engineer, Full Stack Web	Niantic, Inc.	San Francisco, CA, US	Niantic's Engineering Team is seeking a Software Engineer to build intuitive web applications that let millions of us
Software Engineer, Web	Postmates Inc.	San Francisco, CA, US	Engineering at PostmatesWe're building groundbreaking tech solutions that power our revolutionary logistics platf
Software Engineer, University Grad	Facebook	Los Angeles, CA, US	Facebook's mission is to give people the power to build community and bring the world closer together. Through cu
Innovative Software Eng.	Qualcomm	San Diego, CA, US	Company:Qualcomm Technologies, Inc.Job Area:Engineering Group, Engineering Group > Cellular Technologies Scri
Software Engineer	Apple	Cupertino, CA, US	SummaryCome help us build the next generation cloud platform to support internet services across Apple. Our soft
Software Engineer II - Android (Hiring Events 2020)	Uber	San Francisco, CA, US	At Uber, we ignite opportunity by setting the world in motion. We take on big problems to help drivers, riders, del
Full Stack Engineer	Eliassen Group	San Francisco Bay Area	**100% remote opportunity!Are you a Software Engineer experienced in both front end and back end developme
Automation Test Engineer	Accuro	Culver City, CA, US	Automation tester:Strong Selenium IDE and web driver,Strong JavaGood to have robot framework knowledge.
Java Software Engineer	DRISHTICON	San Francisco Bay Area	We have an immediate requirement for Sr Java Developer position with one of our direct clients in San Francisco Ba
Software Development Engineer I	Amazon	San Diego, CA, US	DescriptionAt Amazon, we're working to be the most Customer-centric company on earth. We are looking for Softw
Software Engineer - Backend	Nextdoor	San Francisco, CA, US	#TeamNextdoorNextdoor is the neighborhood hub for trusted connections and the exchange of helpful information
Jr Full Stack Software Engineer	Warner Bros. Entertainment	Burbank, CA, US	Company OverviewWarnerMedia is a leading media and entertainment company that creates and distributes prem
Software Engineer	Danta Technologies	Sunnyvale, CA, US	Danta is hiring software developer with the below required skills with one of our renowned client:Software Enginee
Software Engineer	Snap Inc.	Los Angeles, CA, US	Snap Inc. is a camera company. We believe that reinventing the camera represents our greatest opportunity to imp
Software Engineer	Adobe	San Jose, CA, US	The ChallengeExperience Platform, the foundation of Experience Cloud products, is an open system that transforms
Software Engineer - Frontend (LA)	Route	Los Angeles, CA, US	About RouteRoute is a one-click premium shopping experience for online ordering, and the only way to view all of y
Software Engineer - Shopping Ads (Full Stack)	TikTok	Mountain View, California, United Sta	Responsibilities:TikTok is the leading destination for short-form mobile video. Shopping is an essential component in
Software Engineer, Shopping Discovery	Pinterest	San Francisco, CA, US	About PinterestMillions of people across the world come to Pinterest to find new ideas every day. It's where they g
Vehicle Software Platform Engineer	Bird	Santa Monica, CA, US	Who Is BirdAt Bird, we're on a mission to make cities more livable by reducing traffic and carbon emissions with an
Software Engineer / Java 8	Systems Integration Solutions	Sunnyvale, California, United States	Jr. Java Backend EngineerLocation SunnyvaleRate : $40 on W2Recent graduate with MS in CS degree from a recogr
Software Engineer, New Grad	Brex	San Francisco, CA, US	Brex is building the new global standard for financial services, starting with corporate cards. We are designing the

[104 Web Scrap (Python, Selenium)]

- Scraped job info from 104.com, including job title, company, location, experience, content, employees, applying condition, and additional info





Web Development – Personal Website

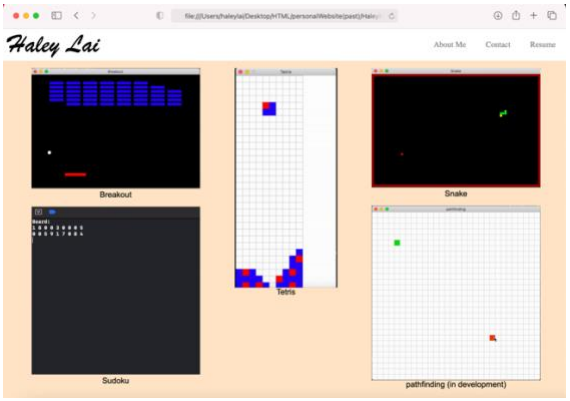
My first version website was made using only HTML and CSS. From time to time, I have been learning new web libraries and tools to update the website's appearance and functionality.

Here are some works I have done throughout the 3 versions of my website:

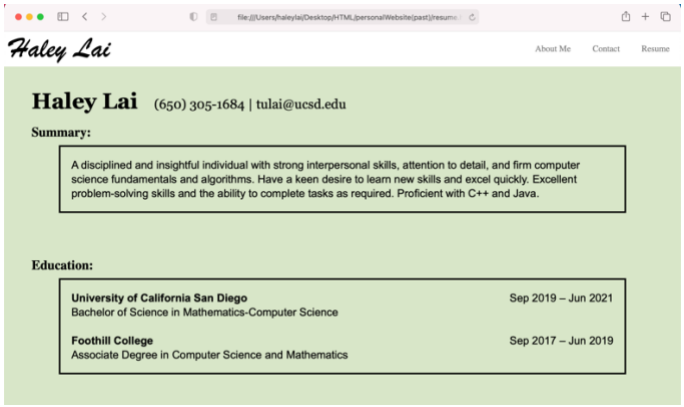
- Built backend server to handle client requests
- Learned database service such as MongoDB Atlas and Firebase Realtime DB to manage my portfolio page
- Integrated domain service provider with a cloud service provider to deploy my website with a custom domain
- Learned React.js to make a better user interface

1st version (HTML, CSS)

Hosting platform: Heroku



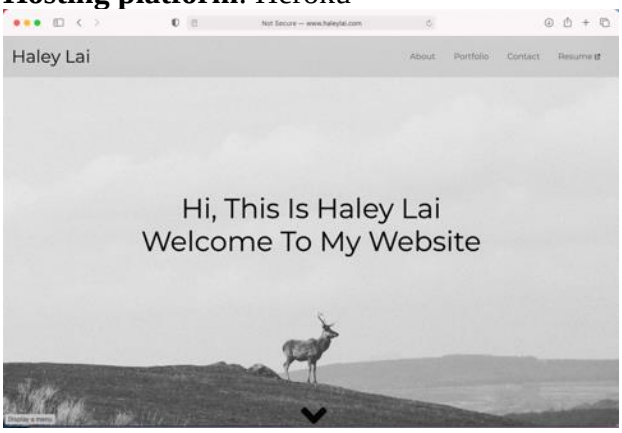
Home Page



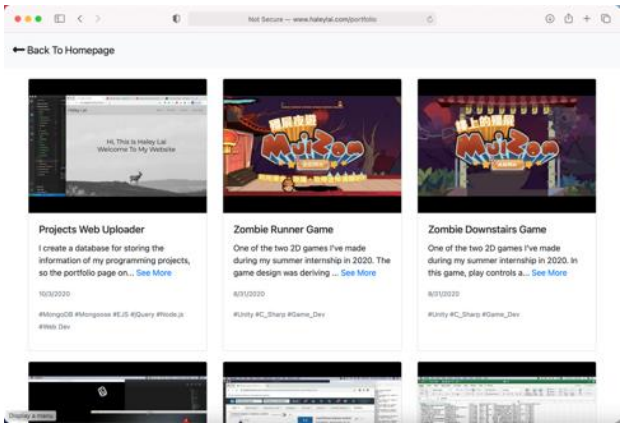
Resume Page

2nd version (Bootstrap 4, Node.js, MongoDB, EJS, Express.js, JavaScript, jQuery)

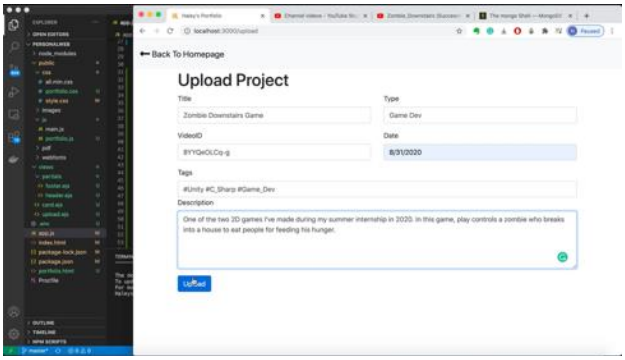
Hosting platform: Heroku



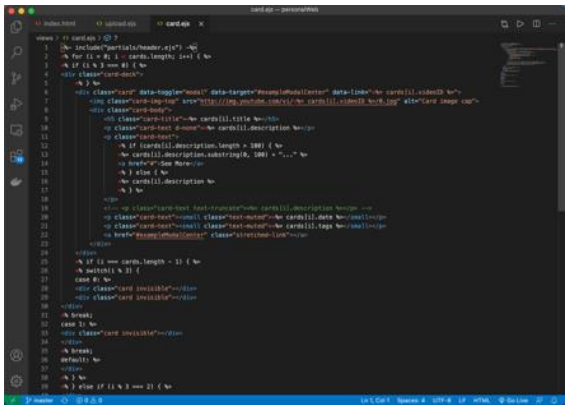
Home Page



Portfolio Page



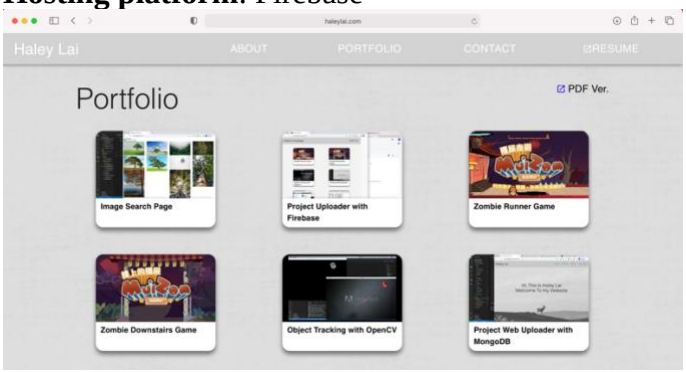
Portfolio page uploader



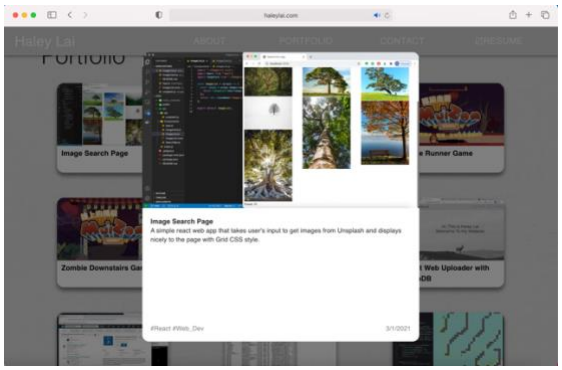
Partial code of project page uploader

3rd version (React.js, Firebase Realtime DB, JavaScript, Material-UI, Sass)

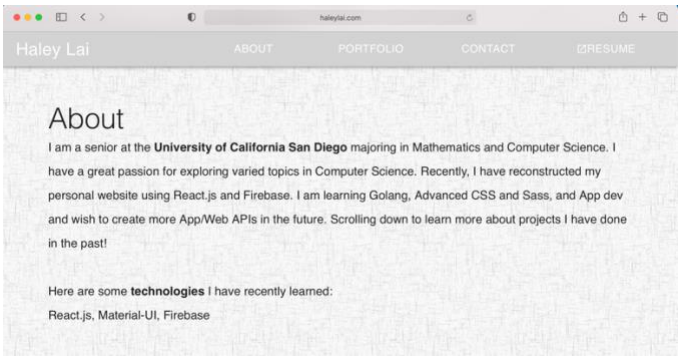
Hosting platform: Firebase



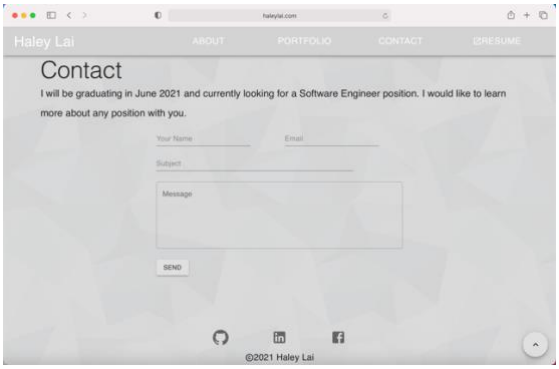
Portfolio page



Project showcase



About page

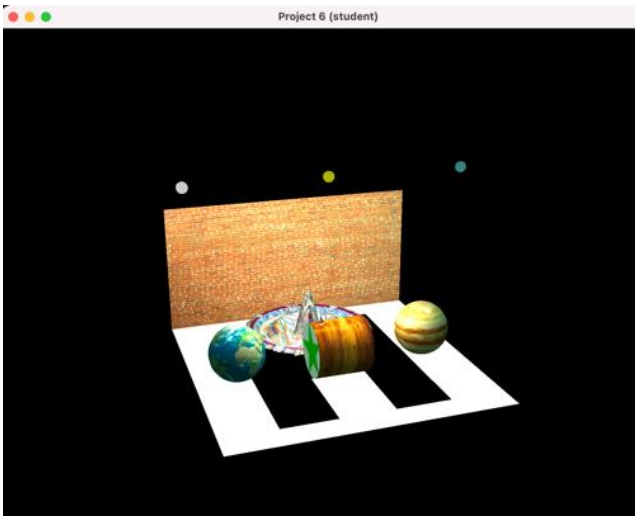


Contact page

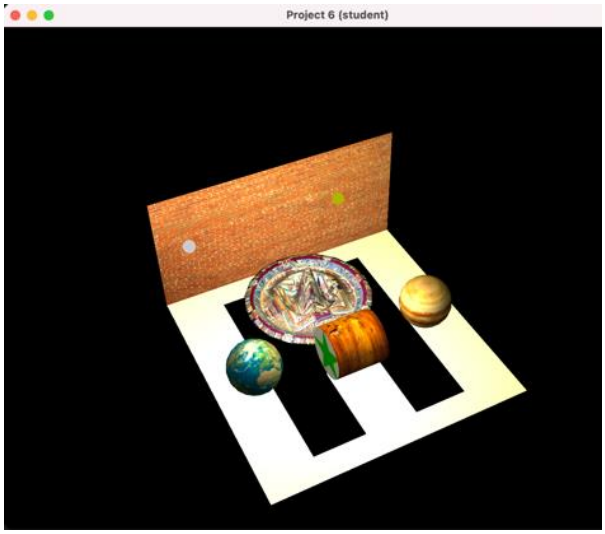
Computer Graphic

Texture Maps (C++ with OpenGL)

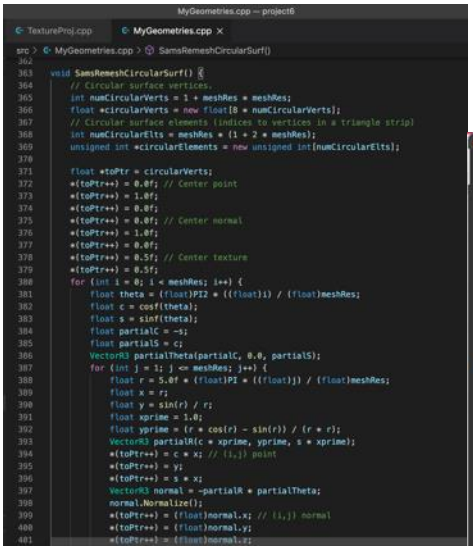
- Add textures to spheres, a rectangle, a cylinder and a surface of rotation.
- Add texture coordinates to the surface of rotation.
- Write a procedural texture for the floor of the scene.



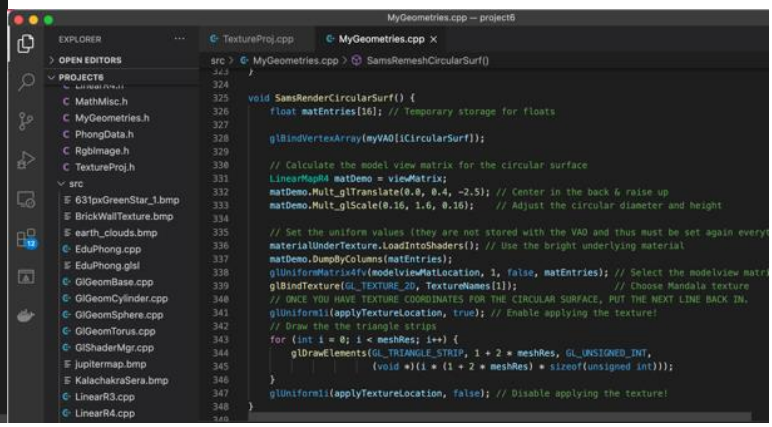
Objects with text



Toggle ambient lighting off



```
void SamsReneshCircularSurf() {
    // Circular surface vertices.
    int numCircularVerts = 1 + meshRes * meshRes;
    float *circularVerts = new float[8 * numCircularVerts];
    // Circular surface elements (indices to vertices in a triangle strip)
    int numCircularElts = meshRes * (1 + 2 * meshRes);
    unsigned int *circularElements = new unsigned int[numCircularElts];

    float *toPtr = circularVerts;
    *(toPtr++) = 0.0f; // Center point
    *(toPtr++) = 1.0f;
    *(toPtr++) = 0.0f;
    *(toPtr++) = 0.0f; // Center normal
    *(toPtr++) = 1.0f;
    *(toPtr++) = 0.0f;
    *(toPtr++) = 0.5f; // Center texture
    *(toPtr++) = 0.5f;
    for (int i = 0; i < meshRes; i++) {
        float theta = (float)PI2 * ((float)i) / (float)meshRes;
        float c = cosf(theta);
        float s = sinf(theta);
        float partialC = -s;
        float partialS = c;
        VectorR3 partialTheta(partialC, 0.0, partialS);
        for (int j = 1; j <= meshRes; j++) {
            float r = 5.0f * (float)PI * ((float)j) / (float)meshRes;
            float x = r;
            float y = sin(r) / r;
            float xprime = 1.0;
            float yprime = (r * cos(r) - sin(r)) / (r * r);
            VectorR3 partialR(c * xprime, yprime, s * xprime);
            *(toPtr++) = c * x; // (i,j) point
            *(toPtr++) = y;
            *(toPtr++) = s * x;
            VectorR3 normal = -partialR * partialTheta;
            normal.Normalize();
            *(toPtr++) = (float)normal.x; // (i,j) normal
            *(toPtr++) = (float)normal.y;
            *(toPtr++) = (float)normal.z;
```



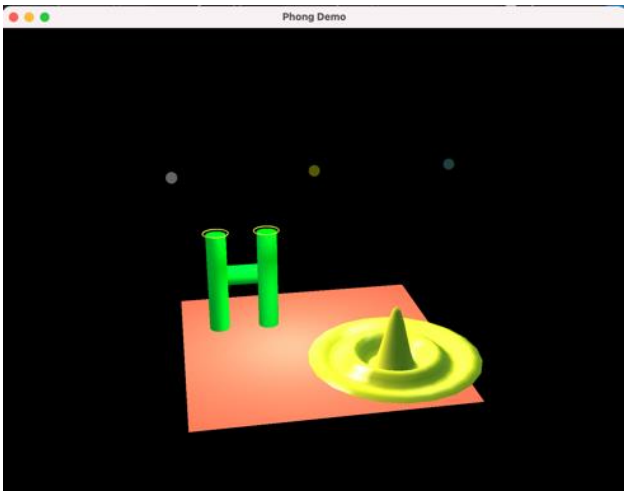
```
void SamsRenderCircularSurf() {
    float matEntries[16]; // Temporary storage for floats

    glBindVertexArray(myVAO[iCircularSurf]);

    // Calculate the model view matrix for the circular surface
    LinearMapR4 matDemo = viewMatrix;
    matDemo.Mult_glTranslate(0.0, 0.4, -2.5); // Center in the back & raise up
    matDemo.Mult_glScale(0.16, 1.6, 0.16);    // Adjust the circular diameter and height

    // Set the uniform values (they are not stored with the VAO and thus must be set again every
    materialUnderTexture.LoadIntoShaders(); // Use the bright underlying material
    matDemo.DumpByColumns(matEntries);
    glUniformMatrix4fv(modelviewMatLocation, 1, false, matEntries); // Select the modelview matri
    glBindTexture(GL_TEXTURE_2D, TextureNames[1]);                  // Choose Mandala texture
    // ONCE YOU HAVE TEXTURE COORDINATES FOR THE CIRCULAR SURFACE, PUT THE NEXT LINE BACK IN.
    glUniform1i(applyTextureLocation, true); // Enable applying the texture!
    // Draw the the triangle strips
    for (int i = 0; i < meshRes; i++) {
        glDrawElements(GL_TRIANGLE_STRIP, 1 + 2 * meshRes, GL_UNSIGNED_INT,
                       (void *)(i * (1 + 2 * meshRes) * sizeof(unsigned int)));
    }
    glUniform1i(applyTextureLocation, false); // Disable applying the texture!
}
```

Calculate texture coordinate for circular surface

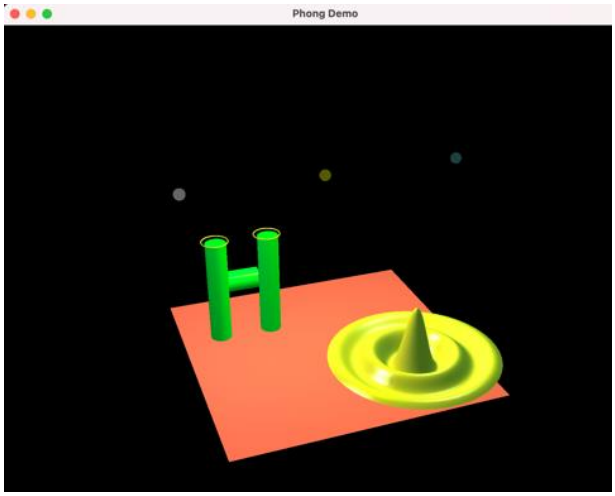
Add texture to circular surface

Phong Lighting (C++ with OpenGL)

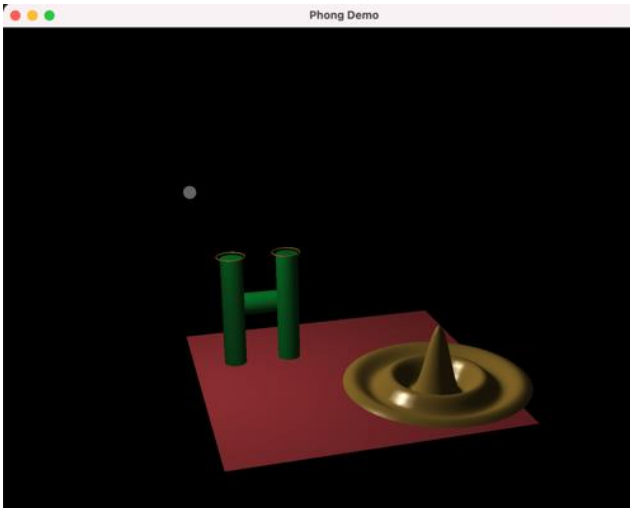
- Use illumination and shading to make scene look more three-dimensional.
- Learn how to shade objects with the Phong lighting model in OpenGL.
- Create four lights, including a spotlight, and create three materials.



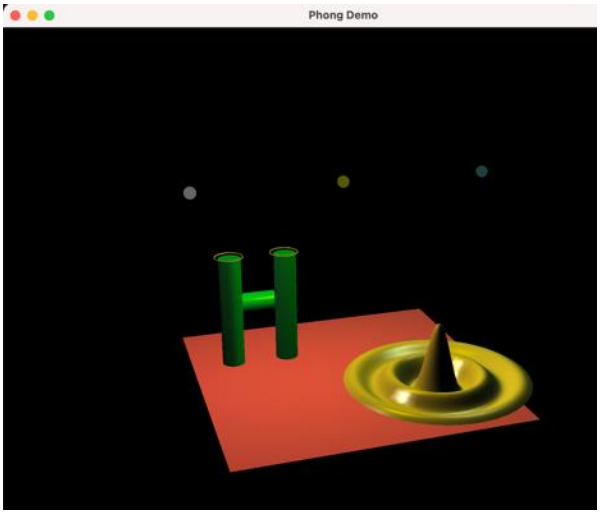
All lights are on



Turn spotlight off



Only light1 is on

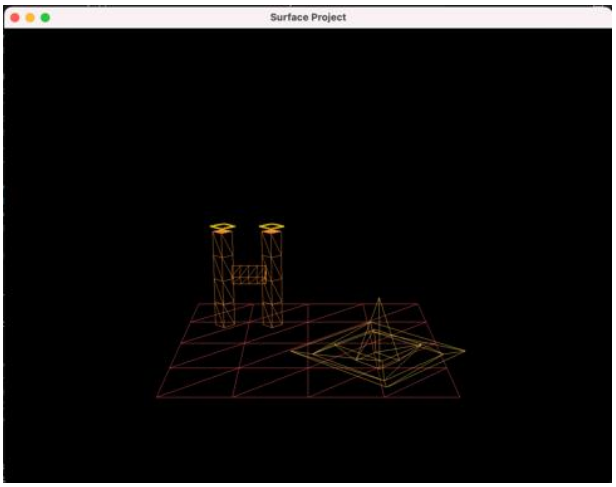


Toggle ambient lighting off

Surface of Rotation and Normals (C++ with OpenGL)

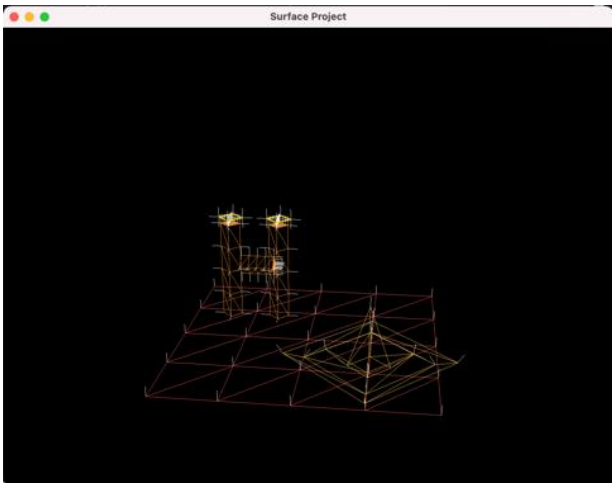
- Create a parametric surface (namely, a surface of rotation) using triangle strips.
- Create a rectangular mesh for the ground plane using triangle strips.
- Dynamically change the resolution of the surface of rotation and the ground plane.
- Correctly calculate normal for both objects.
- Discover that wireframe objects, especially when combined with animation can look very three-dimensional.
- Discover, however, that flat, solidly colored objects look much too flat and non-three-dimensional.

Key press events:
'w': toggle wireframe mode
'c': toggle backfaces culling
'M' or 'm': increases or decreases the mesh resolution
'N' or 'n': visualizing surface normals



Scene with low resolution

Scene with high resolution



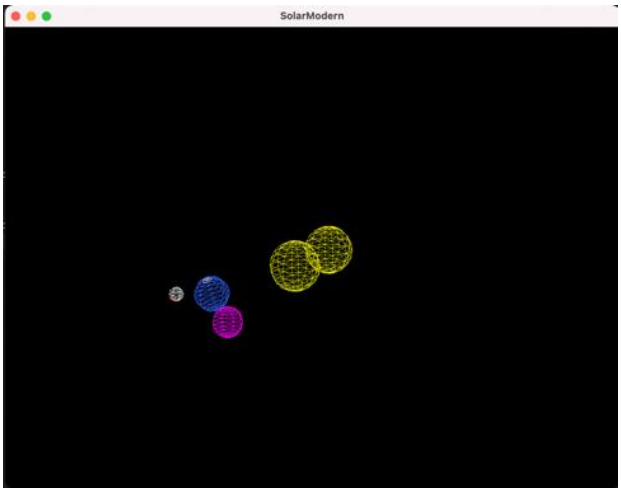
Visualizing normals under low resolution

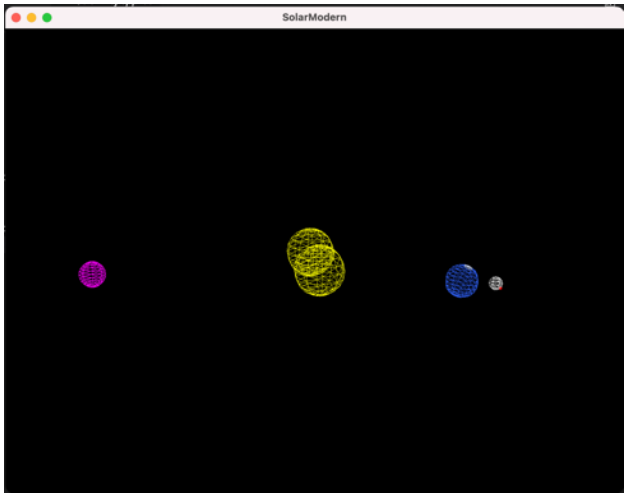
Visualizing normals under high resolution

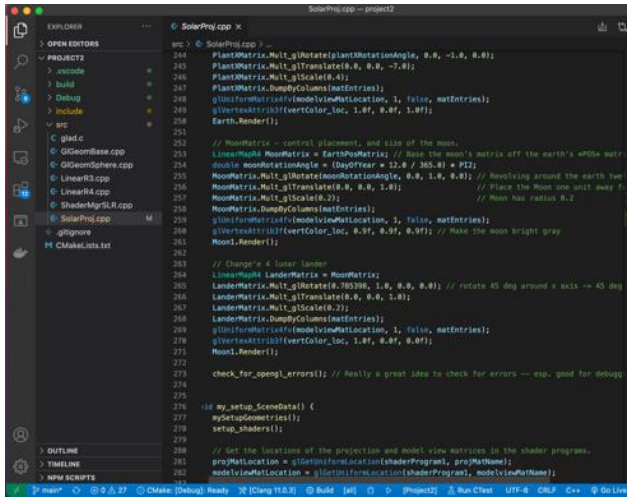
Solar System (C++ with OpenGL)

- Learn more about how to use OpenGL, interrupt-driven programming, animation, and transformations. Program some additions to an animated solar system. Use OpenGL commands to generate transformations that control the animation.

Scene explanation:
Binary Sun in the middle (Yellow spheres): The two suns revolve around the center of the solar system.
Earth (Blue sphere): revolve around the center of the solar system
Moon (White sphere): revolve around the Earth
Planet X (Magenta sphere): orbit the Sun once every 600 days (slower than the Earth)
Change'e 4 Lunar Lander (Red Dot on the moon): a fixed location on the moon



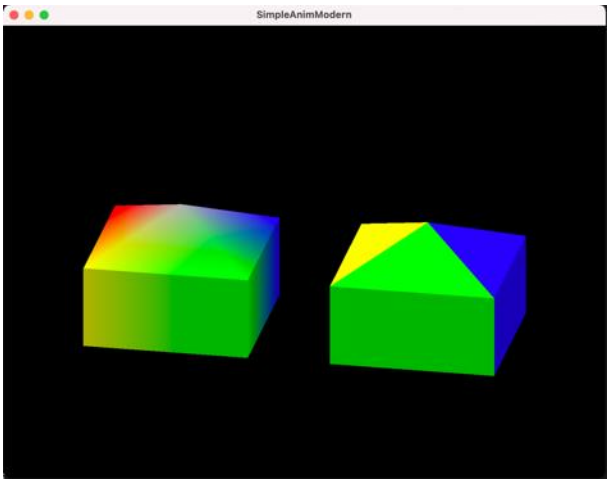




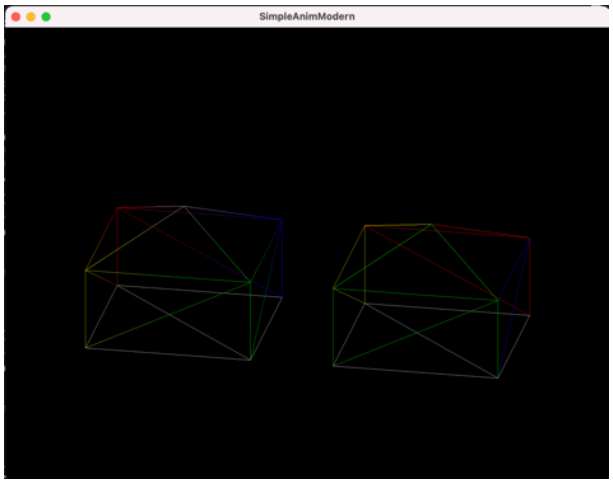
Shaded Tent (C++ with OpenGL)

- Rendering a tent by using 3 different *glDrawArrays* methods, GL_TRIANGLE_FAN, GL_TRIANGLE_STRIP, and GL_TRIANGLES
- Learn how to make triangles of solid color as well as how to shade colors smoothly, how to use key controls to control viewpoint, and toggle wireframe and toggle culling of back faces.

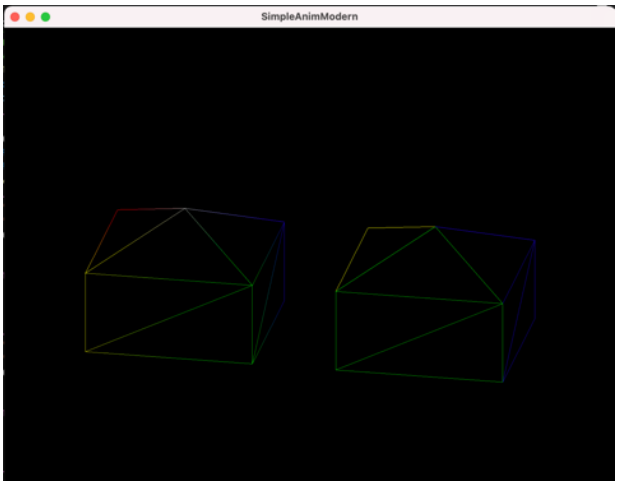
Key press events:
'w': toggle wireframe mode
'c': toggle backface culling



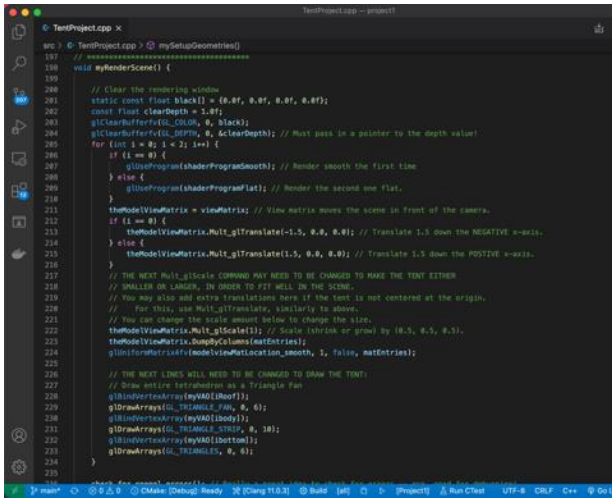
Result Tent



In wireframe mode



Culling backface



```
// ========================================
void myRenderScene() {

    // Clear the rendering window
    static const float black[] = {0.0f, 0.0f, 0.0f, 0.0f};
    const float clearDepth = 1.0f;
    glClearBufferfv(GL_COLOR, 0, black);
    glClearBufferfv(GL_DEPTH, 0, &clearDepth); // Must pass in a pointer to the depth value!
    for (int i = 0; i < 2; i++) {
        if (i == 0) {
            glUseProgram(shaderProgramSmooth); // Render smooth the first time
        } else {
            glUseProgram(shaderProgramFlat); // Render the second one flat.
        }
        theModelViewMatrix = viewMatrix; // View matrix moves the scene in front of the camera.
        if (i == 0) {
            theModelViewMatrix.Mult_glTranslate(-1.5, 0.0, 0.0); // Translate 1.5 down the NEGATIVE x-axis.
        } else {
            theModelViewMatrix.Mult_glTranslate(1.5, 0.0, 0.0); // Translate 1.5 down the POSTIVE x-axis.
        }
        // THE NEXT Mult_glScale COMMAND MAY NEED TO BE CHANGED TO MAKE THE TENT EITHER
        // SMALLER OR LARGER, IN ORDER TO FIT WELL IN THE SCENE.
        // You may also add extra translations here if the tent is not centered at the origin.
        //   for this, use Mult_glTranslate, similarly to above.
        // You can change the scale amount below to change the size.
        theModelViewMatrix.Mult_glScale(1); // Scale (shrink or grow) by (0.5, 0.5, 0.5).
        theModelViewMatrix.DumpByColumns(matEntries);
        glUniformMatrix4fv(modelviewMatLocation_smooth, 1, false, matEntries);

        // THE NEXT LINES WILL NEED TO BE CHANGED TO DRAW THE TENT:
        // Draw entire tetrahedron as a Triangle Fan
        glBindVertexArray(myVAO[iRoof]);
        glDrawArrays(GL_TRIANGLE_FAN, 0, 6);
        glBindVertexArray(myVAO[ibody]);
        glDrawArrays(GL_TRIANGLE_STRIP, 0, 10);
        glBindVertexArray(myVAO[ibottom]);
        glDrawArrays(GL_TRIANGLES, 0, 6);
    }
```

Partial code of rendering scene

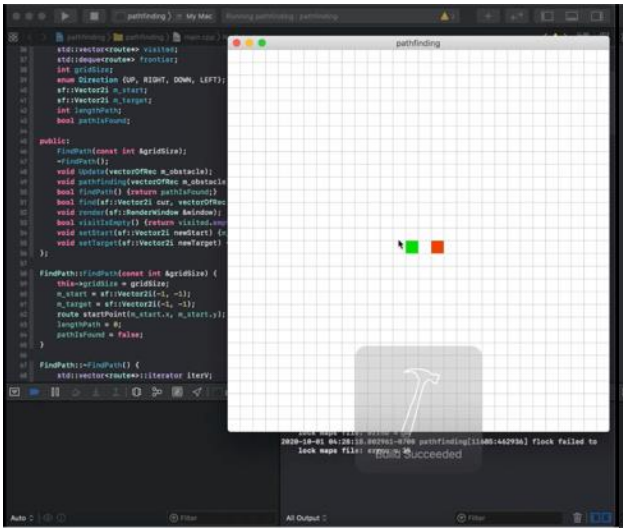
Early Projects (Focus on Game development and algorithm visualization)

Pathfinding Visualizer (C++ with SFML framework)

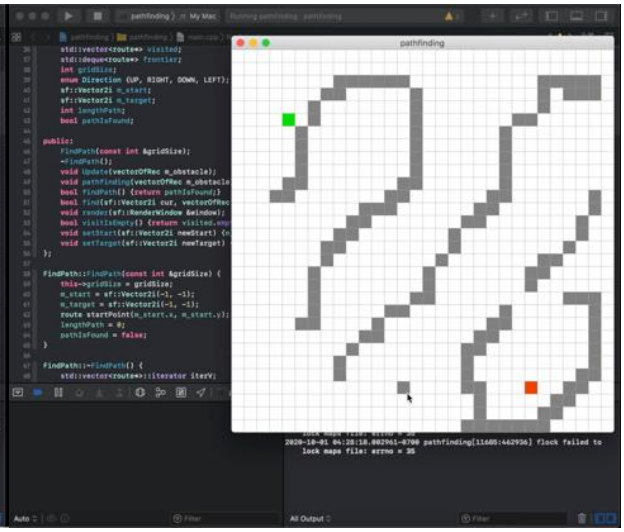
- BFS algorithm

How to play:

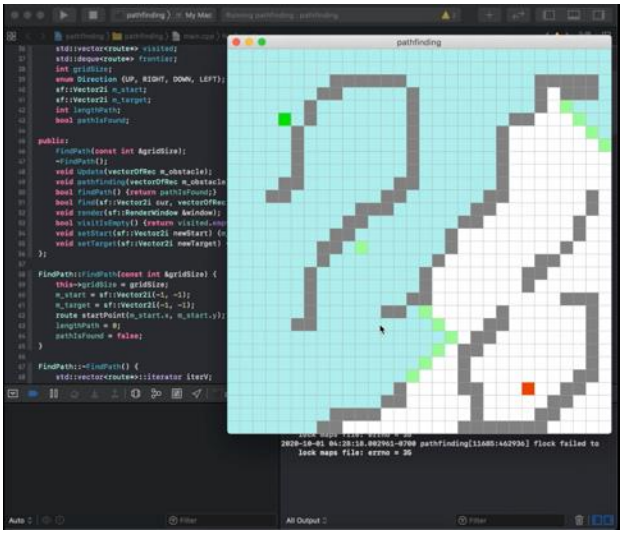
- Drag the green grid and red grid around the board to set the position of the starting point and the target.
- Drawing obstacles
  - Click anywhere on an empty square (white square) to create an obstacle on the board.
  - Hold the mouse while drawing obstacles to create continuous obstacles.
- Removing obstacles
  - Click an obstacle (gray square) again to remove an obstacle on the board.
  - Users can hold the mouse while removing obstacles to clear quickly.
- Click Enter to run the program to show the path between starting point and the target.



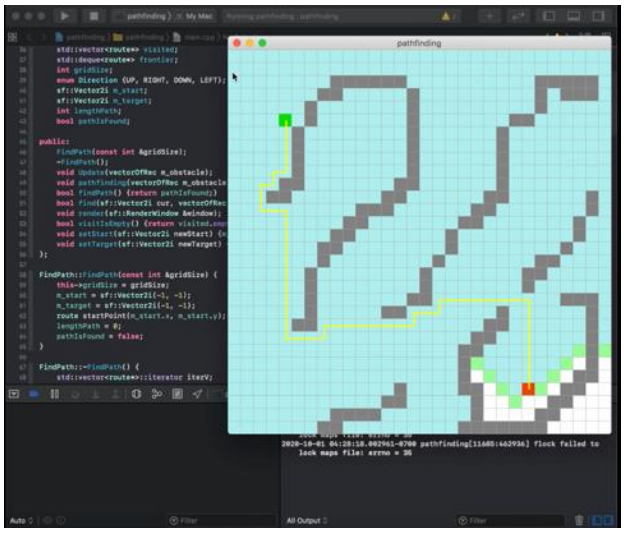
Start Board



Set Obstacles



Start finding path



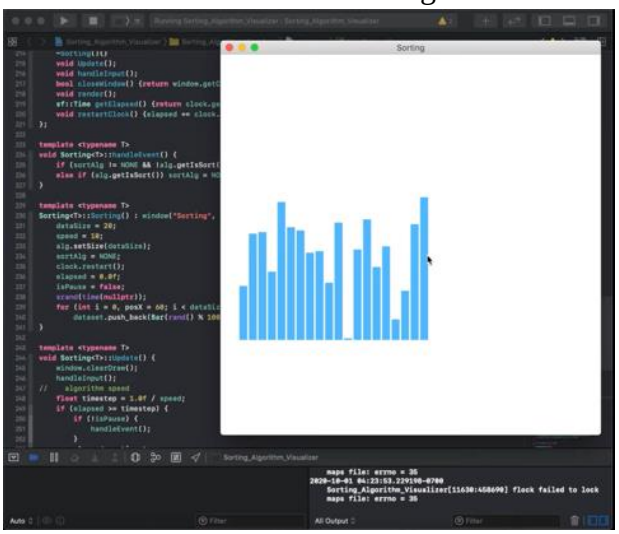
Find a shortest path

Sorting Visualizer (C++ with SFML framework)

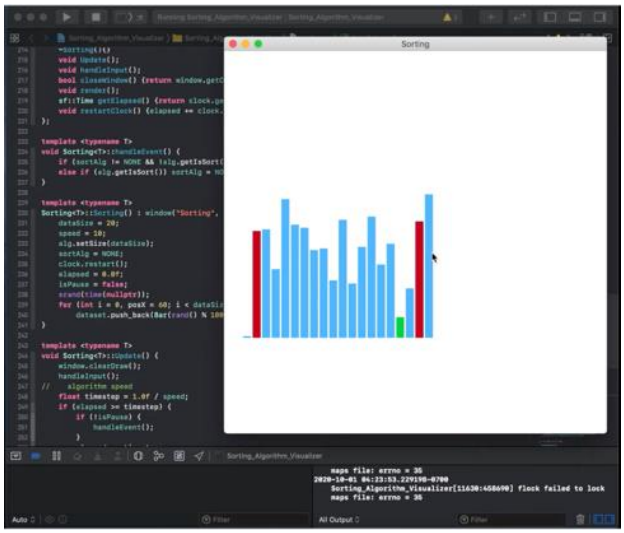
- The following figures are based on selection sort
- Left red bar represents the current index
- Right red bar represents the iterator (loop from current index to the last index to find i^{th} smallest value in the dataset)
- Green bar represents the current i^{th} smallest value

How to play:

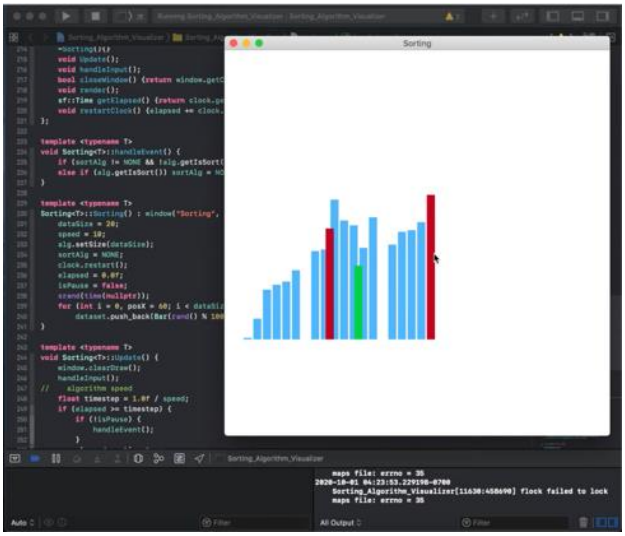
- Press Enter to start sorting



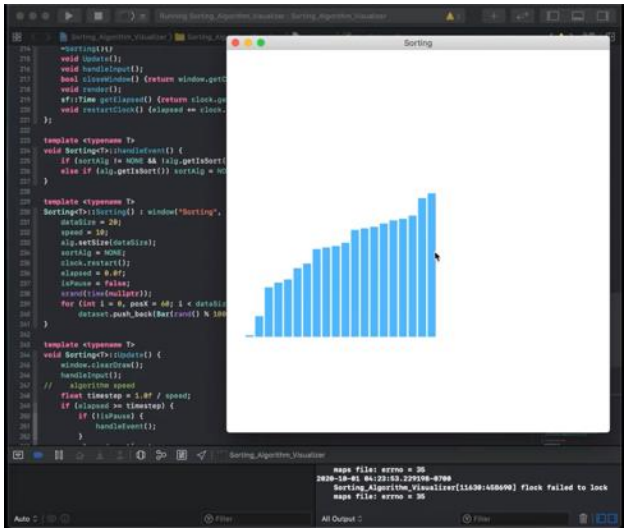
Start Random Dataset



Start sorting



Swapping current index with smallest value



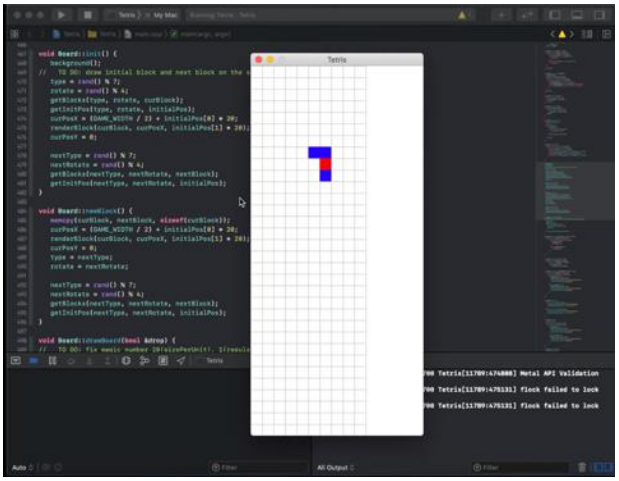
Done sorting

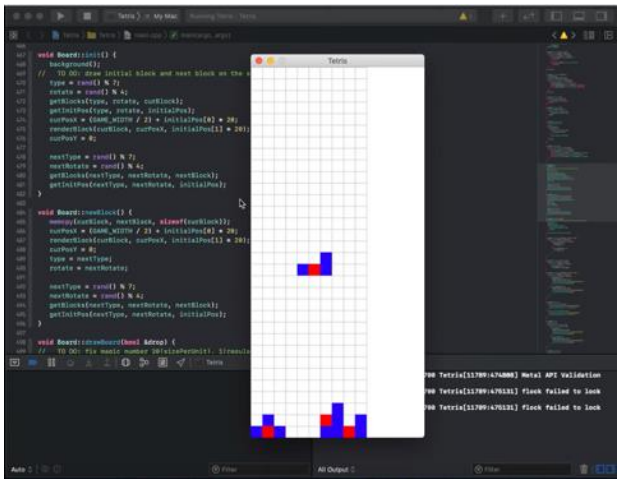
Tetris (C++ with SDL framework)

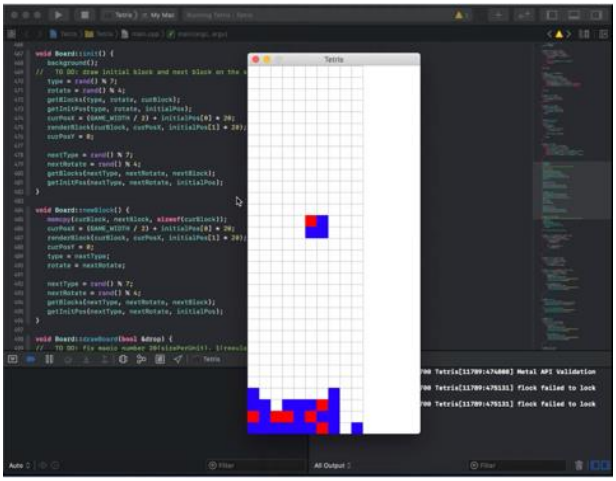
- Classic Tetris game made by C++ and SDL framework.
- The red grid represents the pivot in each block

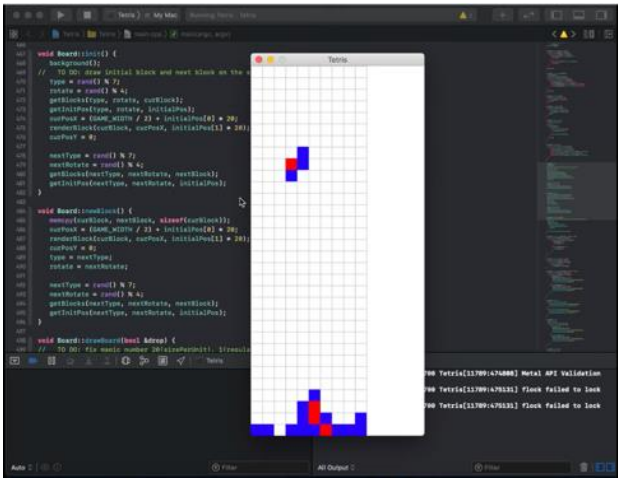
How to play:

- Use the left and right arrow keys to move laterally a falling block.
- Use the down arrow key to accelerate the falling speed.
- Use the z key to rotate a block left and x key to rotate right.







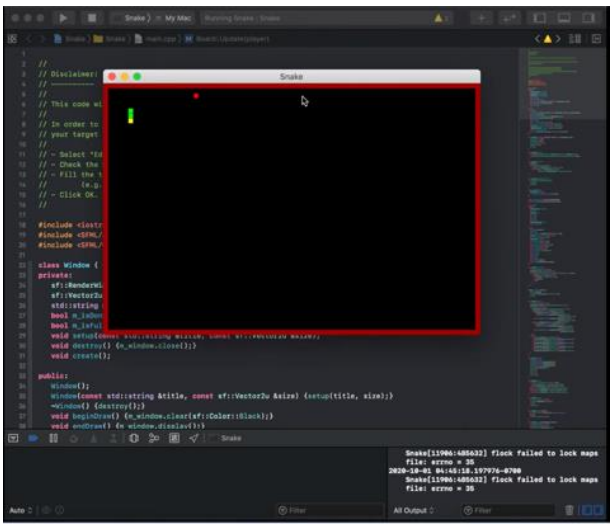


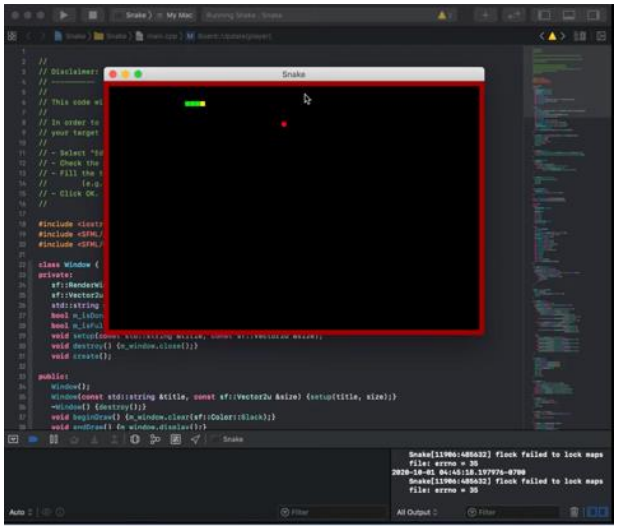
Snake (C++ with SFML framework)

- A classic snake game made by C++ and SFML framework

How to play:

- Use arrow keys to move the snake within the board.
- Be careful not to hit the red barriers.
- Try to eat as many as apple (red dot) on the board to extend the body.





Breakout (C++ with SDL framework)

- Breakout game made by C++ and SFML framework.

How to play:

- Use the left arrow key and right arrow key to move the paddle to ricochet the ball.
- After knocking down all the bricks in the game, the player wins