

I. Computer Vision

- A. CNN, Adversarial Attack, Deep Learning on 3D Data, Camera Model
- B. PCA, NN, CNN (Python with numpy, pytorch, matplotlib, pandas, sklearn)
- C. Object Tracking (OpenCV with C++)

II. AI: Search and Reasoning

- A. Gomoku AI (Python with pygame)
- B. Blackjack AI (Python with pygame)
- C. 2048 AI (Python with pygame)
- D. Grid World (Python with pygame)

III. 2020 Summer Internship – Game Development

- A. Zombie Runner Game (C# with Unity)
- B. Zombie Downstairs Game (C# with Unity)

IV. Web Scrapper

- A. LinkedIn Web Scrap (Python, Selenium)
- B. 104 Web Scrap (Python, Selenium)

V. Personal Website

VI. Computer Graphic

- A. Texture Maps (C++ with OpenGL)
- B. Phong Lighting (C++ with OpenGL)
- C. Surface of Rotation and Normals (C++ with OpenGL)
- D. Solar System (C++ with OpenGL)
- E. Shaded Tent (C++ with OpenGL)

VII. Early Projects (Game development and algorithm visualization)

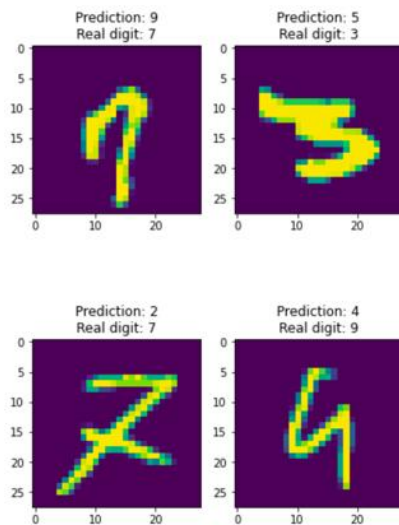
- A. Pathfinding Visualizer (C++ with SFML framework)
- B. Sorting Visualizer (C++ with SFML framework)
- C. Tetris (C++ with SDL framework)
- D. Snake (C++ with SFML framework)
- E. Breakout (C++ with SDL framework)

Computer Vision

[CNN, Adversarial Attack, Deep Learning on 3D Data, Camera Model and Rigid Transformations](#)

Works I have done in this project:

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



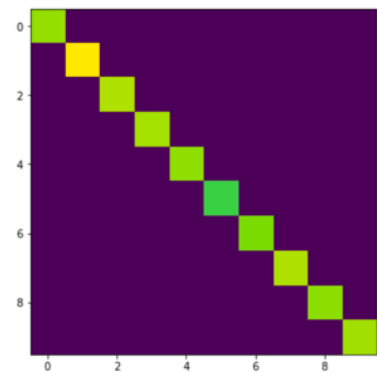
Plot digits that the network got wrong

```
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__()

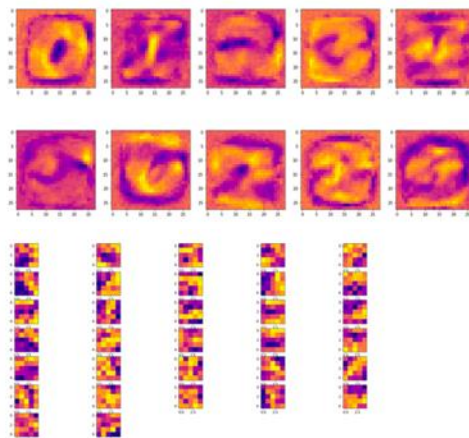
        ## YOUR CODE HERE
        self.layer1 = nn.Conv2d(1, 32, kernel_size=5, padding=1)
        self.layer2 = nn.Conv2d(32, 64, kernel_size=5)
        self.flat = nn.Flatten()
        self.fcl = nn.Linear(1024, 10)
        ## END OF CODE

    def forward(self, x):
        """
        forward function describes how input tensor is transformed to output,
        --tensor
        Args:
            x: (N,1,28,28) 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.fcl(x)
        ## END OF CODE
        return x
```

CNN architecture



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'] = []

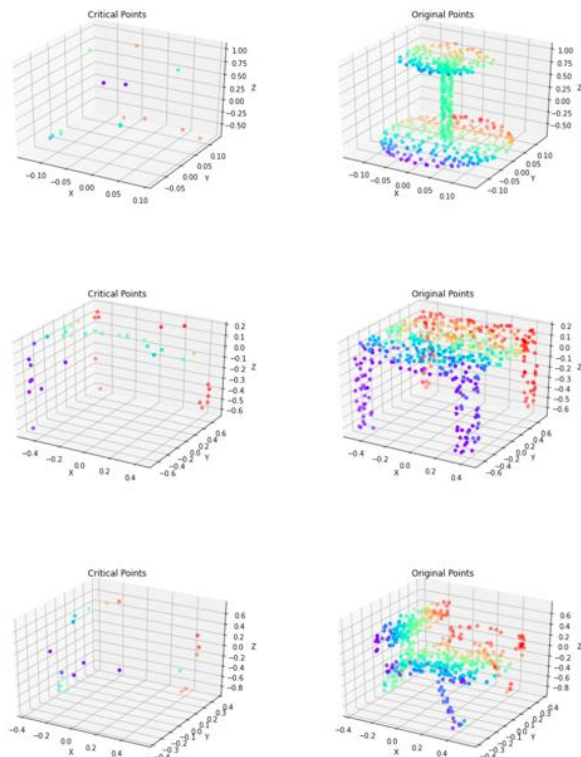
        # run sample through the network

    ### START CODE HERE ###
    data.requires_grad = True
    model.zero_grad()
    output = PtNet.forward(data)
    # output.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

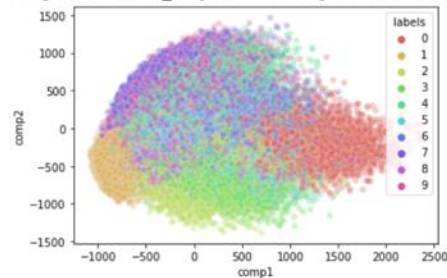
PCA, NN, CNN (Python with numpy, pytorch, matplotlib, pandas, sklearn)

Works I have done in this project:

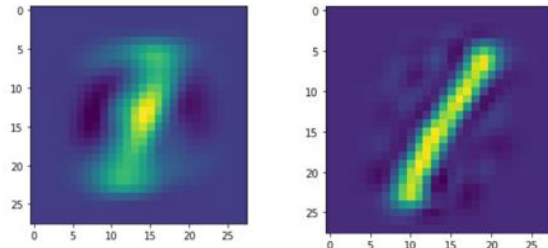
- Preprocess dataset with Principal Component Analysis (PCA)
 - o Perform PCA using 2 methods: singular value decomposition and eigen decomposition
 - o Project data to different dimensions to see how well the digits can be represented after PCA
- Neural Network for regression
 - o Build a basic MultiLayer Perceptron (MLP) for fitting a line to random data
 - o Decide on the right optimizer, choose an appropriate loss function for my learning task, and train the model over the given data
- Corner Detection using Convolution Neural Network (CNN)
 - o The CNN will output the coordinates of the corner in the given image with a corner in it
 - o Build convolution and maxpool layers, and 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
```

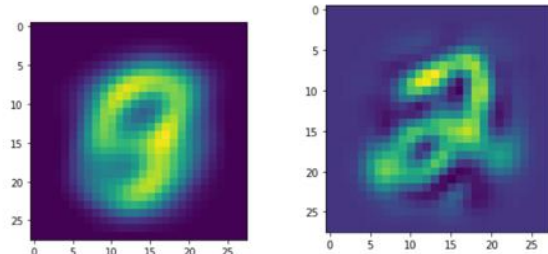
<matplotlib.axes._subplots.AxesSubplot at 0x7faef9eef98>



Data distribution in space spanned by 2 principal components



Data in 2 dimensions



Data in 50 dimensions

```

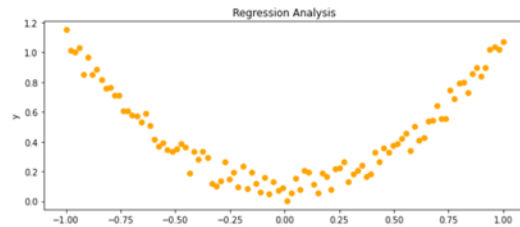
torch.manual_seed(1) # reproducible

x = torch.unsqueeze(torch.linspace(-1, 1, 100), dim=1) # x data (tensor), shape=(100, 1)
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()

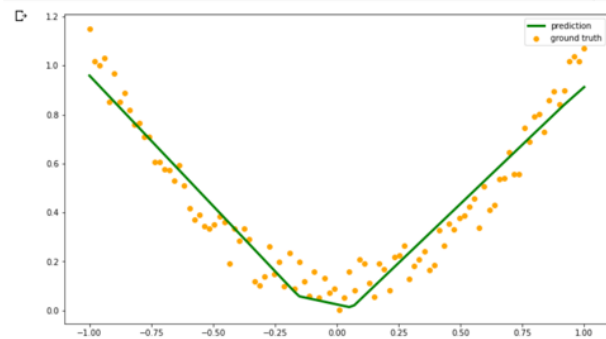
```



```

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

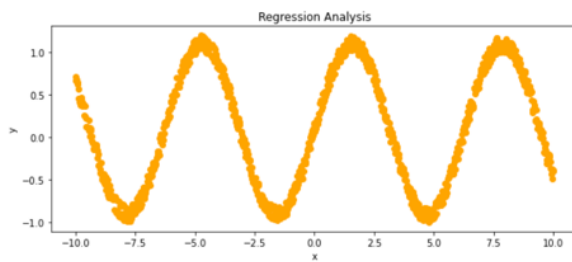
```

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

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()

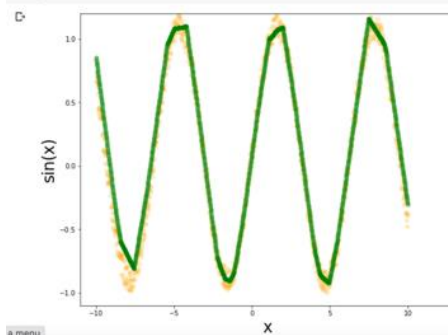
```



```

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, 11.0)
ax.set_ylim(-1.2, 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

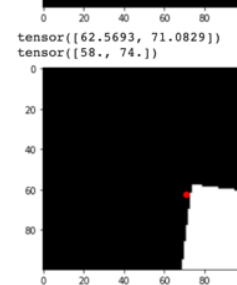
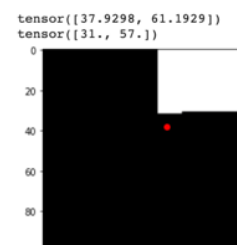
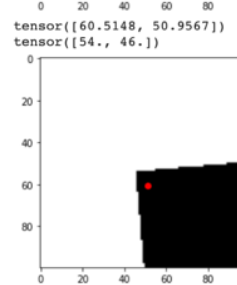
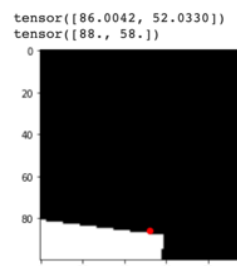
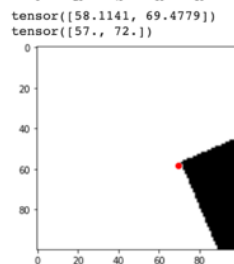
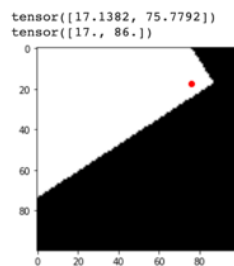
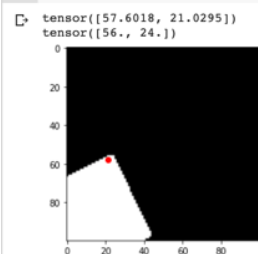
```

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=5)
            plt.show()

        break

```



Corner Detection

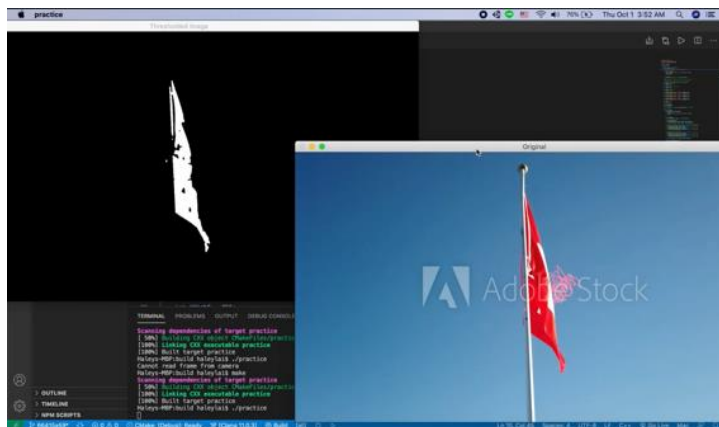
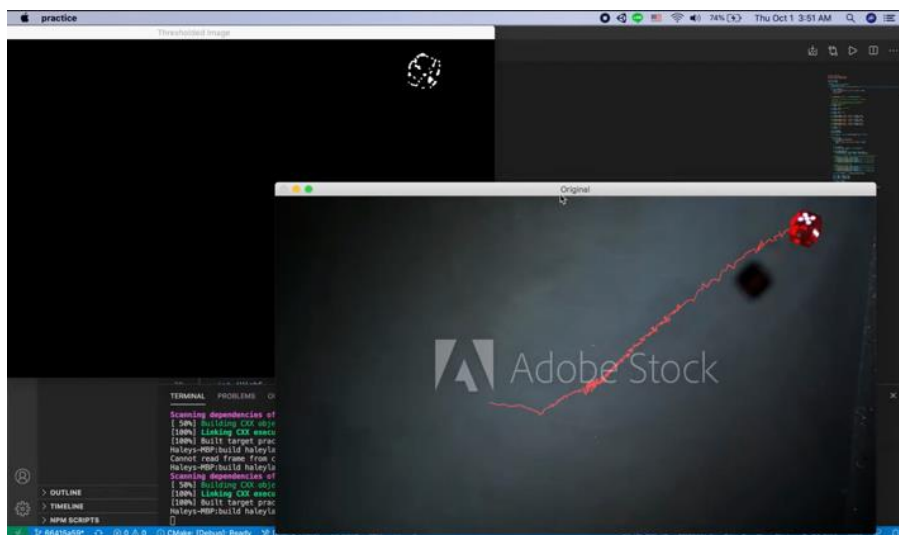
Object Tracking (OpenCV with C++)

- A simple object tracking using color and shape

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

What does the program do:

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

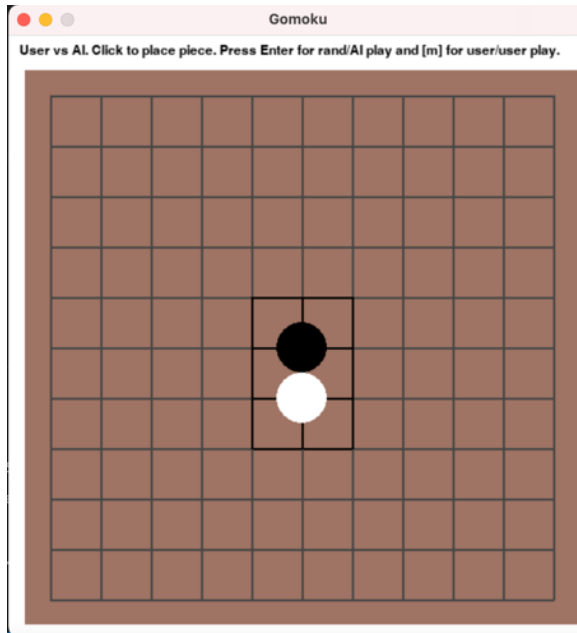


```
40 cv::Mat imgLines = cv::Mat::zeros(imgtemp.size(), CV_8UC3);
41
42 while (true) {
43     cv::Mat image;
44     bool success = cap.read(image);
45     if (!success) {
46         cout << "Cannot read frame from camera" << endl;
47         break;
48     }
49
50     cv::Mat imgHSV;
51     cv::cvtColor(image, imgHSV, cv::COLOR_BGR2HSV);
52
53     cv::Mat imgThresholded;
54     cv::inRange(imgHSV, cv::Scalar(0, 0, 0), cv::Scalar(180, 255, 255),
55                imgThresholded);
56
57     // eliminate close objects which have the same color as our main object
58     // Morphological opening: apply erosion, followed by the dilation
59     cv::erode(imgThresholded, imgThresholded,
60              cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(5, 5)));
61     cv::dilate(imgThresholded, imgThresholded,
62              cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(5, 5)));
63
64     // eliminate noises in the image
65     // Morphological closing: apply dilation, followed by the erosion
66     cv::dilate(imgThresholded, imgThresholded,
67              cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(5, 5)));
68     cv::erode(imgThresholded, imgThresholded,
69              cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(5, 5)));
70
71     cv::Moments oMoments = cv::Moments(imgThresholded);
72 }
```

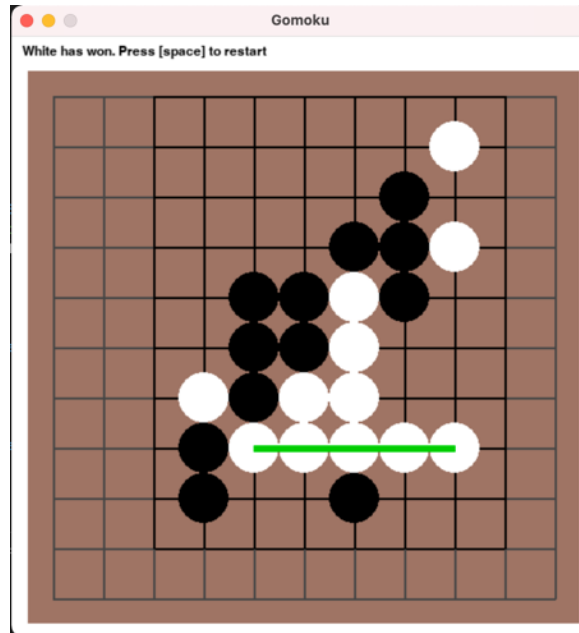
AI: Search and Reasoning

Gomoku AI (Python with pygame)

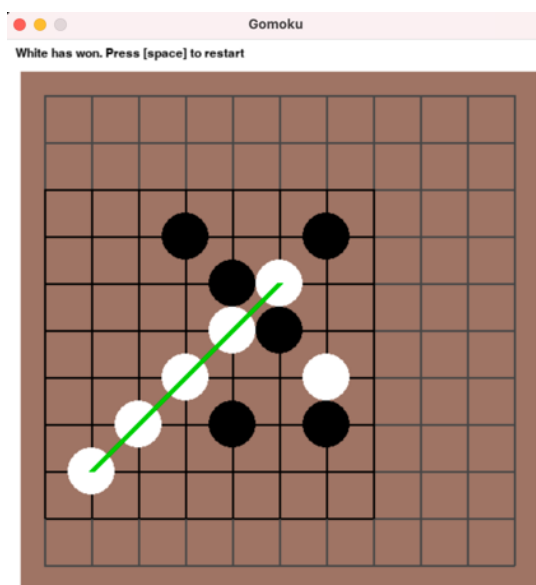
- Implement Monte Carlo Tree Search (MCTS) for playing Gomoku



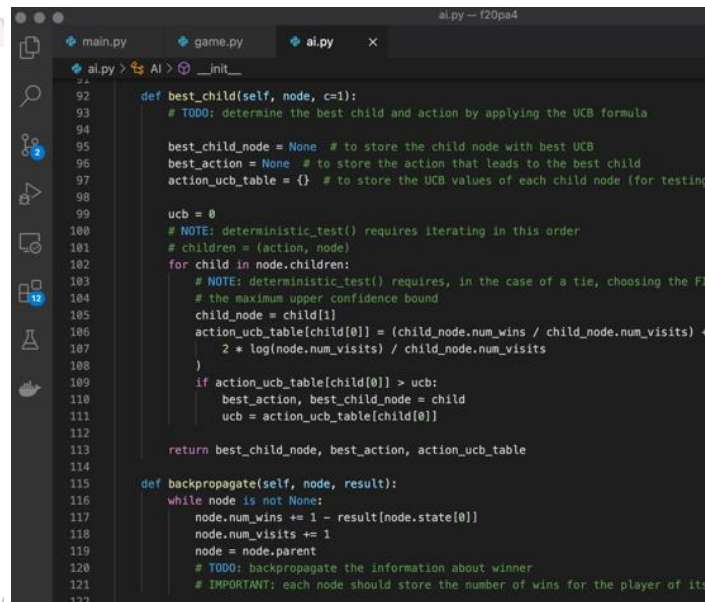
Start Board



Player vs AI (AI won)



AI vs AI (White won)



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 for Blackjack

Description for each Evaluation:

Monte Carlo Policy Evaluation

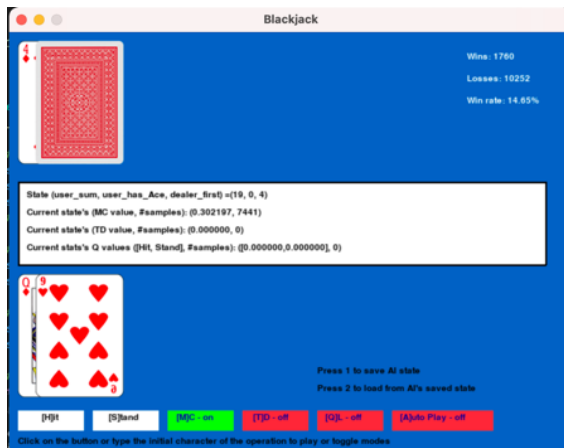
Evaluate the policy "Hit (ask for a new card) if sum of cards is below 14, and Stand (switch to dealer) otherwise" using the Monte Carlo method. Namely, learn the values for each state under the policy.

Temporal-Difference Policy Evaluation

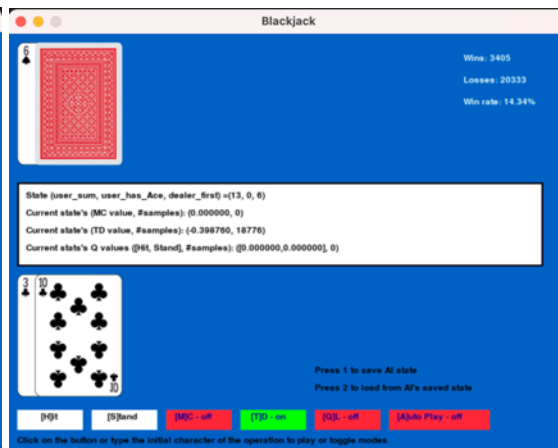
Evaluate the policy "Hit (ask for a new card) if sum of cards is below 14, and Stand (switch to dealer) otherwise" using the Temporal-Difference method.

Q-Learning

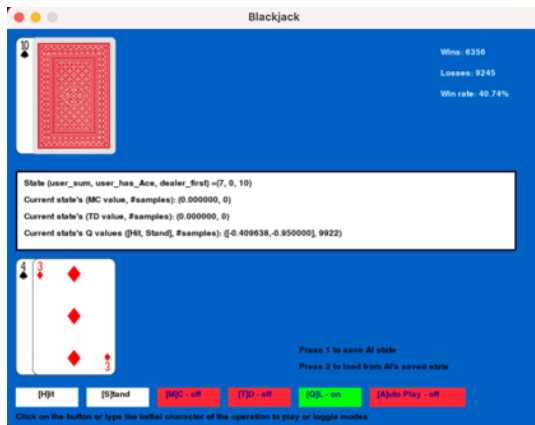
Implement the Q-learning algorithm. Use epsilon=0.4 in the epsilon-strategy in your final submission, but you are encouraged to check the behavior difference for various choices of epsilon. After learning, AutoPlay will follow the Q-learning values to make decisions.



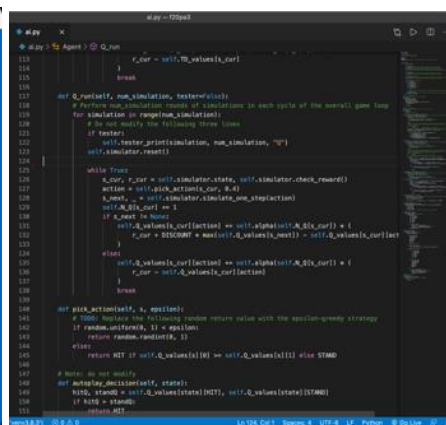
Screenshot of game running under Monte Carlo Policy



Game running under Temporal-Difference Policy



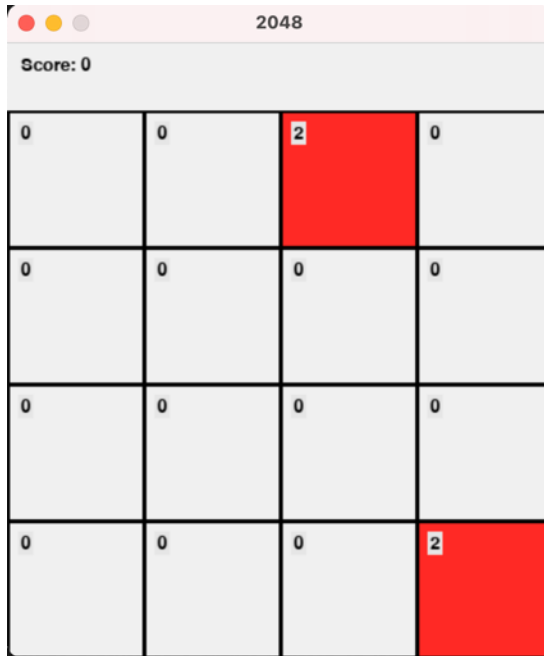
Game running under Q-learning algorithm



Partial code of Q-learning algorithm

2048 AI (Python with pygame)

- Implement a game AI for the 2048 game based on expectimax search



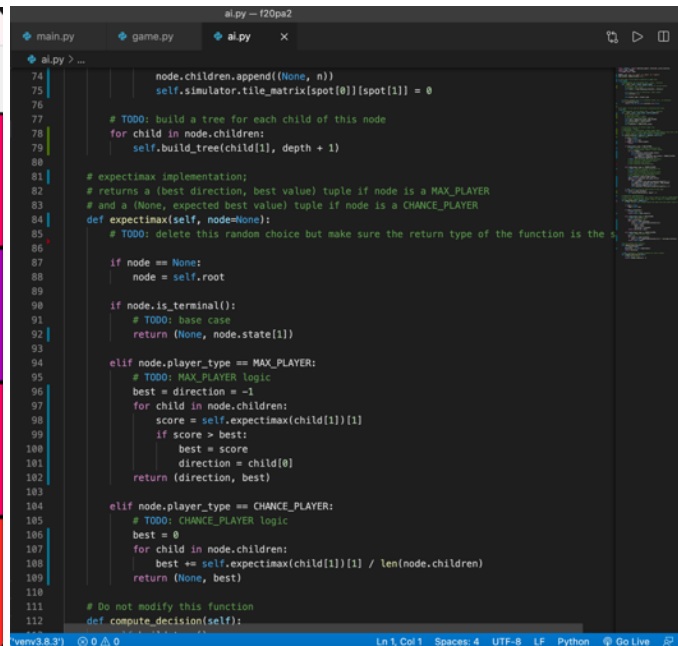
Start board



Middle game



End game



Implementation code of expectimax algorithm

Grid World (Python with pygame)

- Find paths from the start (yellow node) to the goal (orange node)

The program implements the following search 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



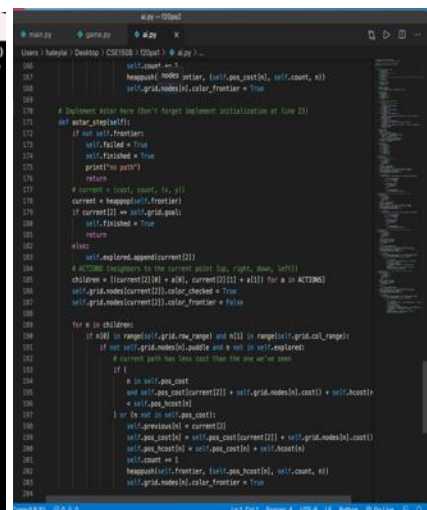
BFS mode



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

LinkedIn Web Scrap (Python, Selenium)

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

The image is a composite of three screenshots from a Mac desktop, illustrating a web scraping project.

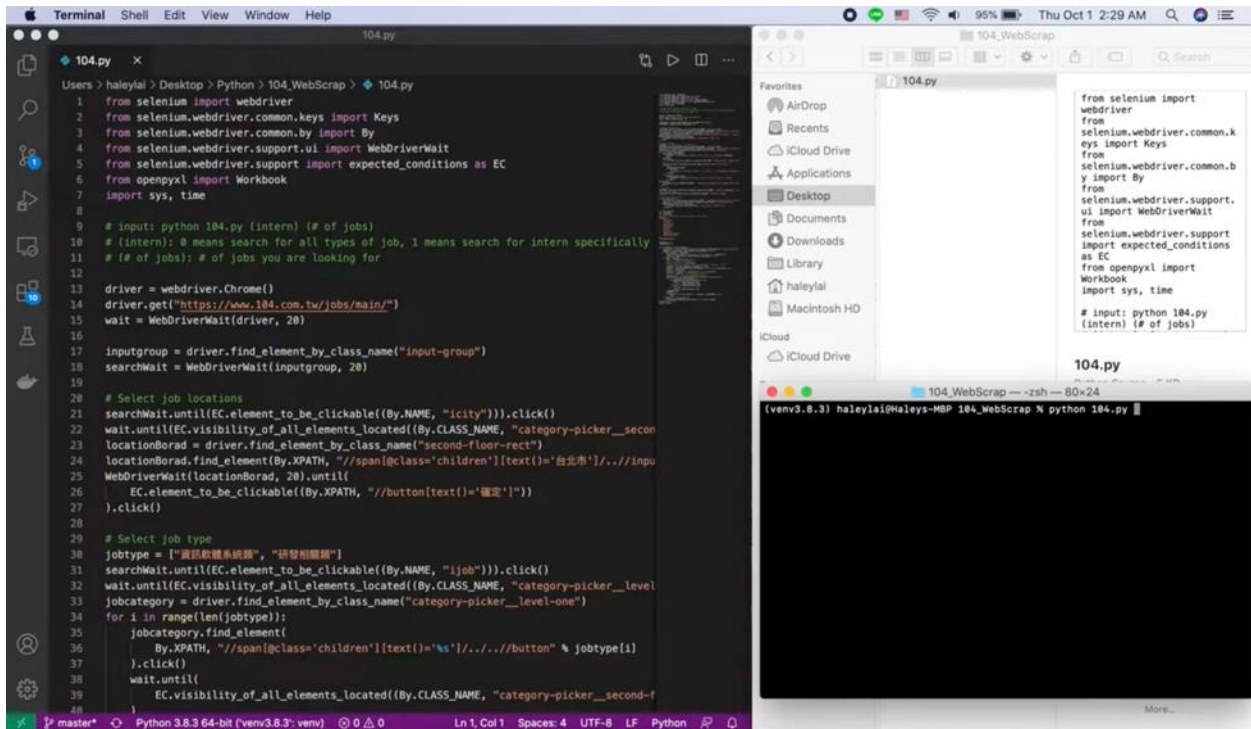
Top Left: Terminal Window
 The terminal shows the execution of a Python script named `Simple_Linkedin_WebScrap.py`. The code defines a `visibility_of_text` function and a `len_of_container` function, both using Selenium WebDriver to interact with the LinkedIn website. The script is run in a virtual environment (venv) with Python 3.8.3. The output shows the script is running successfully.

Top Right: Browser Window
 The browser window shows the LinkedIn website. The page title is "LinkedIn.py". The page content is a list of job titles and company details, which is the output of the web scraper. The page is displayed in a dark theme.

Bottom: Excel Spreadsheet
 The Excel spreadsheet is titled "Job Title". It contains a list of job titles and company details. The columns are labeled "Job Title", "Company", "Location", and "Content". The data is organized into rows, with each row representing a job listing. The spreadsheet is displayed in a light theme.

104 Web Scrap (Python, Selenium)

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



ExcelFileEditViewInsertFormatToolsDataReviewViewTell Me

AutoSave104%

HomeInsertDrawPage LayoutFormulasDataReviewViewTell Me

Calibri (Body)11A A'

B I U L

Wrap Text

General

Conditional FormattingFormat as TableCell Styles

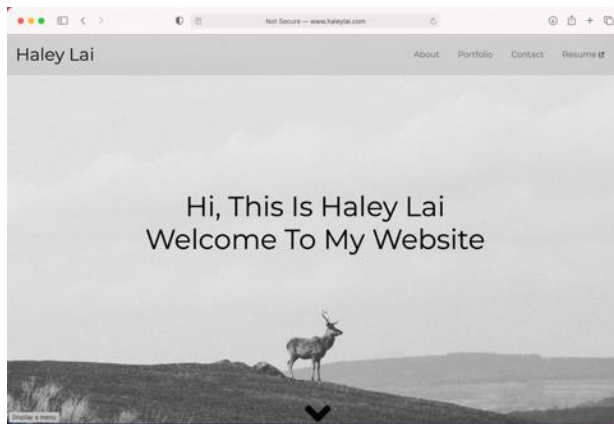
Insert >Delete >Format >Sort & Filter >Find & Select >Ideas >Sensitivity

A1	Job Title																		
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q		
	Job Title	Company	Location	Experience	Education	Content	Salary	Employees Applying Additional Info											
1	「新服務利器」初級Linux系統工程師	紅石資訊有限公司	台北市內湖區	經驗不限	專科	1.完成伺服器及應用系統的安裝 月薪\$0,000元以上	N/A	6-10人應徵	N/A										
2	資深Java工程師	緯創資通股份有限公司台灣分公司	台北市內湖區	5年以上	專科	1.負責軟體技術問題, 解決度: 月薪\$8,000~\$120,000元	N/A	0~5人應徵	N/A										
3	資深軟體工程師 月薪\$60,000-\$80,000**	利通資訊科技股份有限公司	台北市中正區	2年以上	學歷不拘	1.與設計團隊共同合作開發大型 月薪\$60,000~\$80,000元	員工24人, 6~10人應徵	N/A											
4	「新服務利器」Linux系統工程師	紅石資訊有限公司	台北市內湖區	經驗不限	專科	1. 協助處理應用系統的故障, 月薪\$0,000元以上	N/A	0~5人應徵	N/A										
5	CAE工程師 (台北總公司)	瑞品股份有限公司	台北市信義區	2年以上	大學	1.NET Web應用系統開發、維護、持續維護	員工110人, 0~5人應徵	N/A											
6	高級網路工程師 月薪\$60,000-\$80,000**	利通資訊科技股份有限公司	台北市中正區	2年以上	學歷不拘	1.與設計團隊共同合作開發大型 月薪\$60,000~\$80,000元	員工24人, 6~10人應徵	N/A											
7	IT engineer 硬體工程師	聯發科技安訊訊安訊訊有限公司	台北市內湖區	3年以上	大學	崗位描述: 1.專案開發設計階段 持續維護	N/A	11~30人應徵	N/A										
8	Linux工程師 1. 需求分析與系統架構	聯發科技股份有限公司	台北市內湖區	經驗不限	大學	需求分析、系統架構與硬體平台 月薪\$80,000元以上	員工225人, 0~5人應徵	N/A											
9	資深Java軟體工程師/Senior Java developer	奇智維科技股份有限公司	台北市內湖區	5年以上	大學	主要負責: 網路設備安裝以及 月薪\$60,000~\$120,000元	員工10人, 0~5人應徵	N/A											
10	台北JSP軟體工程師	金上智科技股份有限公司	台北市信義區	3年以上	大學	1.熟悉 Easyflow 表單開發設計及 持續維護	員工900人, 0~5人應徵	N/A											
11	軟體工程師	威訊電信股份有限公司	台北市大安區	經驗不拘	大學	通訊軟體工程師 1.與系統分析師 月薪\$45,000~\$65,000元	N/A	11~30人應徵	N/A										
12	軟體工程師	弘華電子股份有限公司	台北市內湖區	經驗不拘	大學	工作內容: 作為產品工程設計師 持續維護	員工390人, 0~5人應徵	N/A											
13	軟體工程師	富士多科技股份有限公司	台北市中山區	1年以上	大學	維護軟體工程師 需求分析軟體 月薪\$70,000~\$85,000元	員工25人, 0~5人應徵	N/A											
14	Java工程師	Trendronet Co., LTD. 聯群股份有限公司	台北市中山區	經驗不拘	高中	維護工程共同協議溝通開發度 月薪\$7,000~\$70,000元	員工10人, 6~10人應徵	N/A											
15	Android軟體工程師	Trendronet Co., LTD. 聯群股份有限公司	台北市中山區	經驗不拘	高中	維護工程共同協議溝通開發度 月薪\$7,000~\$70,000元	員工10人, 0~5人應徵	N/A											
16	Java工程師	立方支付有限公司	台北市中山區	經驗不拘	大學	Java軟體工程師 工作內容【工作 持續維護】	員工32人, 11~30人應徵	N/A											
17	Java工程師	雙子數位科技股份有限公司	台北市內湖區	3年以上	專科	1.負責Java系統開發與設計 2. 網頁持續維護	N/A	0~5人應徵	N/A										
18	PHP工程師	聯群股份有限公司	台北市中山區	經驗不拘	大學	網路系統開發與維護 需求分析 月薪\$60,000~\$80,000元	N/A	0~5人應徵	N/A										
19	軟體工程師	聯群股份有限公司	台北市中山區	經驗不拘	高中	維護軟體工程師 需求分析軟體 月薪\$70,000~\$85,000元	員工10人, 6~10人應徵	N/A											
20	軟體工程師	中聯數位科技股份有限公司	台北市中山區	經驗不拘	高中	需求分析師 是網路網路資訊產品 月薪\$6,000~\$9,000元	員工110人, 6~10人應徵	N/A											
21	.net工程師	雙子數位科技股份有限公司	台北市內湖區	4年以上	專科	1.負責系統架構之分析、設計以及 持續維護	N/A	0~5人應徵	N/A										
22	Android APP 工程師-droids	雙子數位科技股份有限公司	台北市內湖區	3年以上	專科	1.負責Android系統開發 2. 網頁 持續維護	N/A	0~5人應徵	N/A										
23	軟體工程師	Unimax 宇鼎電子股份有限公司 華聯集團	台北市文山区	經驗不拘	大學	負責專用電子相關產品軟體工程 月薪\$40,000元以上	N/A	11~30人應徵	N/A										
24	軟體工程師	得利科技股份有限公司	台北市內湖區	經驗不拘	大學	負責軟體工程師 需求分析軟體 月薪\$22,000~\$45,000元	員工90人, 11~30人應徵	N/A											
25	軟體工程師	香港商德利有限公司台灣分公司	台北市中山區	3年以上	大學	1. 負責ERP EXP BPM AM 等 持續維護	員工400人, 0~5人應徵	N/A											
26	軟體工程師	ACT Genomics Co. Ltd. 丹趣基因生技股份有限公司	台北市內湖區	5年以上	大學	1.開發基因檢測軟體系統、處理 持續維護	員工200人, 0~5人應徵	N/A											
27	軟體工程師	快樂華南教育科技股份有限公司	台北市中山區	1年以上	大學	【工作內容】以C# Android APP 月薪\$40,000元以上	員工20人, 0~5人應徵	N/A											
28	軟體工程師	順祥科技股份有限公司	台北市內湖區	經驗不拘	大學	參與網路行銷與廣告平台設計 月薪\$40,000~\$80,000元	員工50人, 11~30人應徵	N/A											
29	軟體工程師	宏利科技股份有限公司	台北市中山區	經驗不拘	大學	月薪\$60,000~\$80,000元 網路開發 月薪\$40,000~\$60,000元	員工20人, 0~5人應徵	N/A											
30	軟體工程師	宏利科技股份有限公司	台北市內湖區	經驗不拘	碩士	1. 熟悉使用Linux命令以及Linux 月薪\$30,000~\$2,000元	員工20人, 6~10人應徵	N/A											
31	軟體工程師	聯訊國際股份有限公司	台北市內湖區	1年以上	大學	有金銀業、系統整合、資料庫 持續維護	員工100人, 0~5人應徵	N/A											
32	軟體工程師	網際網路股份有限公司	台北市內湖區	經驗不拘	高中	網際網路是台灣金融科技解決方案 月薪\$35,000~\$2,000元	員工110人, 11~30人應徵	N/A											
33	Java工程師	雙子數位科技股份有限公司	台北市內湖區	3年以上	專科	1.負責Java系統開發與設計 2. 網頁持續維護	N/A	0~5人應徵	N/A										
34	軟體工程師	弘華電子股份有限公司	台北市內湖區	2年以上	大學	公司資訊系統開發 2. 網路開發 持續維護	員工1300人, 0~5人應徵	N/A											
35	軟體工程師	七鑫電子(香港)有限公司	台北市內湖區	經驗不拘	高中	1. 負責Linux系統上開發C# 月薪\$30,000~\$5,000元	員工22人, 6~10人應徵	N/A											
36	Frontend Engineer 前段軟體工程師	圖靈科技股份有限公司	台北市松山區	經驗不拘	大學	1. 負責公司Aui產品Web前段開發 持續維護	N/A	0~5人應徵	N/A										
37	軟體工程師	聯群科技股份有限公司	台北市松山區	經驗不拘	大學	1.負責軟體之分析、設計以及 月薪\$6,000~\$70,000元	員工90人, 6~10人應徵	N/A											
38	軟體工程師	達實資訊科技股份有限公司	台北市內湖區	1年以上	專科	1.負責遠端集團網路全球化 月薪\$35,000~\$5,000元	員工206人, 0~5人應徵	N/A											

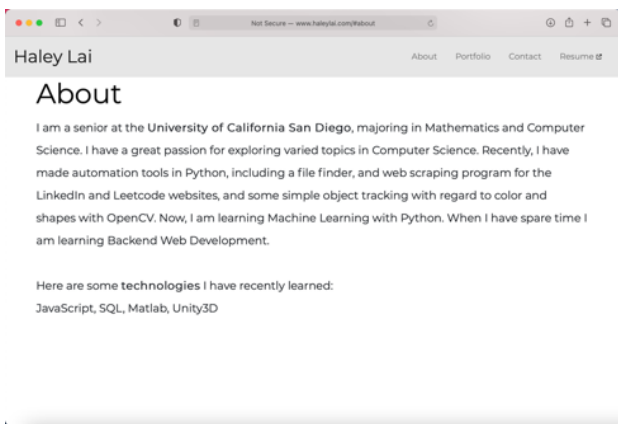
Sheet+

100%

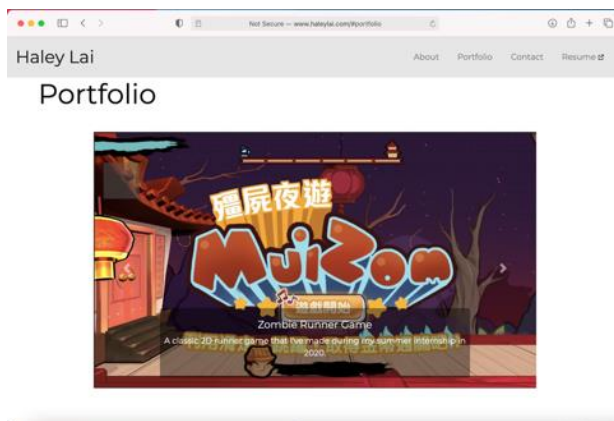
Personal Website (HTML, CSS, JavaScript, jQuery, Node.js, Express.js, EJS, MongoDB)



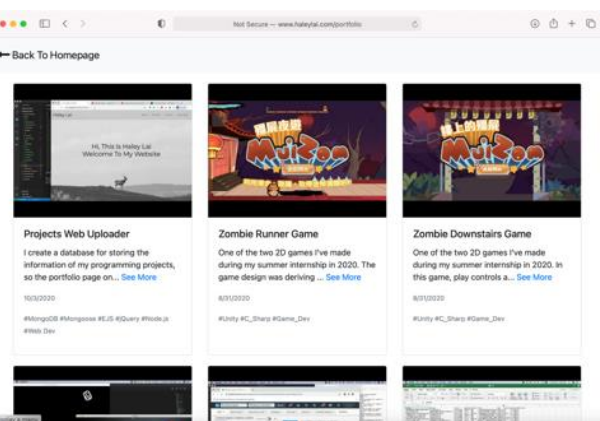
Home Page



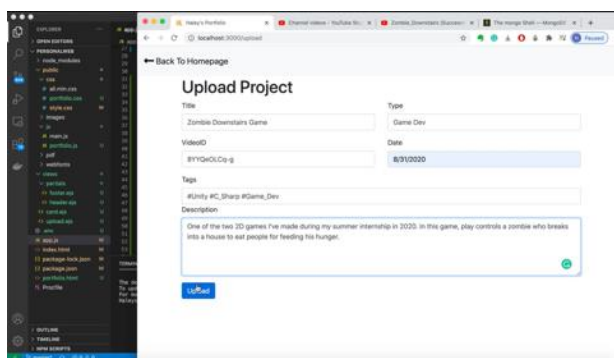
About Page



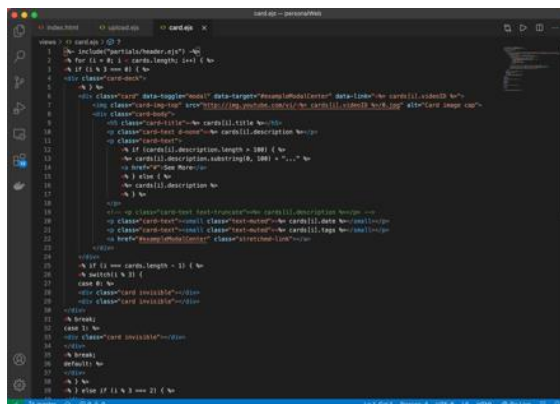
Portfolio Overview



Portfolio Page



Portfolio page uploader



Partial code of project page uploader

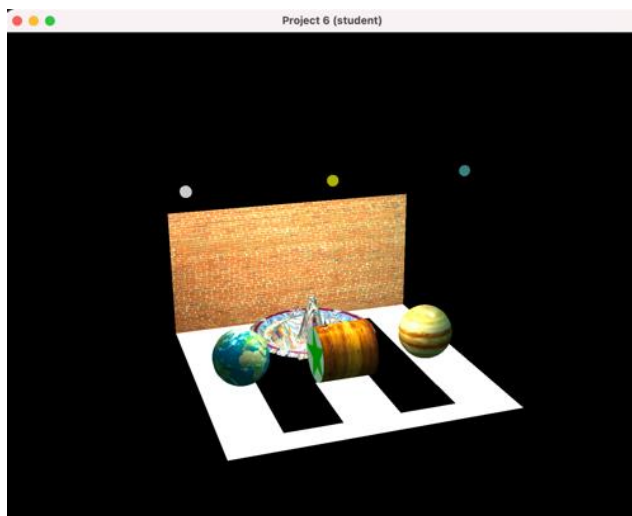
Portfolio page uploader:

A simple page I made for uploading projects to my portfolio page. A project card (refer to Portfolio Page figure) will be created automatically after submitting the form in the upload page.

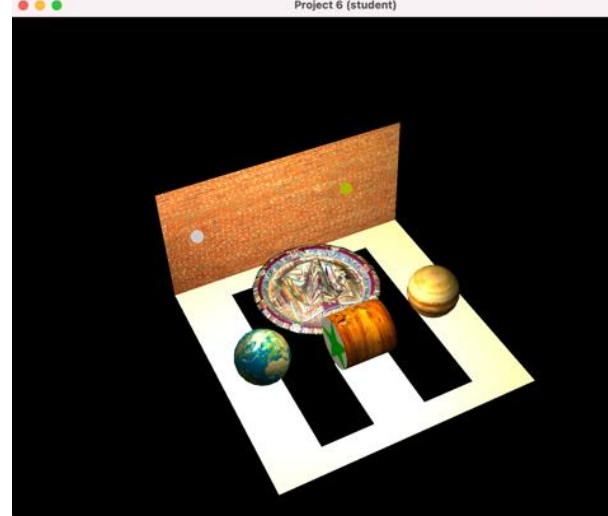
Computer Graphic

Texture Maps (C++ with OpenGL)

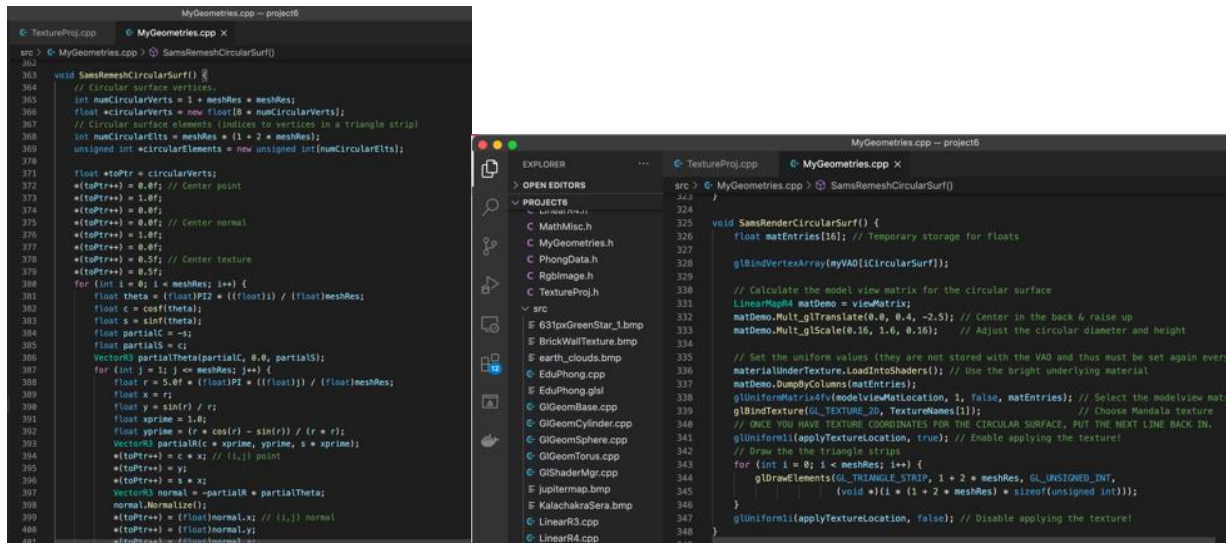
- Add textures to spheres, a rectangle, a cylinder and a surface of rotation.
- Add texture coordinates to the surface of rotation.
- Write a procedural texture for the floor of the scene.



Objects with text



Toggle ambient lighting off



Calculate texture coordinate for circular surface

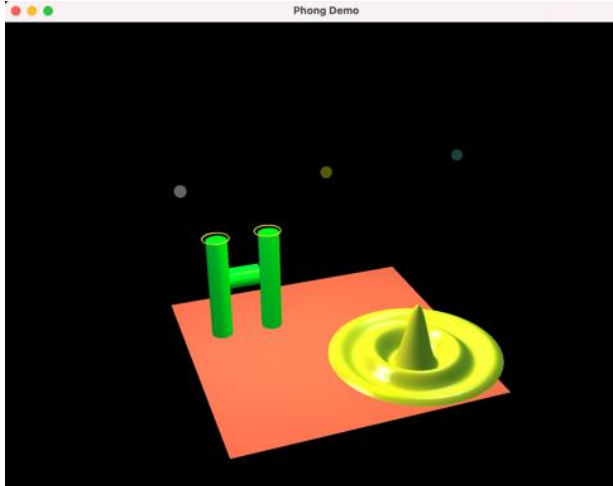
Add texture to circular

Phong Lighting (C++ with OpenGL)

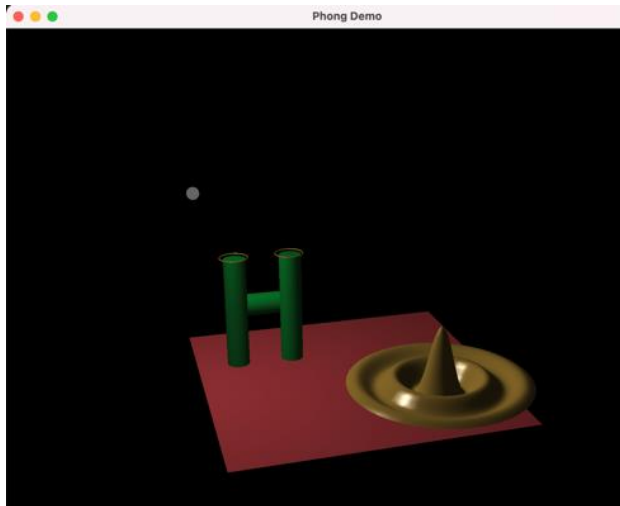
- Use illumination and shading to make scene look more three-dimensional.
- Learn how to shade objects with the Phong lighting model in OpenGL.
- Create four lights, including a spotlight, and create three materials.



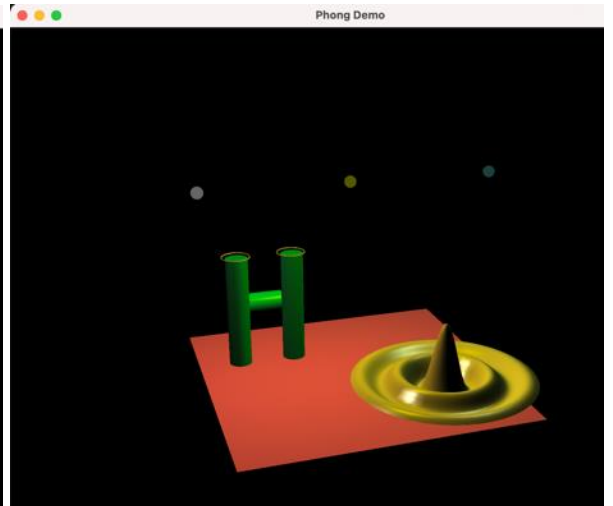
All lights are on



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.

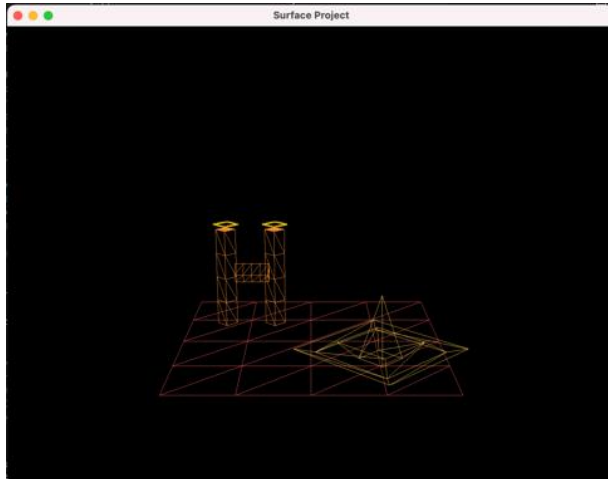
Key press events:

‘w’: toggle wireframe mode

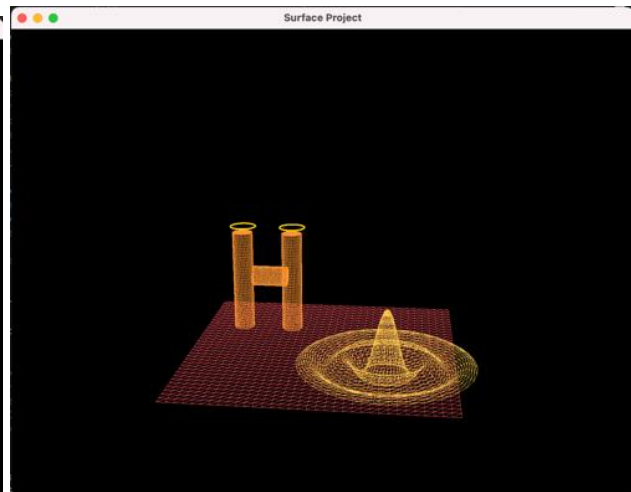
‘c’: toggle backfaces culling

‘M’ or ‘m’: increases or decreases the mesh resolution

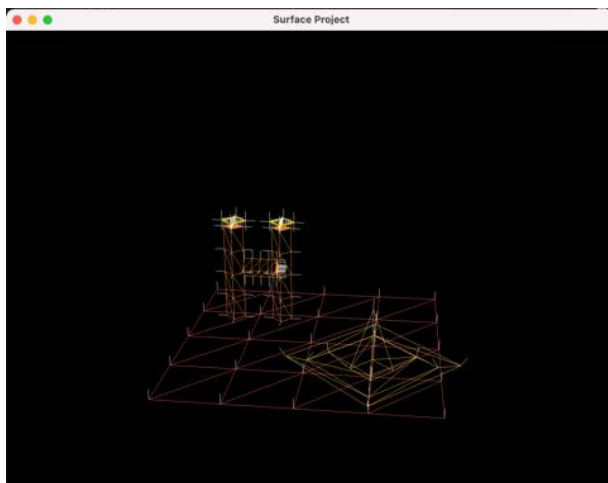
‘N’ or ‘n’: visualizing surface normals



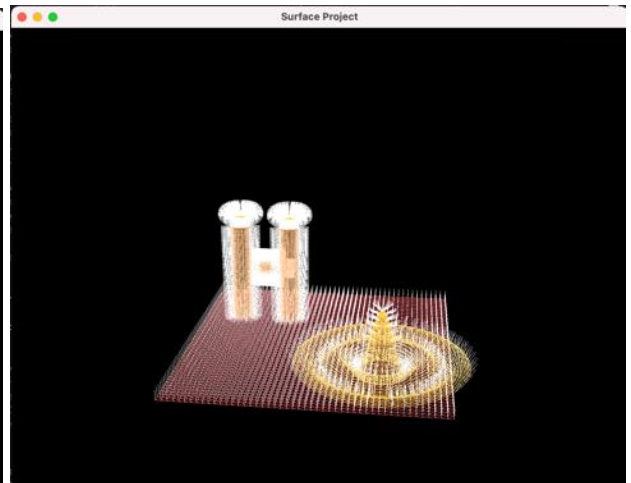
Scene with low resolution



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:

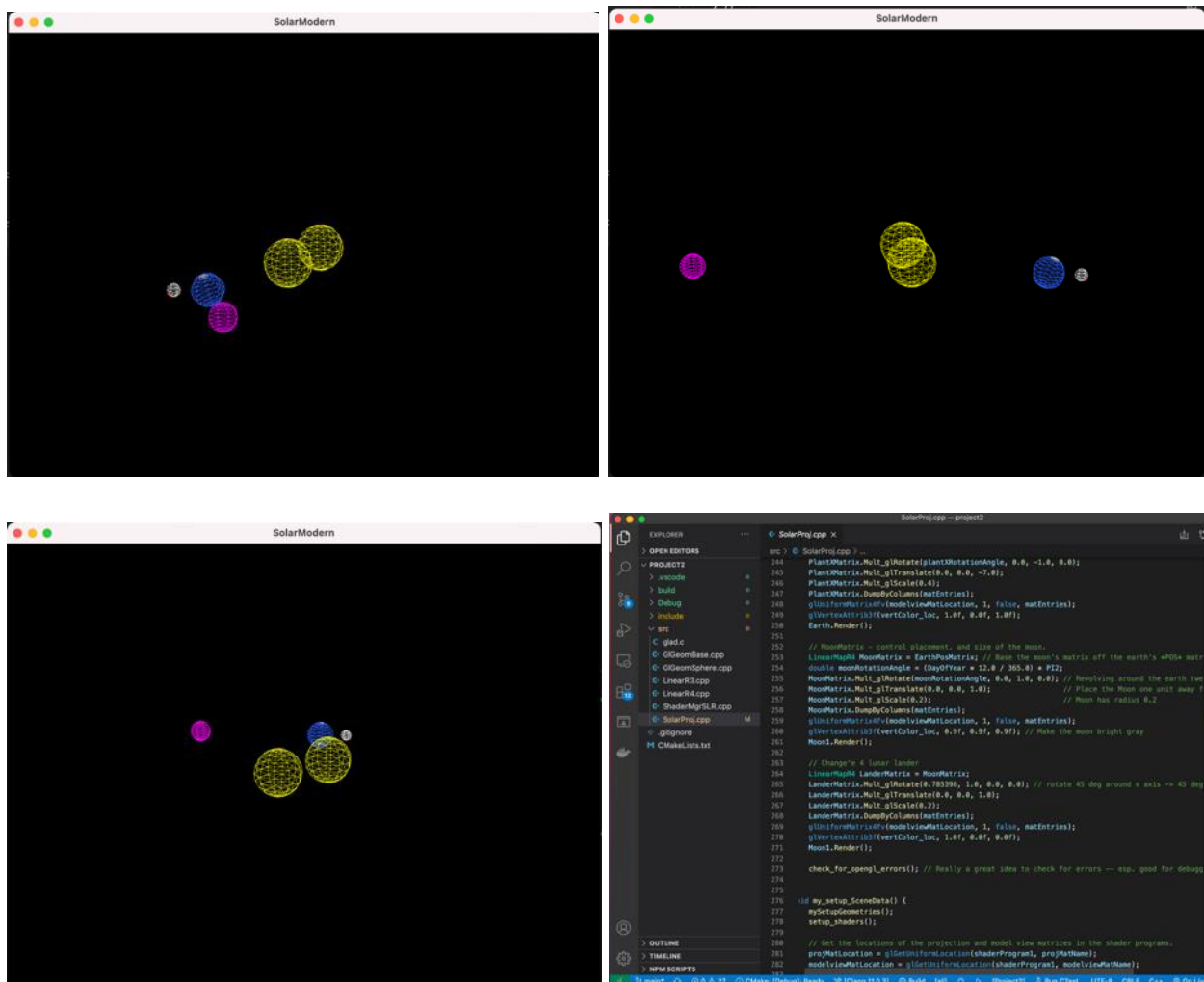
Binary Sun in the middle (Yellow spheres): The two suns revolve around the center of the solar system.

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



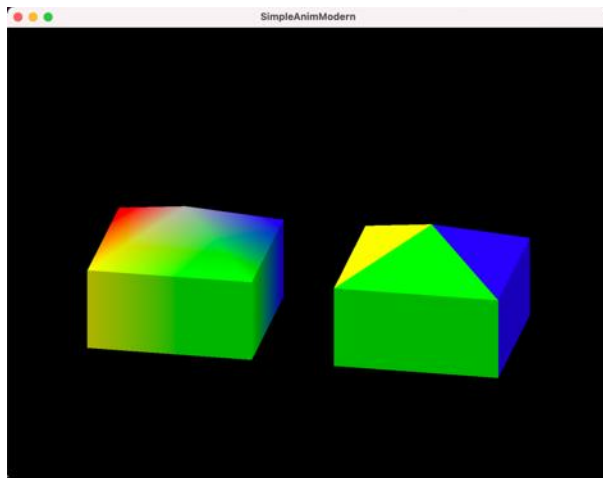
Shaded Tent (C++ with OpenGL)

- 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.

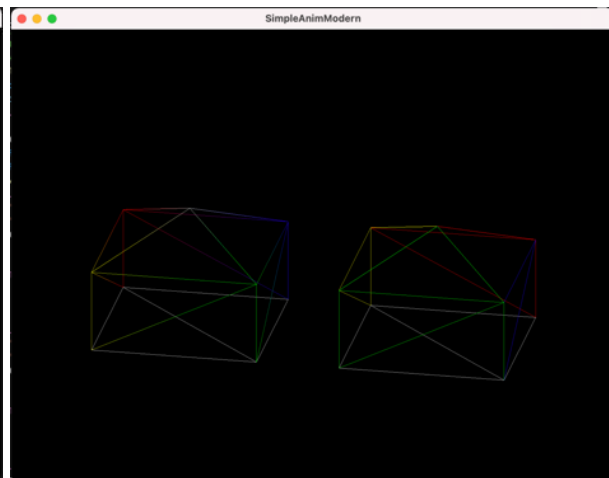
Key press events:

‘w’: toggle wireframe mode

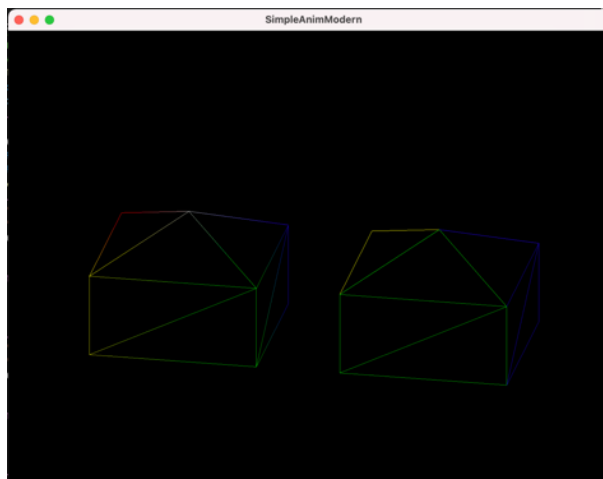
‘c’: toggle backface culling



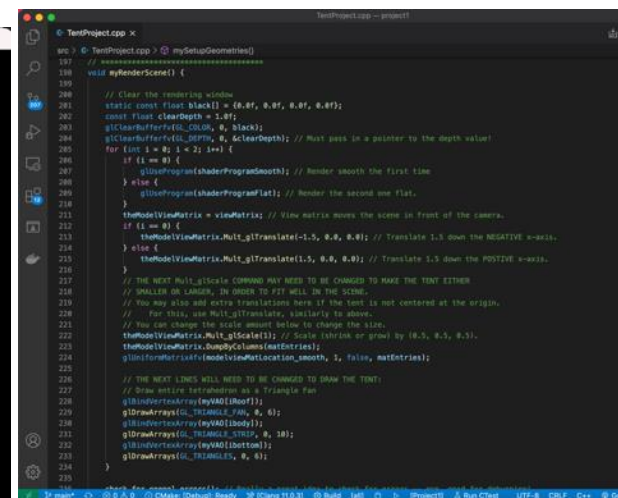
Result Tent



In wireframe mode



Culling backface



Partial code of rendering scene

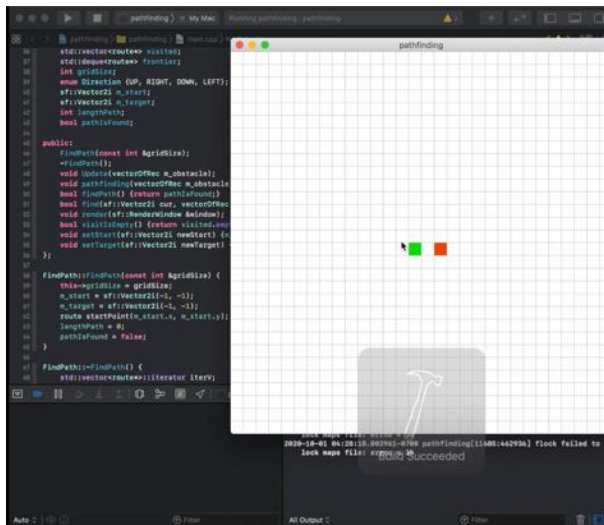
Early Projects (Focus on Game development and algorithm visualization)

Pathfinding Visualizer (C++ with SFML framework)

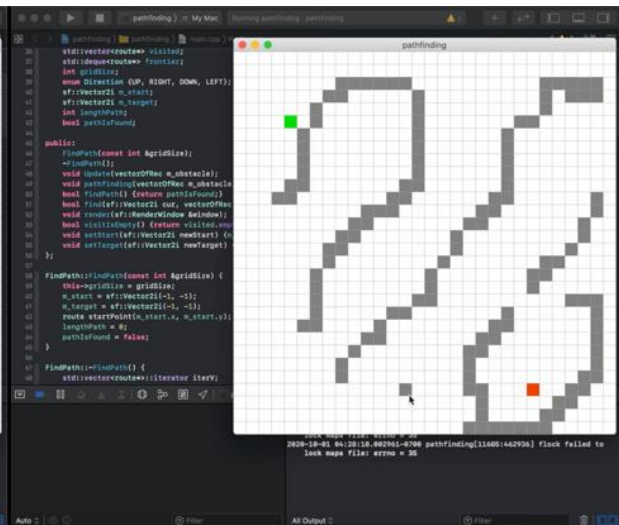
- BFS search algorithm

How to play:

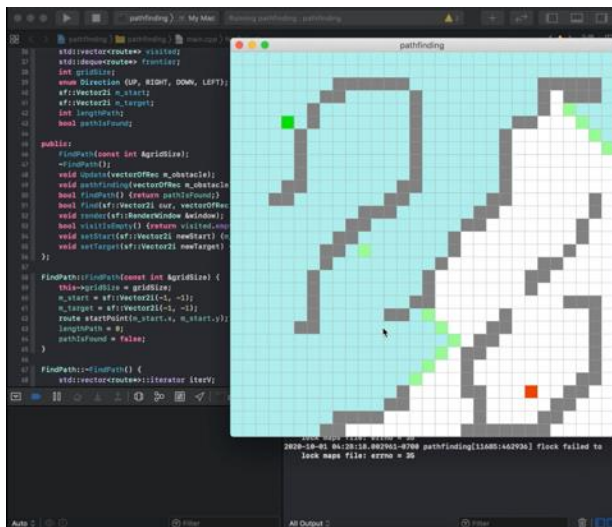
- Drag the green grid and red grid around the board to set the position of the starting point and the target.
- Drawing obstacles
 - o Click anywhere on an empty square (white square) to create an obstacle on the board.
 - o Hold the mouse while drawing obstacles to create continuous obstacles.
- Removing obstacles
 - o Click an obstacle (gray square) again to remove an obstacle on the board.
 - o 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.



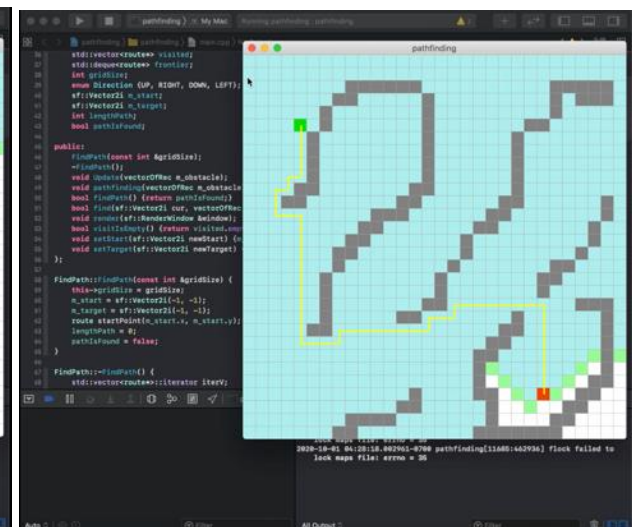
Start Board



Set Obstacles



Start finding path



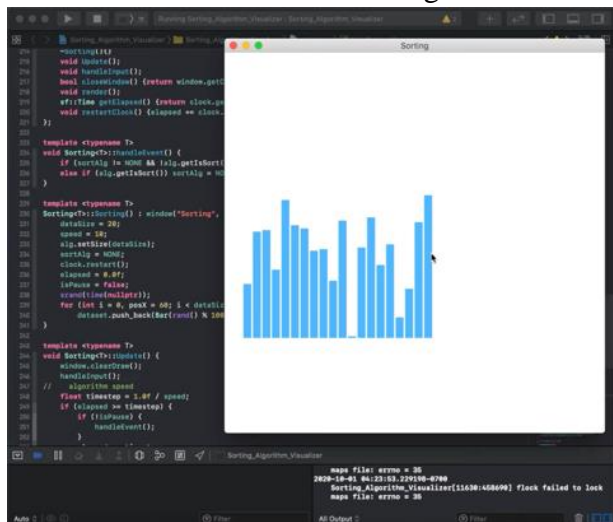
Find a shortest path

Sorting Visualizer (C++ with SFML framework)

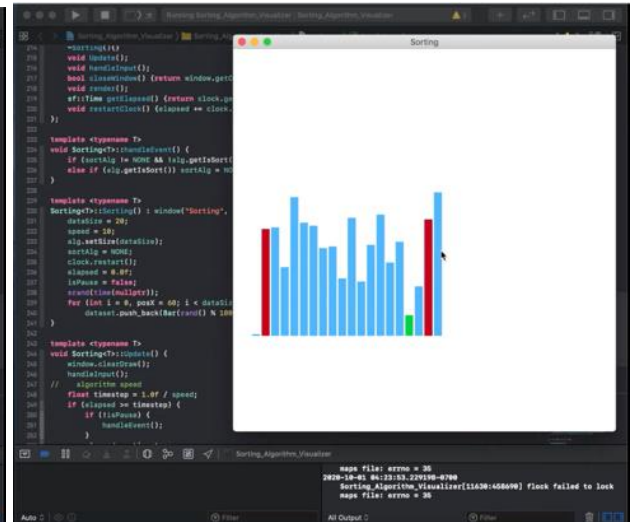
- The following figures are based on selection sort
- Left red bar represents the current index
- Right red bar represents the iterator (loop from current index to the last index to find i^{th} smallest value in the dataset)
- Green bar represents the current i^{th} smallest value

How to play:

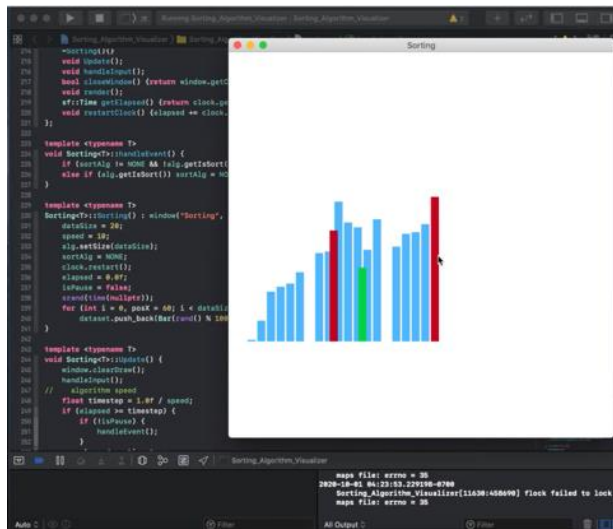
- Press Enter to start sorting



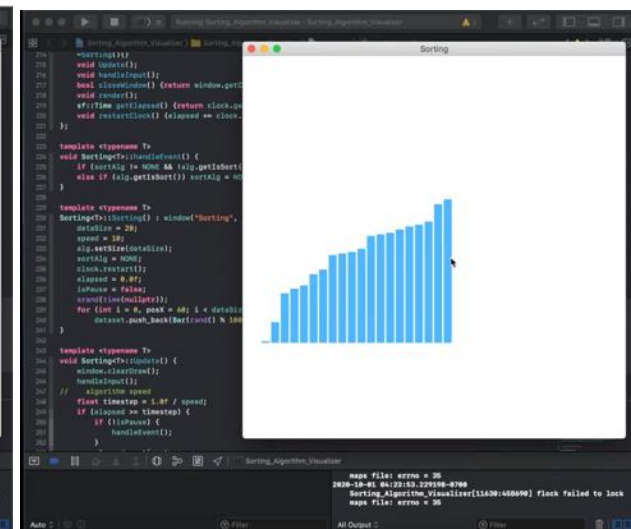
Start Random Dataset



Start sorting



Swapping current index with smallest value



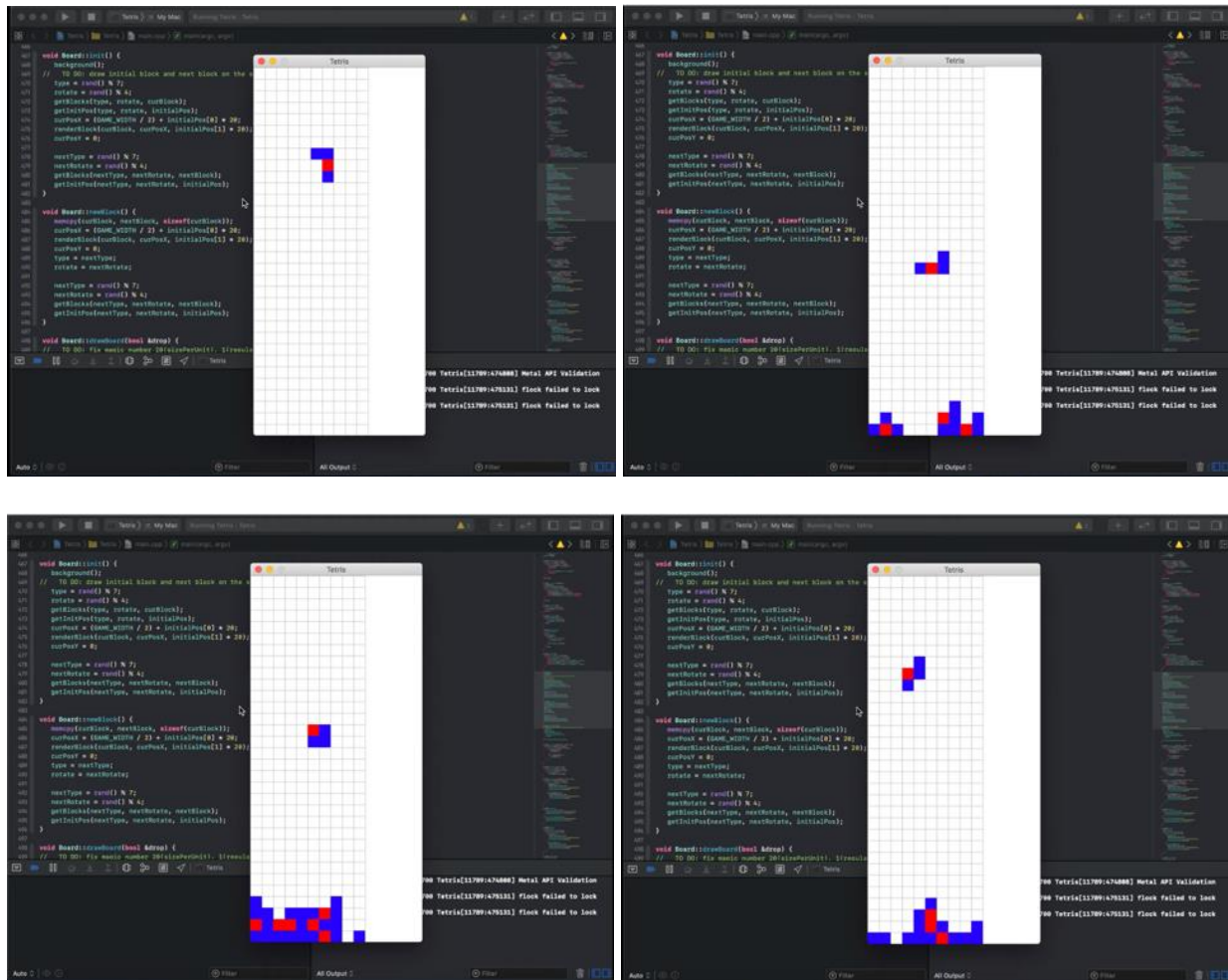
Done sorting

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.

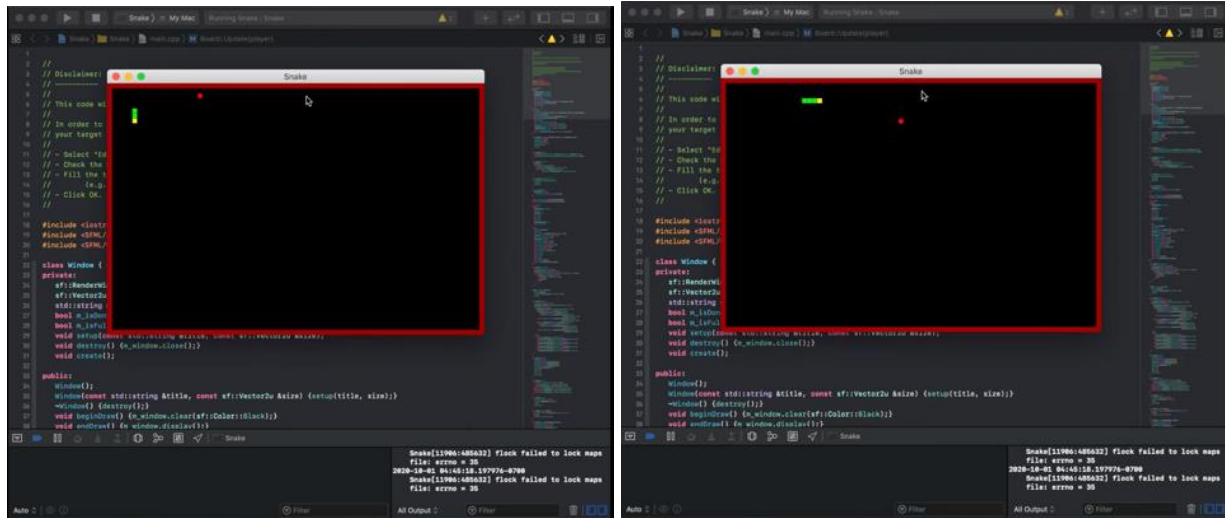


Snake (C++ with SFML framework)

- 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

