Robotron: 2084 inspired Game

Written in PyGame with MVC architecture

John Montgomery
Candidate Number: 5199
Centre Name: Kimberley College Sixth Form
Centre Number: 15125
Qualification: AQA 7517 (A–Level Computer Science)

Supervisor: B. Harris

2020-2022

Contents

List of Figures

Chapter 1

Abstract

In essence the project is an implementation of Robotron in PyGame, using Model-View-Controller, with a Flask based high scores bored. The main content of the code is in the game itself, with flask acting only as an API. This allows for shared usage of the route by a static web page, and by the PyGame code itself. The webpage is simply served off as static, where JS is able to communicate with the API to retrieve the information needed. The database used is Postgres.

Chapter 2

Analysis

2.1 What is MVC?

Model-View-Controller plays a large part in the project, the diagram [Figure 2.1] shows the main way that MVC works. It isolates the components of the game into 3 main components. The View, which is the screen, or what the user will see. The controller, which is where the user interacts with the game, in this case it is the interaction with the keyboard. The model, which is the part the user never interacts with, and stores the state of the game and current information about it.

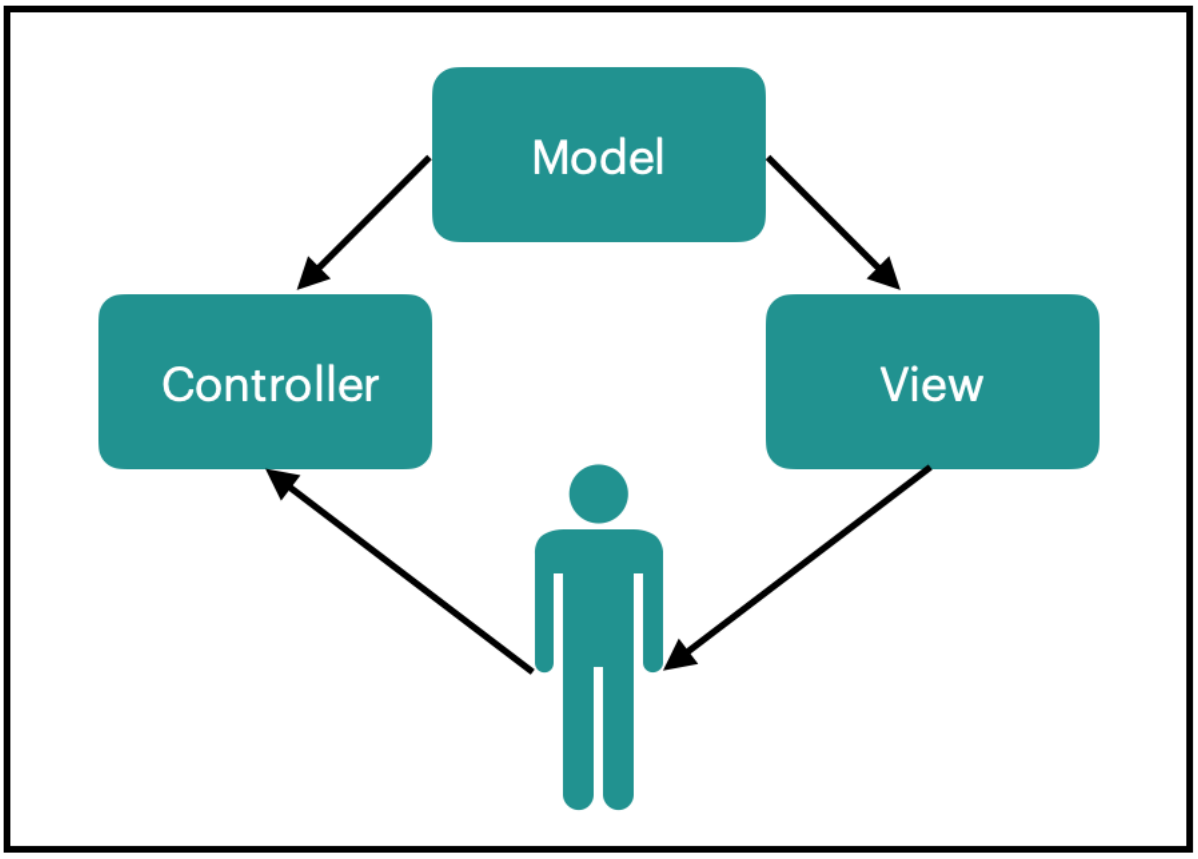

Figure 2.1: A diagram showing the MVC architecture

There are many benefits to this set up, for example, it will easily allow me to swap out what controller is used. If desired, it is much simpler to replace the keyboard as the human interface, and replace it with a game controller. Even more useful may be the ability to remove

the controller and view entirely, allowing for a streamlined game which an AI could learn how to play. This flexibility, along with ease of programming is what drew me to use MVC for the game.

Another important information is the way information travels between the 3 sections. This is done with events, and an event manager is responsible for maintaining the sending and receiving of events through the system. A similarly important section is the States, and state machine, which controls the current 'state' the game is in, that is to say what level is being played, or what screens should be shown, such as a loading or help screen.

2.2 The Game

"Robotron: 2084" was released in 1982 by Williams Electronics. It was revolutionary as a dual stick shooter, was high energy and loved by many. This is important to capture into the game, where I want it to have a similar feeling to the original game, with some modern twists.

The game is about a species of 'Robotrons' created by humans in the year 2084, after realising their failings and created an advanced species. The goal is to save the humans (Mommies, Daddies and Mikeys), whilst fighting the robots, which have many kinds. The most basic are electrodes, which are static obstacles that kill on contact, but can be shot by players. The other basic enemy is the grunt, which is simply a basic soldier, which kills on contact, but moves towards the player. There are some other robots that will be talked about and implemented later, but the details about them are less important.

2.3 Limitations

The dual stick shooter nature means the player uses one joystick to move, and one joystick to shoot. This is difficult to implement well with a keyboard, but a simple setup which I am using is having WASD to move, and IJKL to shoot. Holding 2 keys diagonally at the same time will result it movement in an angle, allowing for shooting in 8 directions, and moving in 8 too.

Robotron is a fast fast game, I had to slow it down slightly in order to make it more playable on my laptop, and so it does feel somewhat different to the original. However by slowing it as I have I have made it a much smoother game to play.

2.4 Objectives

1. Create basic playing ability
    (a) Player can move in 8 directions
    (b) Player can shoot in 8 directions
    (c) Players animation is correct for direction of travel
2. Create basic enemies
    (a) Enemy is spawned in random position
    (b) Enemy can move
    (c) Enemy is animated
    (d) Enemy kills players
3. Create Loading Screens
    (a) Fuzzy loading screen
    (b) 'All test' screen

   (c) Home Screen

4. Create levels and transitions
   (a) Player moves between levels
   (b) Level transitions
   (c) Player is invincible on load
5. Create the API
6. Create login system
   (a) Basic API sign up works
   (b) GUI interactions with PyGame
7. High Scores
   (a) Top 10
   (b) Player Search
8. Create sounds with Game
9. Create scoring and score counter
10. Create a life counter
11. Automate testing on API and basic functions in PyGame

2.5 Design and Inspiration

The design for all the game is heavily taken from the original game. I used many places to research this, but below is a selection of screenshots and videos which were used in the creation of the game.

- https://www.youtube.com/watch?v=ccltMtkFBSI
- https://www.youtube.com/watch?v=aOVA2Axxfdk

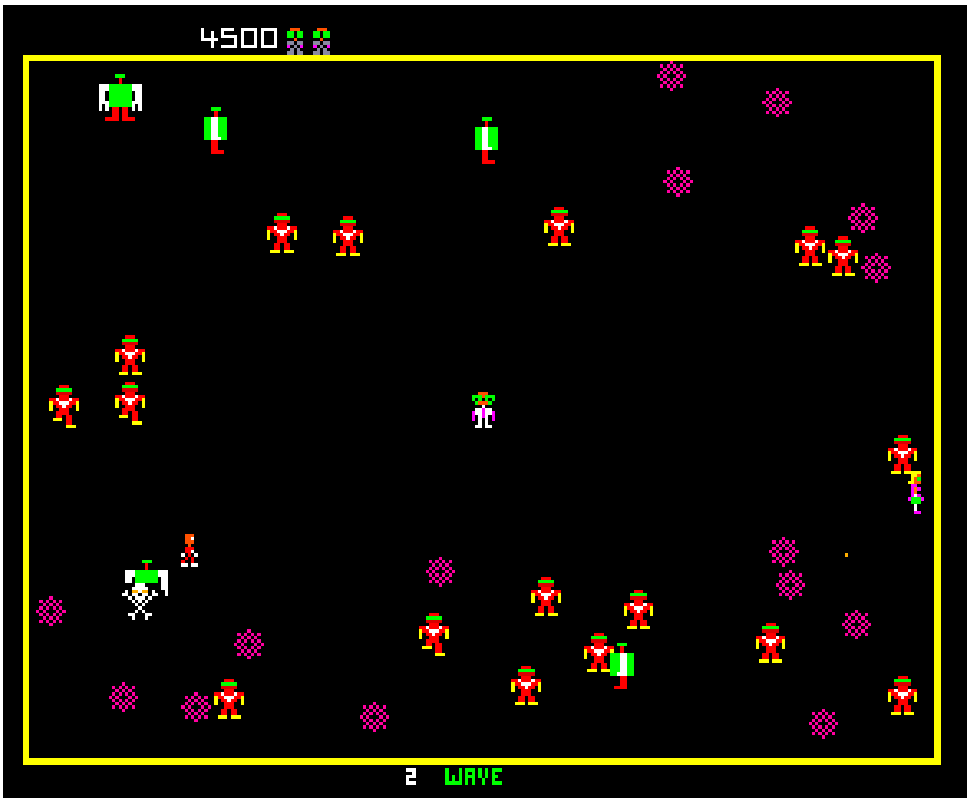

Figure 2.2: Screen from original game - https://arcadeblogger.com/2020/06/27/the-development-of-robotron/

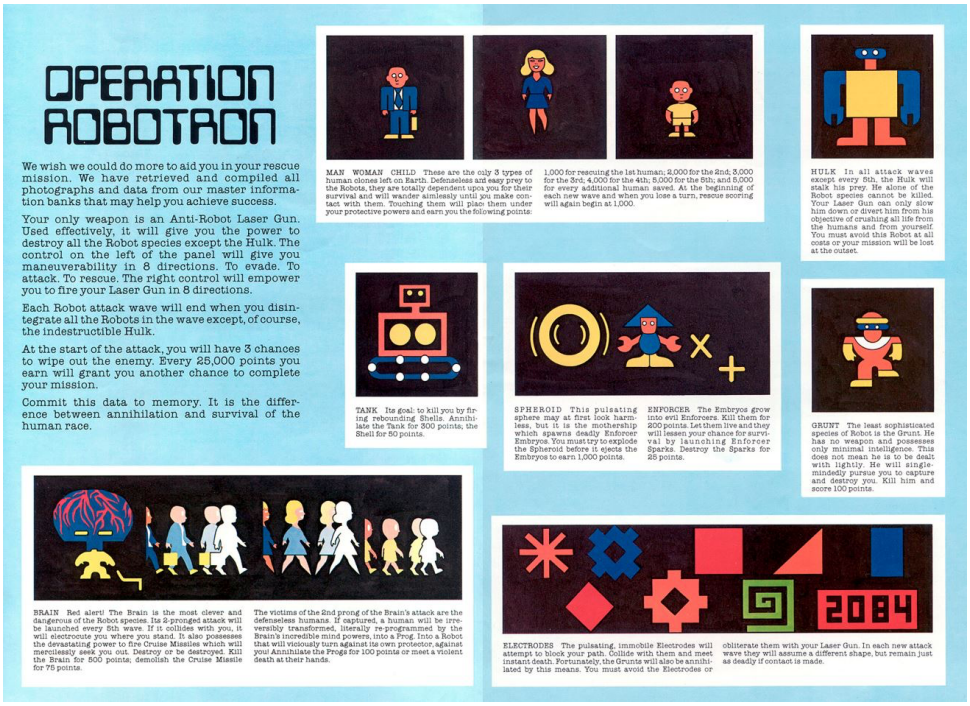

Figure 2.3: Advertising Material - https://arcadeblogger.com/2020/06/27/the-development-of-robotron/

Chapter 3

Documented Design

The main design aspect is the MVC architecture and how it forms the basis of the game. Fig 1, from the analysis section, gave a very brief, high level and non technical view of MVC. In this section I will go into more detail about my own implementation, and how it works in greater detail. This section also details the database on the web side, the API, the technical setup of the servers, the data structures and HCI designs.

3.1 MVC in practice

In the analysis section I gave a very high level overview of MVC, this part will detail further into my design on its implementation in python. The first main, basic components of MVC are of course, the model, the view, and the controller. Figure 3.3 shows the 3 classes diagrams for each of the implementations of these in python.

On top of these key features, there's also a range of other important cogs in the system. One of the most important, to allow for the communication between the M, V and C are Events, and an event manager. A Sample of events, and the event manager is given in Fig fig:events.

The other key class is the state machine. Each state is not given its own class, rather there is a constant number which is attributed to a given state. The states are used for the larger changes in the program and events are for the smaller interactions, and ticks.

In order to run through a basic idea of what happens when the program is run, I have created a step by step flowchart. This flowchart [Fig 9] is a gross oversimplification, but works as a high level description of what it is my code is doing when executed.

3.2 Database

This section will show the database design and set up, and explain some of the SQL used in the program. Fig 10 shows the database diagram.

[TODO - Database diagram]

There are 3 tables, scores, users and tokens. The scores database has 2 fields which store the users ID and their Score for a given game. The Users table stores the users info, such as emails, password hashes, etc, and then the tokens database is used to store validated tokens (with time limits) which are used to validate the GUI and avoids needing to login to the the program every time the game is run. Fig 11 shows the process of creating the tokens.

3.3 The API

The leaderboard contains only 6 routes, as these were all that are necessary, the details for the routes are detailed in the table below.

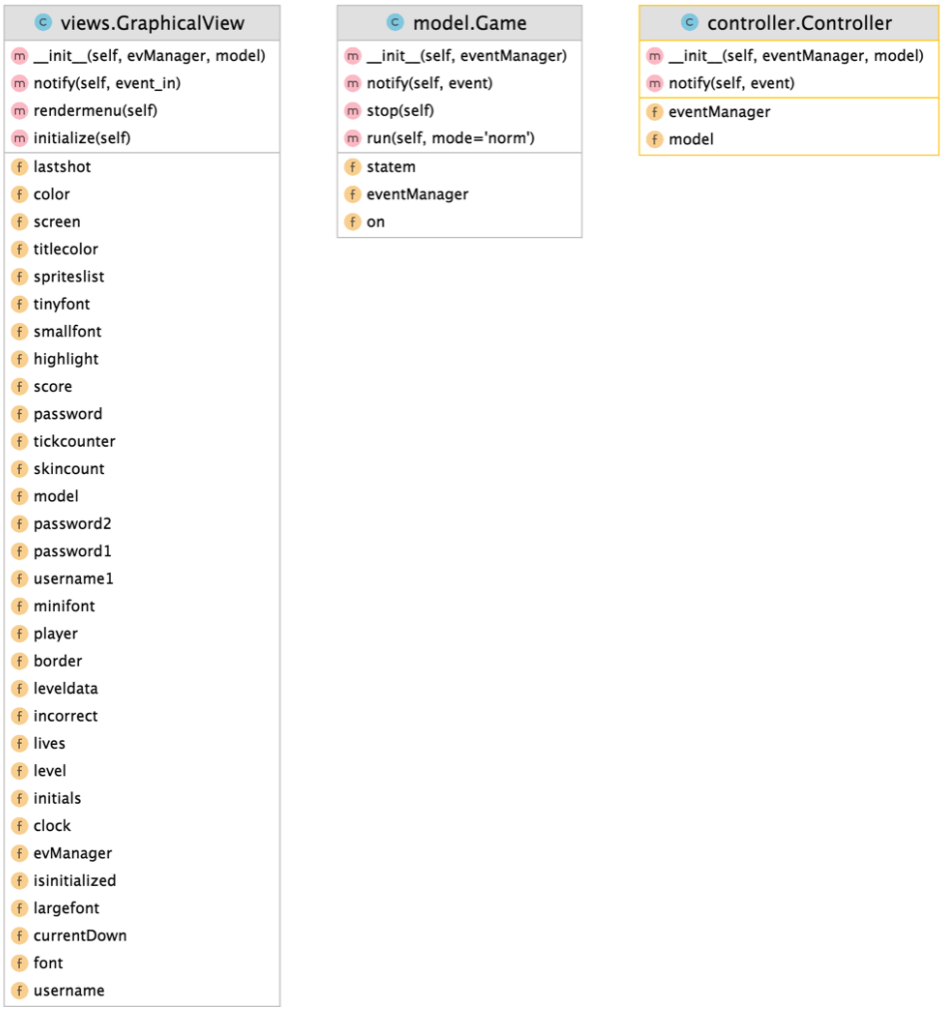

Figure 3.1: Class diagram

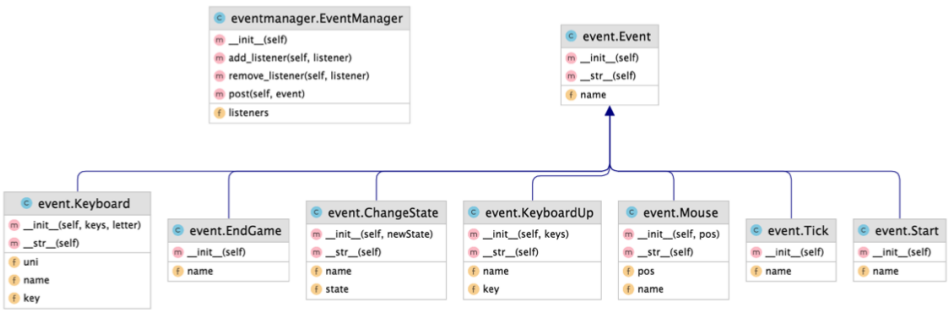

Figure 3.2: Class diagram

ROUTE	METHOD	DESCRIPTION
/leaderboard	GET	Returns JSON of top 10 users (initials + scores) in Database
/user/userid	GET	Returns JSON of top score
/username/userid	GET	Returns ID of given username
/login	POST	Logs in a user, sends token, or logs user in with token
/addscore	POST	Adds a score, given score and a token
/adduser	POST	Adds a user to the database

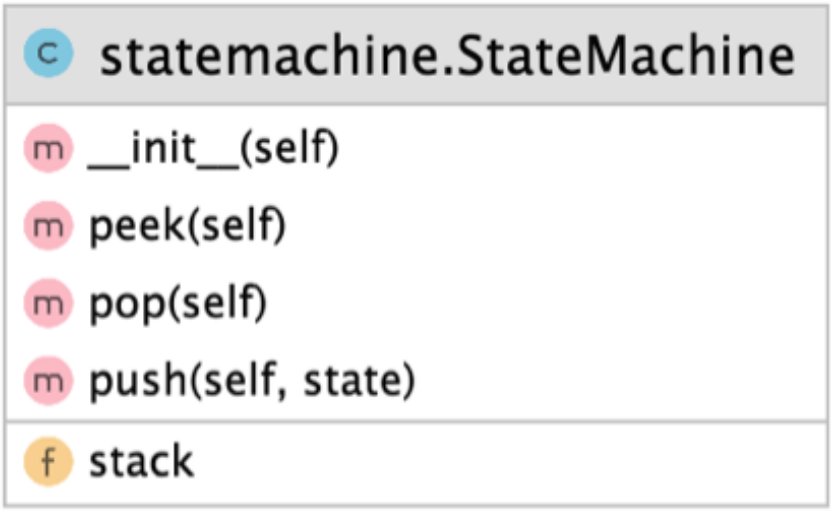

Figure 3.3: Class diagram

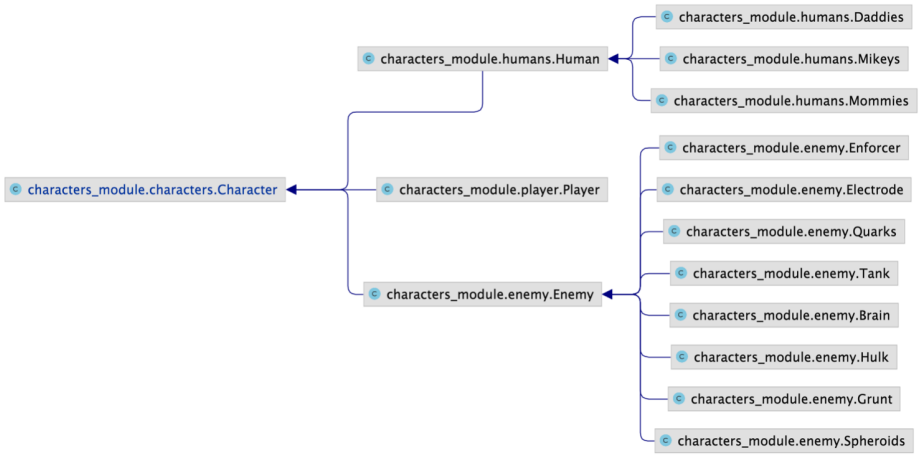

Figure 3.4: Class diagram of characters

3.4 The Server Setup

Fig 12 shows the set up the server is in. All using AWS, there is an RDS Postgres database, and EC2 instance (this is the server running the actual flask) and then an S3 bucket to handle sending the static files. It may also be possible to use NGINX or Apache to serve and handle the API. This system may end up being better, so my current architecture could change.

3.5 Security

Because the database and client handles personal details like email and passwords, there needs to be a thought to security. First off, there is an enforcement of passwords and a strong policy. Users passwords will need to be 8 characters, with 1 special, and my plan is to check them against a list of common passwords (rocky.txt) using hashes. For this I will probably use MD5,

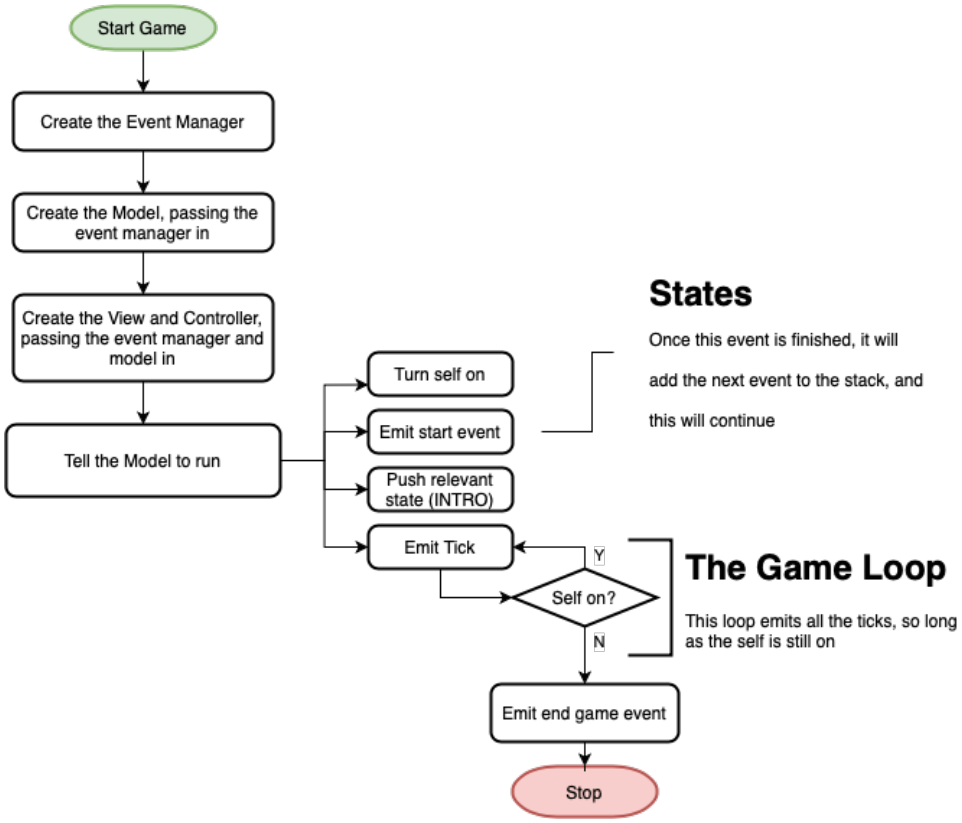

Figure 3.5: Flowchart of MVC

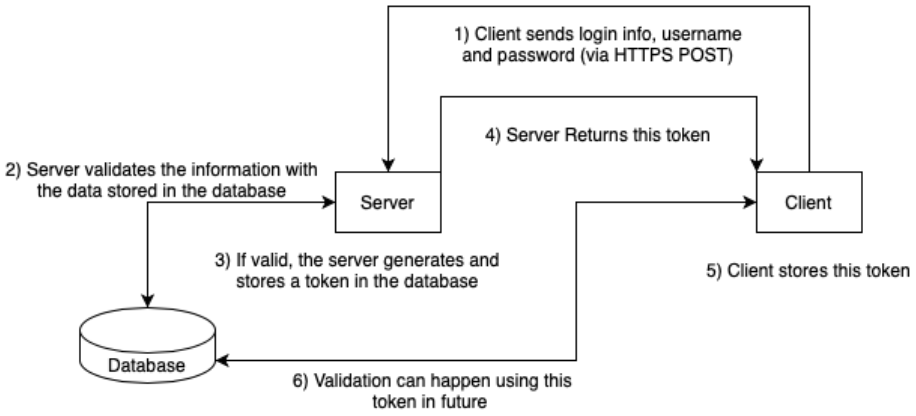

Figure 3.6: How tokens are generated

or something even faster. However it is important to avoid these fast algorithms when hashing passwords for storage. As such, passwords will undergo key derivation through bcrypt, an algorithm which not only salts, but performs many rounds of hashing. I could implement a similar algorithm using the basic functions like SHA, but rolling your own crypto is never good, so its going to be done with bcrypt, as this is essentially the best option available, and more than secure enough.

To help further security, HTTPS is being used for all the sending and receiving of data, this avoids man in the middle attacks of the data as it gets sent over the internet.

Chapter 4

Technical Solution

Im working on a way to easily do this. I am using LaTeX as this will allow me to not have to change the code on my report manually, as it will link to the files. However i am still learning. This file will also update on my github, so if you go to https://github.com/john-montgomery2003/Robotron2084 you will be able to find a pdf of this document which is more up to date.

4.1 Example

Chapter 5

Testing - TODO

Chapter 6

Evaluation - TODO

Chapter 7

Appendix & Bibliography

7.1 Appendix

Name	Server/Web/Game/Dev	Use
Flask	Server	Handles the API and web on server side
SQLalchemy	Server	Used to connect to the Postgres database
BCrypt	Server	Key derivation
Waitress	Server	WSGI server
PyGame	Game	Graphics and input handling
S3	Server	AWS static file hosting / serving
EC2	Server	AWS server to run flask app
Hetzner	Server	Alternative option to run flask and serve files
PyCharm	Dev	My IDE choice

7.2 Listings

Listings

7.2.1 Website Code

app.py

```
from waitress import serve
# Flask is used to handle the web requests
from flask import Flask, jsonify, request, render_template

# Sql alchemy handles all SQL interactions, but rather than using and overly relying on the ORM,
# Ill use raw SQL commands. The SQL server is running on RDS (AWS) with PostgreSQL
from sqlalchemy import create_engine
from sqlalchemy.orm import scoped_session, sessionmaker

# Allow CORS - so it will work from both the webserver and python
from flask_cors import CORS

# This is used to hash passwords and validate them - could of used a different tool, or built it myself, but
# But this is prebuilt and purpose designed
import bcrypt
import secrets
# This starts the App
app = Flask(__name__)
# Allow the cors to work
CORS(app)
# Gets the database URL, creates the connection

engine = create_engine(
'sqlite:///test.db',
connect_args={'check_same_thread': False}
)

db = scoped_session(sessionmaker(bind=engine))


db.execute('''
CREATE TABLE IF NOT EXISTS leaderboard (
    id INTEGER UNIQUE PRIMARY KEY AUTOINCREMENT,
    initials VARCHAR(255),
    username VARCHAR(255) UNIQUE,
    password VARCHAR(255)
)''')
db.commit()
```

```
db.execute('''
CREATE TABLE IF NOT EXISTS scores(
    id INT,
    scores INT
)
''')
db.commit()
db.execute('''
CREATE TABLE IF NOT EXISTS tokens (
    id INT,
    token VARCHAR
    )
''')
db.commit()

@app.route('/test', methods=['GET'])
def test():
    return render_template('error.html')

@app.route('/', methods=['GET'])
def index():
    leaders = db.execute('''SELECT leaderboard.initials, scores
    FROM leaderboard
    LEFT JOIN scores
    ON leaderboard.id = scores.id
    ORDER BY scores DESC
    LIMIT 10;''')
    # ...so we convert it into a dictionary
    a, d = [], {}
    for lead in leaders:
        for column, value in lead.items():
            d = {**d, **{column: value}}
        a.append(d)
    return render_template('index.html', a=a)


@app.errorhandler(500)
def page_not_found(e):
    # note that we set the 404 status explicitly
    return render_template('error.html')


@app.route('/robo/leaderboard', methods=['GET'])
def leader():
    """
    This route fetches the top 10 results from the server, allowing the page to
     display the leaderbaord
    :return:
    """
    # This returns a Result Proxy object...
    leaders = db.execute('''SELECT leaderboard.initials, scores
FROM leaderboard
LEFT JOIN scores
ON leaderboard.id = scores.id
ORDER BY scores DESC
LIMIT 10;''')
    # ...so we convert it into a dictionary
    a , d= [], {}
    for lead in leaders:
        for column, value in lead.items():
            d = {**d, **{column: value}}
        a.append(d)

```

```
    return jsonify(a)


@app.route('/robo/user/<string:userid>', methods=['GET'])
def user(userid):
    """
    This returns a users high score, given their ID - this means that the API will have to fetch the ID first
    Could it have used the username? probably.
    :param userid:
    :return:
    """
    score = list(db.execute(f'''SELECT score
FROM leaderboard
LEFT JOIN scores
ON leaderboard.id = scores.id
WHERE leaderboard.id = {userid}
ORDER BY scores DESC
LIMIT 1;'''))[0][0]
    return jsonify({'score': score})


@app.route('/robo/userid/<string:username>', methods=['GET'])
def useridget(username):

    """
    This is used to get the id of a user, from their username (which has to be unique)
    Returns a 0 if the username is not unique
    :param username:
    :return:
    """
    userid = list(db.execute(f"""SELECT leaderboard.id
FROM leaderboard
WHERE leaderboard.username = '{username}'
LIMIT 1;"""))
    try:
        print(userid)
        return jsonify({'id': userid[0][0]})
    except IndexError:
        return jsonify({'id': 0})


@app.route('/login', methods=['POST'])
def login():
    """
    Used to login to the game, returns a  token which is used to verify the user.
    :return:
    """

    userid = request.values.get('userid')
    password = request.values.get('password')
    print(userid)
    hashed = list(db.execute(f'''SELECT password
        FROM leaderboard
        WHERE leaderboard.id = {userid}
        LIMIT 1;'''))[0][0]
    valid = bcrypt.checkpw(password.encode(), hashed.encode())
    if not valid:
        return jsonify({'message': 'password fail'})
    else:
        try:
```

```
161             token = list(db.execute(f'''SELECT token
162             FROM tokens
163             WHERE id = {userid}
164             LIMIT 1;'''))[0][0]
165             return jsonify({'token': token})
166         except:
167             token = secrets.token_urlsafe(30)
168             db.execute(f"""INSERT INTO tokens (id, token)
169                 VALUES ('{userid}','{token}');""")
170             db.commit()
171             return jsonify({'token': token})
172
173
174 @app.route('/robo/addscore', methods=['POST'])
175 def add():
176     """
177     Used to add scores to the database, uses a post request. Must provide a
    password to add a score.
178     This might be slightly annoying, but adding in functionality for tokens and
     storing them in python
179     feels like a lot of work, maybe I will, but I probably wont invest my time
    there, I could always cache the
180     password inputted in the python code instead.
181     :return:
182     """
183     userid = request.values.get('userid')
184     score = int(request.values.get('score'))
185     token = request.values.get('token')
186
187     tokenDB = list(db.execute(f'''SELECT token
188     FROM tokens
189     WHERE id = {userid}
190     LIMIT 1;'''))[0][0]
191     valid = token == tokenDB
192
193     if not valid:
194         return jsonify({'message': 'password fail'})
195     try:
196         db.execute(f'''INSERT INTO scores (id, scores)
197     VALUES ({userid},{score});''')
198         db.commit()
199
200         return jsonify({'message': 'success'})
201     except:
202         return jsonify({'message': 'fail'})
203
204
205 @app.route('/robo/adduser', methods=['POST'])
206 def adduser():
207     """
208     This is the API used to add a user to the database, users provide a
    username, initials and their password.
209     Password validation will be done client side, need to keep this app as
    lightweight as possible.
210     :return:
211     """
212     username = request.values.get('username')
213     initials = request.values.get('initials')
214     password = request.values.get('password')
215
216     tostore = bcrypt.hashpw(password.encode(), bcrypt.gensalt()).decode()
217
218     db.execute(f"""INSERT INTO leaderboard (initials, username, password)
```

```
    VALUES ('{initials}','{username}','{tostore}');""")
    db.commit()

    return jsonify({'message': 'success'})


if __name__ == '__main__':
    serve(app, host="0.0.0.0", port=80)
```

index.html

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
    <title>Robotron</title>
    <meta name="theme-color" content="rgb(194,1,0)">
    <meta name="description" content="Robotron leaderboard for robotron by John Montgomery - a pygame game.">
    <link rel="icon" type="image/png" sizes="360x360" href="../static/img/Screenshot%202020-12-13%20at%2020.56.23.png">
    <link rel="icon" type="image/png" sizes="360x360" href="../static/img/Screenshot%202020-12-13%20at%2020.56.23.png">
    <link rel="icon" type="image/png" sizes="360x360" href="../static/img/Screenshot%202020-12-13%20at%2020.56.23.png">
    <link rel="icon" type="image/png" sizes="360x360" href="../static/img/Screenshot%202020-12-13%20at%2020.56.23.png">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="manifest" href="manifest.json">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/aos/2.2.0/aos.css">
    <link rel="stylesheet" href="../static/css/styles.css">
</head>

<body style="background: rgb(0,0,0);max-height: 100vh">
    <div data-aos="zoom-out" data-aos-duration="2000" style="margin-right: 1%;margin-bottom: 0;margin-left: 1%;height: 98vh;width: 98%;margin-top: 1vh;border: 3px dotted #9f095c;">
        <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #9f095c;">
            <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #970b60;">
                <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #970b60;">
                    <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #900c64;">
                        <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #900c64;">
                            <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #880e68;">
                                <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #880e68;">
                                    <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #81106b;">
                                        <div style="margin-top: 0px;margin-
```

```
    right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;
    border: 3px dotted #81106b;">
31                                              <div style="margin-top: 0px;margin-
    right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;
    border: 3px dotted #7a126f;">
32                                                  <div style="margin-top: 0px;
    margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width:
    100%;border: 3px dotted #7a126f;">
33                                                      <div style="margin-top: 0px
    ;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width:
    100%;border: 3px dotted #721473;">
34                                                          <div style="margin-top:
     0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;
    width: 100%;border: 3px dotted #721473;">
35                                                              <div style="margin-
    top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;
    width: 100%;border: 3px dotted #6b1577;">
36                                                                  <div style="
    margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height
    : 100%;width: 100%;border: 3px dotted #6b1577;">
37                                                                      <div style=
    "margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;
    height: 100%;width: 100%;border: 3px dotted #63177b;">
38                                                                          <div
    style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px
    ;height: 100%;width: 100%;border: 3px dotted #63177b;">
39                                                                              <
    div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left:
     0px;height: 100%;width: 100%;border: 3px dotted #5b197e;">
40
     <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-
    left: 0px;height: 100%;width: 100%;border: 3px dotted #5b197e;">
41
         <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin
    -left: 0px;height: 100%;width: 100%;border: 3px dotted #541b82;">
42
             <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;
    margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #541b82;">
43
                 <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0
    px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #4d1c86;">
44
                     <div style="margin-top: 0px;margin-right: 0px;margin-bottom
    : 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #4d1c86;"
    >
45
                         <div style="margin-top: 0px;margin-right: 0px;margin-
    bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted
    #451e8a;">
46
                             <div style="margin-top: 0px;margin-right: 0px;
    margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px
    dotted #451e8a;">
47
                                 <h2 style="color: rgb(69,31,138);font-family:
    Conv_robotron-2084;text-align: center;margin-top: 9px;">robotron heroes</h2>
48
                                 <div class="container" style="padding-right: 50
    px;padding-left: 50px;margin-top: 50px;">
49
                                     <div class="row" style="margin-right: -15px
    ;">
50
```

```
                                        <div class="col">
51
                                            <h3 style="font-family:
    Conv_robotron-2084;color: rgb(255,51,38);">1 > {{ a[0].initials }} - {{ a
    [0].scores }}</h3>
52
                                        </div>
53
                                        <div class="col">
54
                                            <h3 style="font-family:
    Conv_robotron-2084;color: rgb(255,51,38);">6 > {{ a[5].initials }} - {{ a
    [5].scores }}</h3>
55
                                        </div>
56
                                    </div>
57
                                    <div class="row" style="margin-right: -15px
    ;">
58
                                        <div class="col">
59
                                            <h3 style="font-family:
    Conv_robotron-2084;color: rgb(255,51,38);">2 > {{ a[1].initials }} - {{ a
    [1].scores }}</h3>
60
                                        </div>
61
                                        <div class="col" style="font-family:
    Conv_robotron-2084;color: rgb(255,51,38);">
62
                                            <h3 style="font-family:
    Conv_robotron-2084;color: rgb(255,51,38);">7 > {{ a[6].initials }} - {{ a
    [6].scores }}</h3>
63
                                        </div>
64
                                    </div>
65
                                    <div class="row" style="margin-right: -15px
    ;">
66
                                        <div class="col">
67
                                            <h3 style="font-family:
    Conv_robotron-2084;color: rgb(255,51,38);">3 > {{ a[2].initials }} - {{ a
    [2].scores }}</h3>
68
                                        </div>
69
                                        <div class="col">
70
                                            <h3 style="font-family:
    Conv_robotron-2084;color: rgb(255,51,38);">8 > {{ a[7].initials }} - {{ a
    [7].scores }}</h3>
71
                                        </div>
72
                                    </div>
73
                                    <div class="row" style="margin-right: -15px
    ;">
```

```
74                                      <div class="col" style="font-family:
    Conv_robotron-2084;color: rgb(255,51,38);">
75                                          <h3 style="font-family:
    Conv_robotron-2084;color: rgb(255,51,38);">4 > {{ a[3].initials }} - {{ a
    [3].scores }}</h3>
76                                      </div>
77                                      <div class="col">
78                                          <h3 style="font-family:
    Conv_robotron-2084;color: rgb(255,51,38);">9 > {{ a[8].initials }} - {{ a
    [8].scores }}</h3>
79                                      </div>
80                                  </div>
81                                  <div class="row" style="margin-right: -15px
    ;">
82                                      <div class="col">
83                                          <h3 style="font-family:
    Conv_robotron-2084;color: rgb(255,51,38);">5 > {{ a[4].initials }} - {{ a
    [4].scores }}</h3>
84                                      </div>
85                                      <div class="col">
86                                          <h3 style="font-family:
    Conv_robotron-2084;color: rgb(255,51,38);">10 > {{ a[9].initials }} - {{ a
    [9].scores }}</h3>
87                                      </div>
88                                  </div>
89                              </div>
90                              <div class="row" style="margin-top: 10%;">
91                                  <div class="col">
92                                      <h1></h1>
93                                      <h2 style="color: rgb(69,31,138);font-
    family: Conv_robotron-2084;text-align: center;margin-top: 9px;">play the
    game</h2>
94                                      <p style="font-family: Conv_robotron
    -2084;color: rgb(254,51,38);text-align: center;margin-top: 16px;font-size:
    16px;">Get the game - <a href="#">Github</a></p>
95                                      <p style="font-family: Conv_robotron
    -2084;color: rgb(254,51,38);text-align: center;margin-top: 16px;font-size:
    16px;">Original game info - <a href="#">here</a></p>
96                                  </div>
97
```

```
                                    </div>
98
                                    <h2 style="color: rgb(113,113,113);font-family:
     Conv_robotron-2084;text-align: center;margin-top: 50px;font-size: 12px;">&
    nbsp;by John Montgomery</h2>
99
                                </div>
100
                            </div>
101
                        </div>
102
                    </div>
103
                </div>
104
            </div>
105
        </div>
106                                                                                 </
    div>
107                                                                                 </div>
108                                                                             </div>
109                                                                         </div>
110                                                                     </div>
111                                                                 </div>
112                                                             </div>
113                                                         </div>
114                                                     </div>
115                                                 </div>
116                                             </div>
117                                         </div>
118                                     </div>
119                                 </div>
120                             </div>
121                         </div>
122                     </div>
123                 </div>
124             </div>
125
126             <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min
    .js"></script>
127             <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap
    /4.5.2/js/bootstrap.bundle.min.js"></script>
128             <script src="https://cdnjs.cloudflare.com/ajax/libs/aos/2.2.0/aos.js"></
    script>
129             <script src="../static/js/script.min.js"></script>
130 </body>
131
132 </html>
```

error.html

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0,
    shrink-to-fit=no">
7     <title>Robotron</title>
8     <meta name="theme-color" content="rgb(194,1,0)">
9     <meta name="description" content="Robotron leaderboard for robotron by John
     Montgomery - a pygame game.">
```

```
10      <link rel="icon" type="image/png" sizes="360x360" href="../static/img/
    Screenshot%202020-12-13%20at%2020.56.23.png">
11      <link rel="icon" type="image/png" sizes="360x360" href="../static/img/
    Screenshot%202020-12-13%20at%2020.56.23.png">
12      <link rel="icon" type="image/png" sizes="360x360" href="../static/img/
    Screenshot%202020-12-13%20at%2020.56.23.png">
13      <link rel="icon" type="image/png" sizes="360x360" href="../static/img/
    Screenshot%202020-12-13%20at%2020.56.23.png">
14      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter
    -bootstrap/4.5.2/css/bootstrap.min.css">
15      <link rel="manifest" href="manifest.json">
16      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/aos
    /2.2.0/aos.css">
17      <link rel="stylesheet" href="../static/css/styles.css">
18  </head>
19
20  <body style="background: rgb(0,0,0);">
21      <div data-aos="zoom-out" data-aos-duration="2000" style="margin-right: 1%;
    margin-bottom: 0;margin-left: 1%;height: 98vh;width: 98%;margin-top: 1vh;
    border: 3px dotted #9f095c;">
22          <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin
    -left: 0px;height: 100%;width: 100%;border: 3px dotted #9f095c;">
23              <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;
    margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #970b60;">
24                  <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0
    px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #970b60;">
25                      <div style="margin-top: 0px;margin-right: 0px;margin-bottom
    : 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #900c64;"
    >
26                          <div style="margin-top: 0px;margin-right: 0px;margin-
    bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted
    #900c64;">
27                              <div style="margin-top: 0px;margin-right: 0px;
    margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px
    dotted #880e68;">
28                                  <div style="margin-top: 0px;margin-right: 0px;
    margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px
    dotted #880e68;">
29                                      <div style="margin-top: 0px;margin-right: 0
    px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px
    dotted #81106b;">
30                                          <div style="margin-top: 0px;margin-
    right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;
    border: 3px dotted #81106b;">
31                                              <div style="margin-top: 0px;margin-
    right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;
    border: 3px dotted #7a126f;">
32                                                  <div style="margin-top: 0px;
    margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width:
    100%;border: 3px dotted #7a126f;">
33                                                      <div style="margin-top: 0px
    ;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;width:
    100%;border: 3px dotted #721473;">
34                                                          <div style="margin-top:
     0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;
    width: 100%;border: 3px dotted #721473;">
35                                                              <div style="margin-
    top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height: 100%;
    width: 100%;border: 3px dotted #6b1577;">
36                                                                  <div style="
    margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;height
    : 100%;width: 100%;border: 3px dotted #6b1577;">
37                                                                      <div style=
```

```
    "margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px;
    height: 100%;width: 100%;border: 3px dotted #63177b;">
38                                                                      <div
    style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left: 0px
    ;height: 100%;width: 100%;border: 3px dotted #63177b;">
39                                                                          <
    div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-left:
     0px;height: 100%;width: 100%;border: 3px dotted #5b197e;">
40
     <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin-
    left: 0px;height: 100%;width: 100%;border: 3px dotted #5b197e;">
41
          <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;margin
    -left: 0px;height: 100%;width: 100%;border: 3px dotted #541b82;">
42
               <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0px;
    margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #541b82;">
43
                    <div style="margin-top: 0px;margin-right: 0px;margin-bottom: 0
    px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #4d1c86;">
44
                         <div style="margin-top: 0px;margin-right: 0px;margin-bottom
    : 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted #4d1c86;"
    >
45
                              <div style="margin-top: 0px;margin-right: 0px;margin-
    bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px dotted
    #451e8a;">
46
                                   <div style="margin-top: 0px;margin-right: 0px;
    margin-bottom: 0px;margin-left: 0px;height: 100%;width: 100%;border: 3px
    dotted #451e8a;">
47
                                        <h1 style="color: rgb(69,31,138);font-family:
    Conv_robotron-2084;text-align: center;margin-top: 9px;">UH OH</h1>
48
49
                                        <h2 style="color: rgb(69,31,138);font-family:
    Conv_robotron-2084;text-align: center;margin-top: 9px;">Something went wrong
     :(</h2>
50
51
                                        <h2 style="color: rgb(113,113,113);font-family:
     Conv_robotron-2084;text-align: center;margin-top: 50px;font-size: 12px;">&
    nbsp;by John Montgomery</h2>
52
                                   </div>
53
                              </div>
54
                         </div>
55
                    </div>
56
               </div>
57
          </div>
58
     </div>
59                                                                          </
    div>
60                                                                      </div>
61                                                                  </div>
```

```
                                                                    </div>
                                                                </div>
                                                            </div>
                                                        </div>
                                                    </div>
                                                </div>
                                            </div>
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.5.2/js/bootstrap.bundle.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/aos/2.2.0/aos.js"></script>
    <script src="../static/js/script.min.js"></script>
</body>

</html>
```

styles.css

```
@font-face {
    font-family: Conv_robotron-2084;
    src: url(../fonts/robotron-2084.eot) format("embedded-opentype"), url(../fonts/robotron-2084.woff) format("woff"), url(../fonts/robotron-2084.ttf) format("truetype"), url(../fonts/robotron-2084.svg) format("svg");
    font-weight: 400;
    font-style: normal;
}
#inputcmd,
body {
    background-color: #333;
    color: #0f0;
    font-family: "andale mono", "monotype.com", monaco, "courier new", courier, monospace;
}
#terminal-window {
    padding: 10px;
    display: block;
    position: absolute;
    width: 100%;
    height: 100%;
    top: 0;
    left: 0;
    background-color: #111;
    overflow: hidden;
}
#terminal-window:before {
    content: "";
    z-index: 4010;
    width: 100%;
    height: 100%;
    position: absolute;
    top: 0;
    left: 0;
    background: linear-gradient(#444 50%, #111 50%);
```

```
    background-size: 100% 4px;
    background-repeat: repeat-y;
    opacity: 0.14;
    box-shadow: inset 0 0 1px 1px rgba(0, 0, 0, 0.8);
    animation: 5s linear infinite pulse;
}
#cursor {
    color: #0f0;
    box-sizing: border-box;
    border-left: 0.5em solid;
}
.blink {
    animation: 6s steps(13, end) infinite typing, 1s step-end infinite blinking
    ;
}
.scanlines {
    z-index: 4100;
}
.hide {
    display: none;
}

#inputcmd {
    background-color: #111;
    border: 1px;
    font-size: 1em;
    color: transparent;
    text-shadow: 0 0 0 #0f0;
}
#inputcmd:focus {
    outline: 0;
}
@keyframes pulse {
    0% {
        transform: scale(1.001);
        opacity: 0.14;
    }
    8% {
        transform: scale(1);
        opacity: 0.13;
    }
    15% {
        transform: scale(1.004);
        opacity: 0.14;
    }
    30% {
        transform: scale(1.002);
        opacity: 0.11;
    }
    100% {
        transform: scale(1);
        opacity: 0.14;
    }
}
@keyframes vline {
    0% {
        top: 0;
    }
    100% {
        top: 100%;
    }
}
@keyframes blinking {
```

```
95      from,
96      to {
97          border-color: transparent;
98      }
99      50% {
100         border-color: green;
101     }
102 }
```

7.2.2 Game Code