EXERCISE_1 REPORT

Γιώργος Μαθιουδάκης

Csd4674

In this project I implemented a serverless architecture on AWS. **Part 1** of the report will be the **User Access Management** . **Part 2** of the report will be the **Scoring System** . **Part 3** will be the background options . The code used is included in separate .py files .

The objective of this project is :

- Designing and implementing user access management to facilitate registration and profile retrieval
- Developing a scoring system that allows for real-time score submissions and updates
- Configuring notifications to keep players informed about their rankings
- Integrating features that allow players to customize their game backgrounds

Services used for this project:

- Lamda function
- DynamoDB
- API Gateway
- SNS
- SQS
- S3
- Step Functions
- Cloudwatch

Important notes for the project to work:

In order for the game to play in our custom server we need to navigate to 'Settings' -> 'Server' -> 'Custom Server' and click enter .

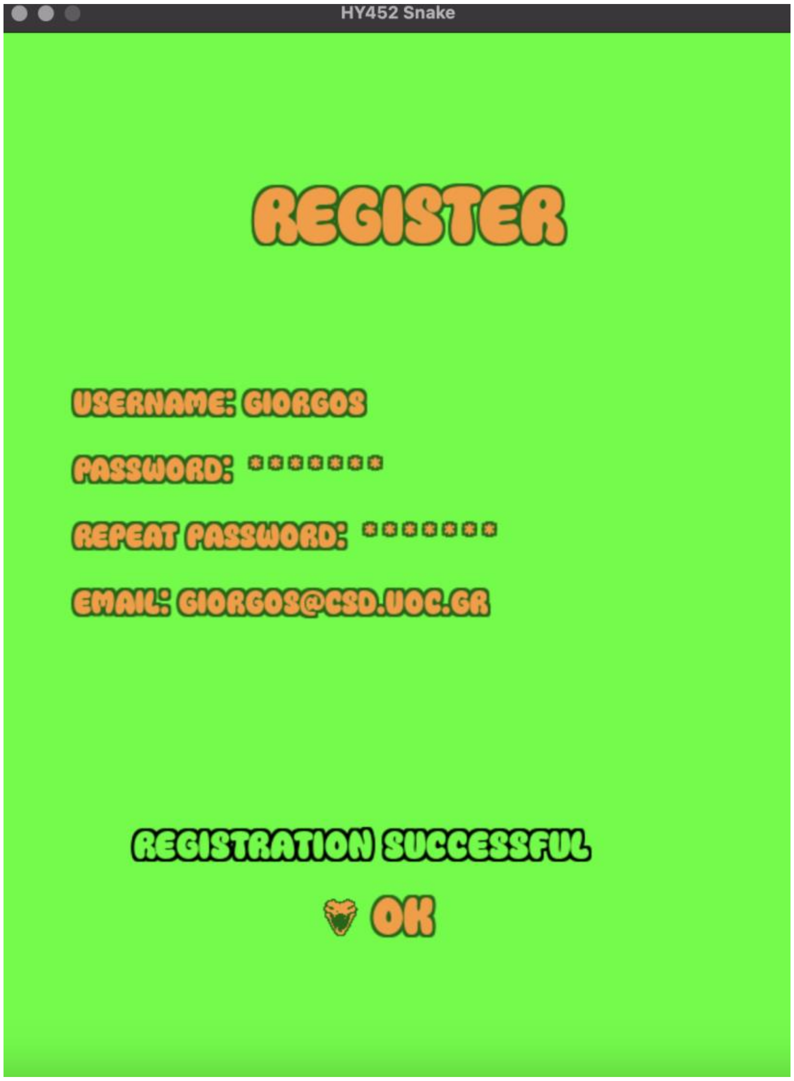
The custom_url_server: https://pnypivou7d.execute-api.us-east-1.amazonaws.com/alpha

Part 1 – User Access Management

It includes 1. Register and 2. Login . I will demonstrate them in this order so first we can register and then login into these accounts starting from empty tables.

1.

Register In game



USERNAME: GIORGOS

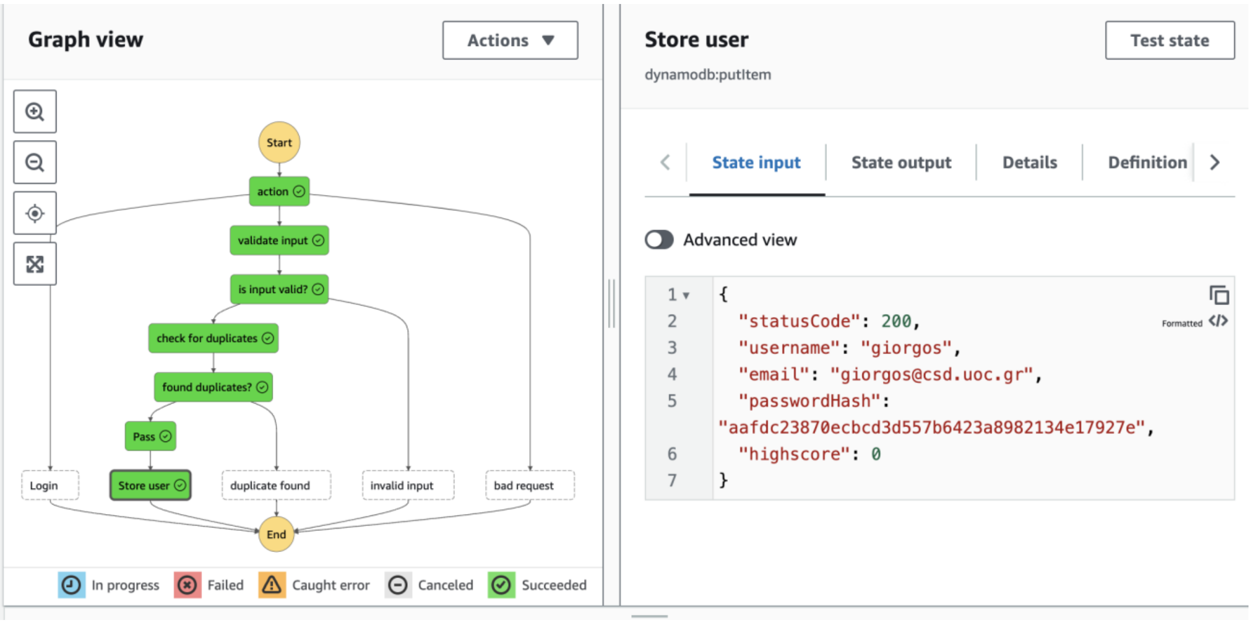
PASSWORD: *******

REPEAT PASSWORD: *******

EMAIL: GIORGOS@CSD.UOC.GR

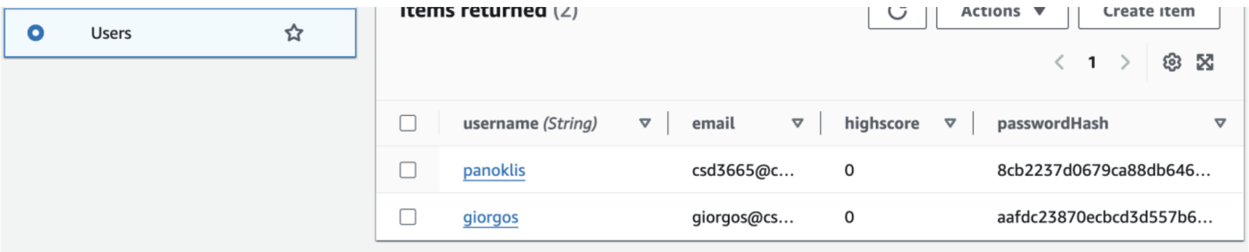
REGISTRATION SUCCESSFUL

What that looks like in State Machine:



```
{
  "statusCode": 200,
  "username": "giorgos",
  "email": "giorgos@csd.uoc.gr",
  "passwordHash":
"aafdc23870ecbcd3d557b6423a8982134e17927e",
  "highscore": 0
}
```

And also the new user in table : (there is one more user from testing , focus on giorgos)

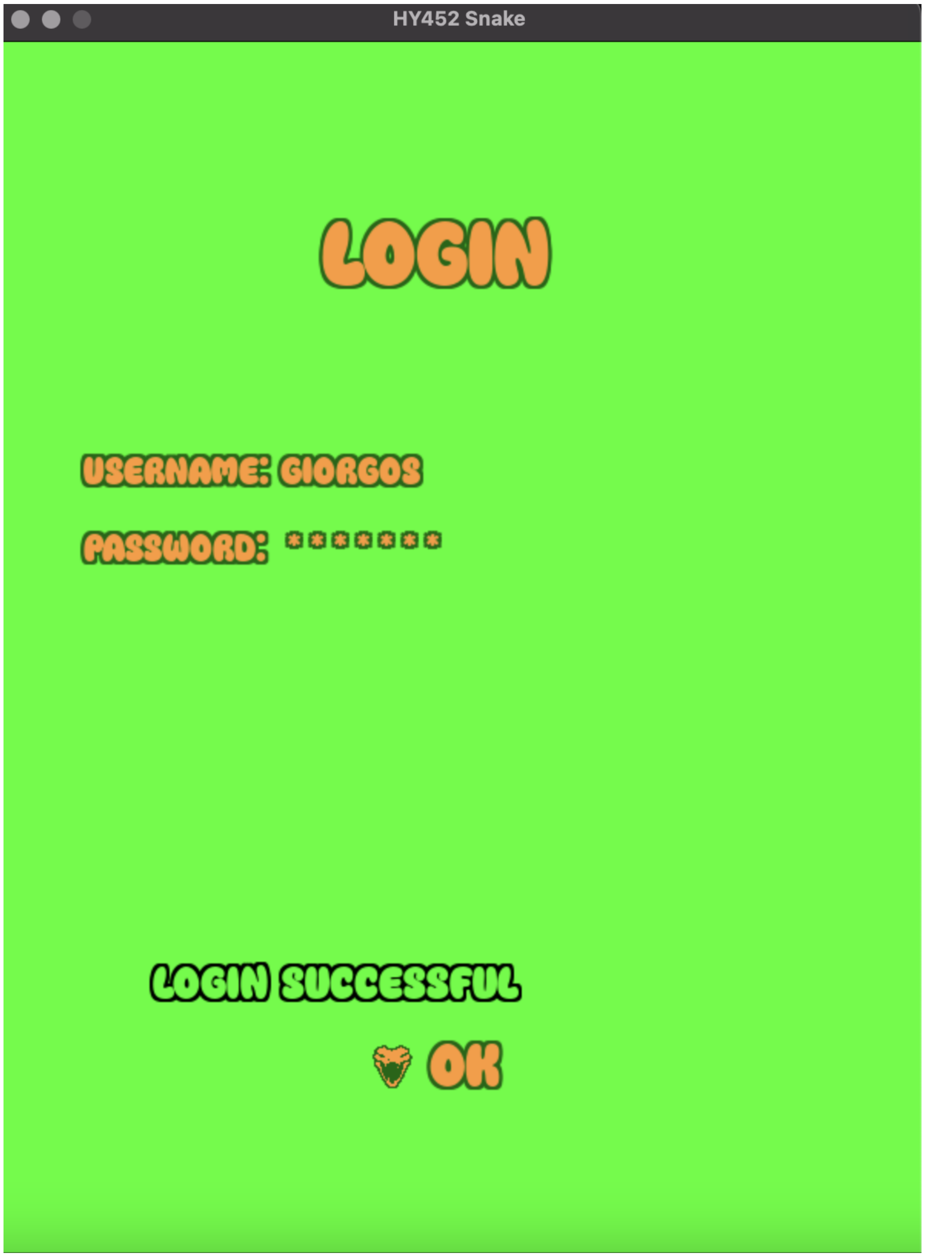


□	username (String)	email	highscore	passwordHash
□	panoklis	csd3665@c...	0	8cb2237d0679ca88db646...
□	giorgos	giorgos@cs...	0	aafdc23870ecbcd3d557b6...

Now lets Login into that user .

2.

Login in game



HY452 Snake

LOGIN

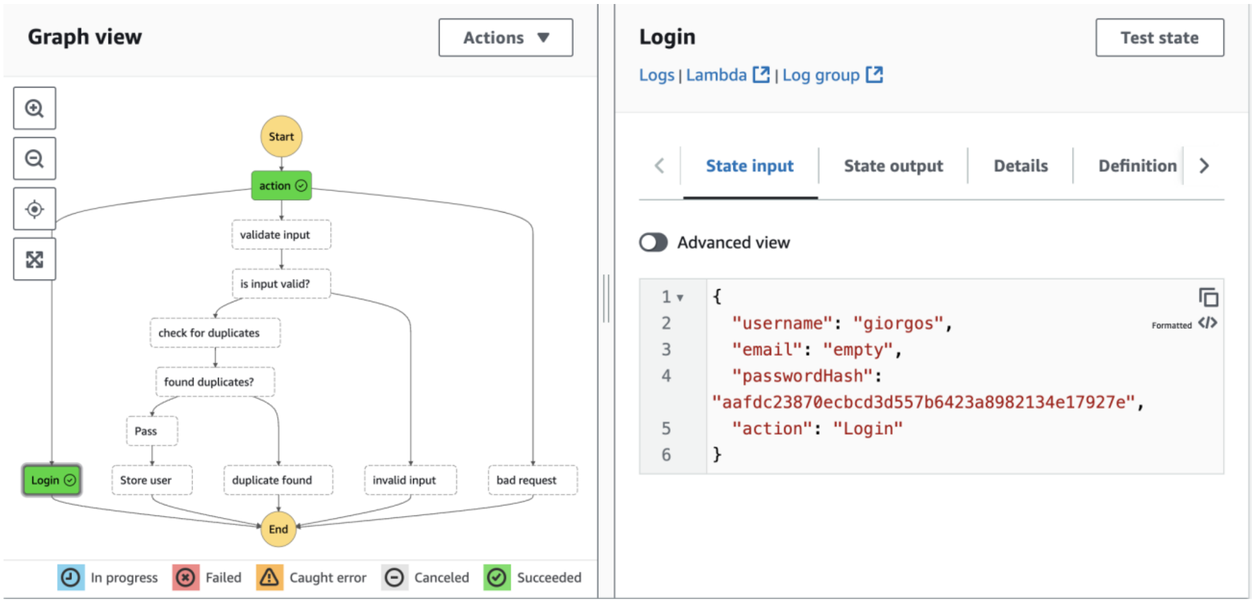
USERNAME: GIORGOS

PASSWORD: *******

LOGIN SUCCESSFUL

OK

Login rightfully successful . Lets check our state machine:



○ Advanced view

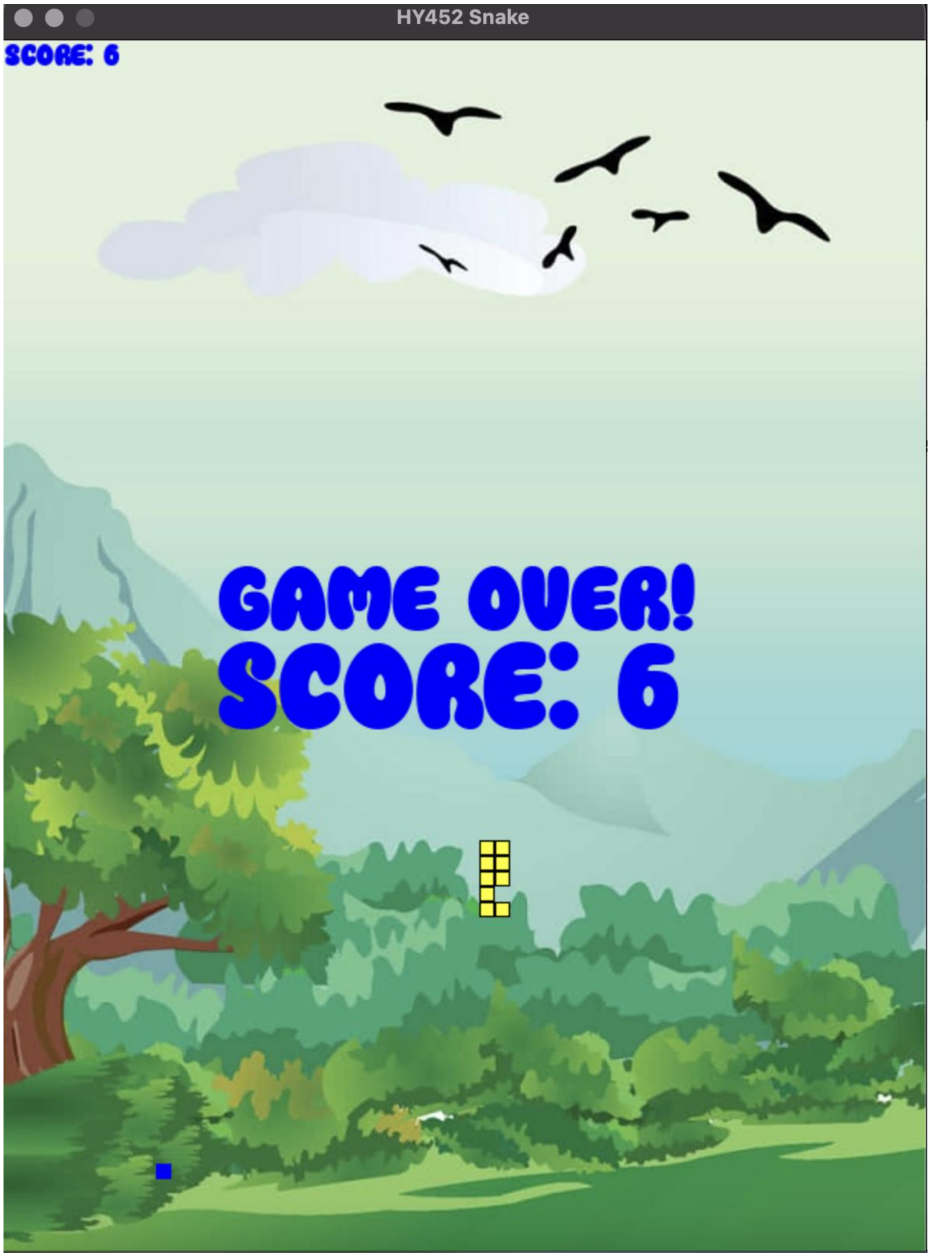
```
{
  "username": "giorgos",
  "email": "empty",
  "passwordHash":
"aafdc23870ecbcd3d557b6423a8982134e17927e",
  "action": "Login"
}
```

Part 2 – Scoring System

In this part of the project we expect to see the user score points . That should trigger a change in the Users table if the score is a new highscore , and a change in Leaderboard where the user will get a new spot if it is their first time playing and their score attribute will also increase if it is the highest one yet . Also for highscores a notification will appear in our inbox of the mail/phone we have arranged .

In account Giorgos the highscore is 0 since the account is new so any score should trigger all changes in tables and a notifaction as well .

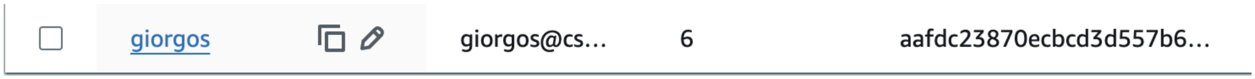
Lets play!



I managed to score 6 ! Lets track all the changes that happened during the game .

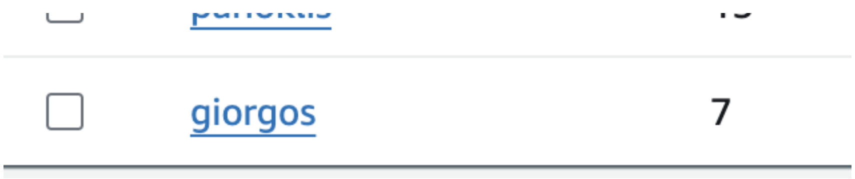
Firstly the tables :

Users table:



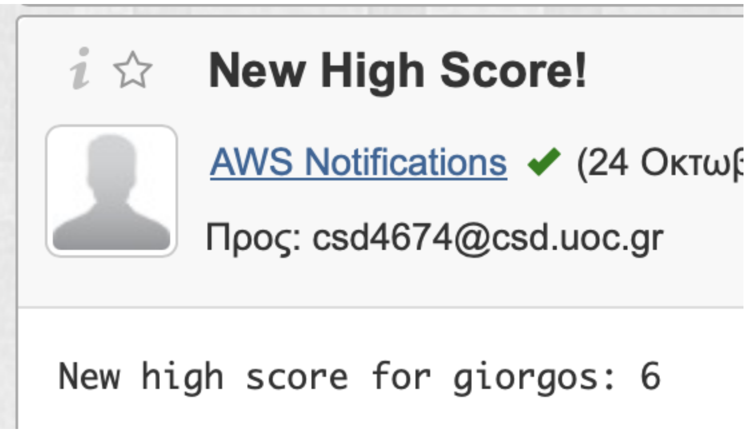
□	giorgos	giorgos@cs…	6	aafdc23870ecbcd3d557b6…

Leaderboard:



□	giorgos	7

And the notification:



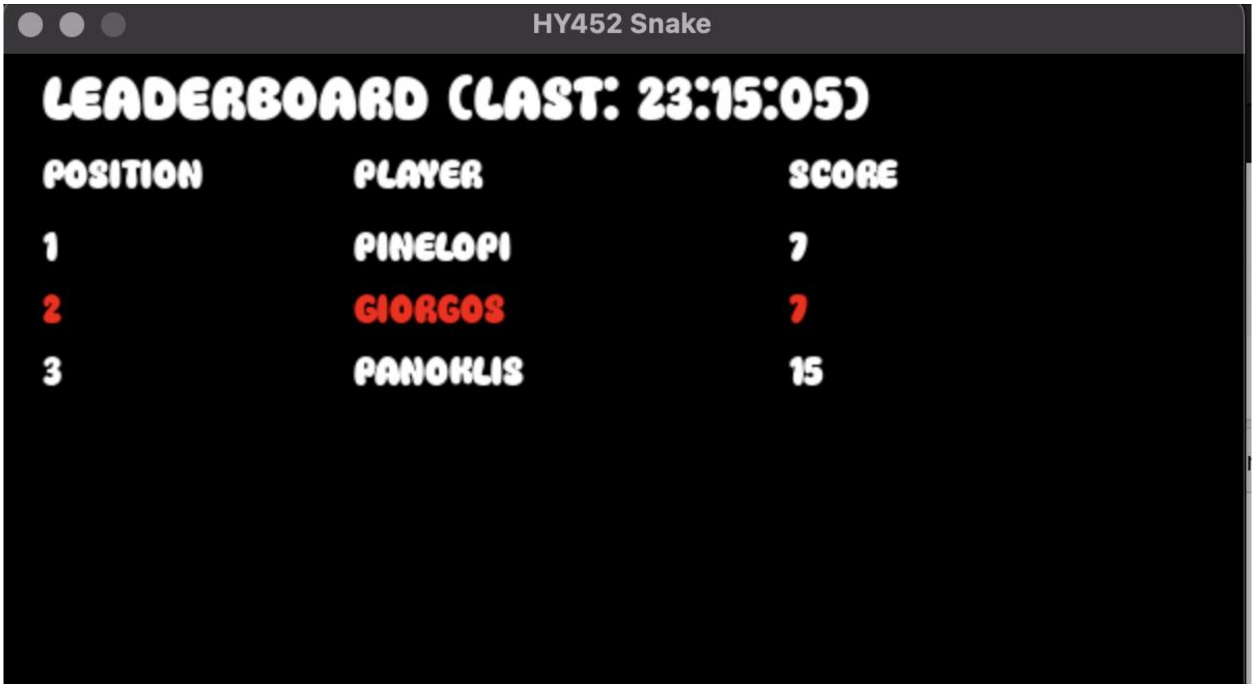
New High Score!

AWS Notifications ✓ (24 Οκτωβ

Προς: csd4674@csd.uoc.gr

```
New high score for giorgos: 6
```

We can also see the leaderboard in game :

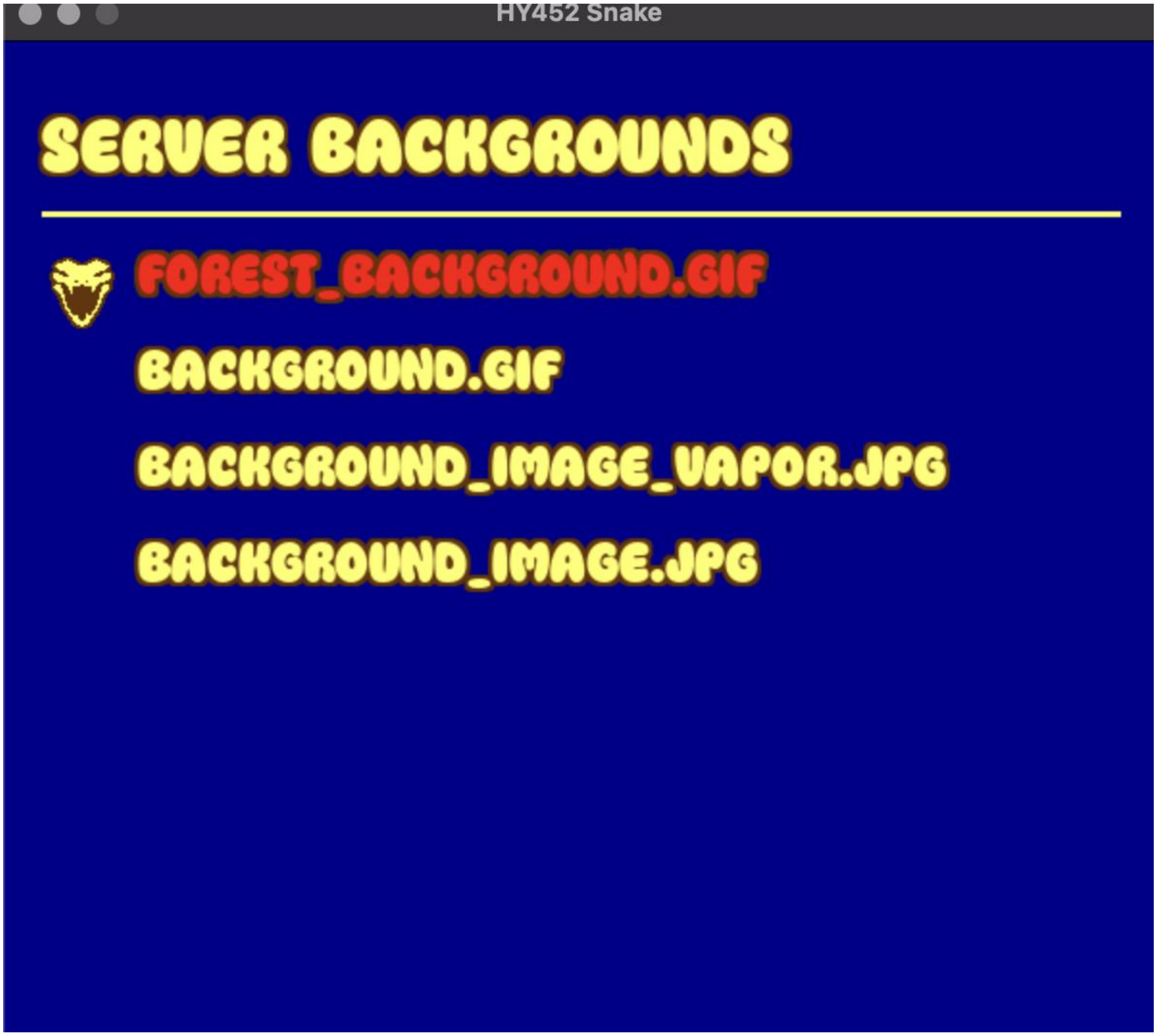


LEADERBOARD (LAST: 23:15:05)

POSITION	PLAYER	SCORE
1	PINELOPI	7
2	GIORGOS	7
3	PANOKLIS	15

Part 3 – Backgrounds

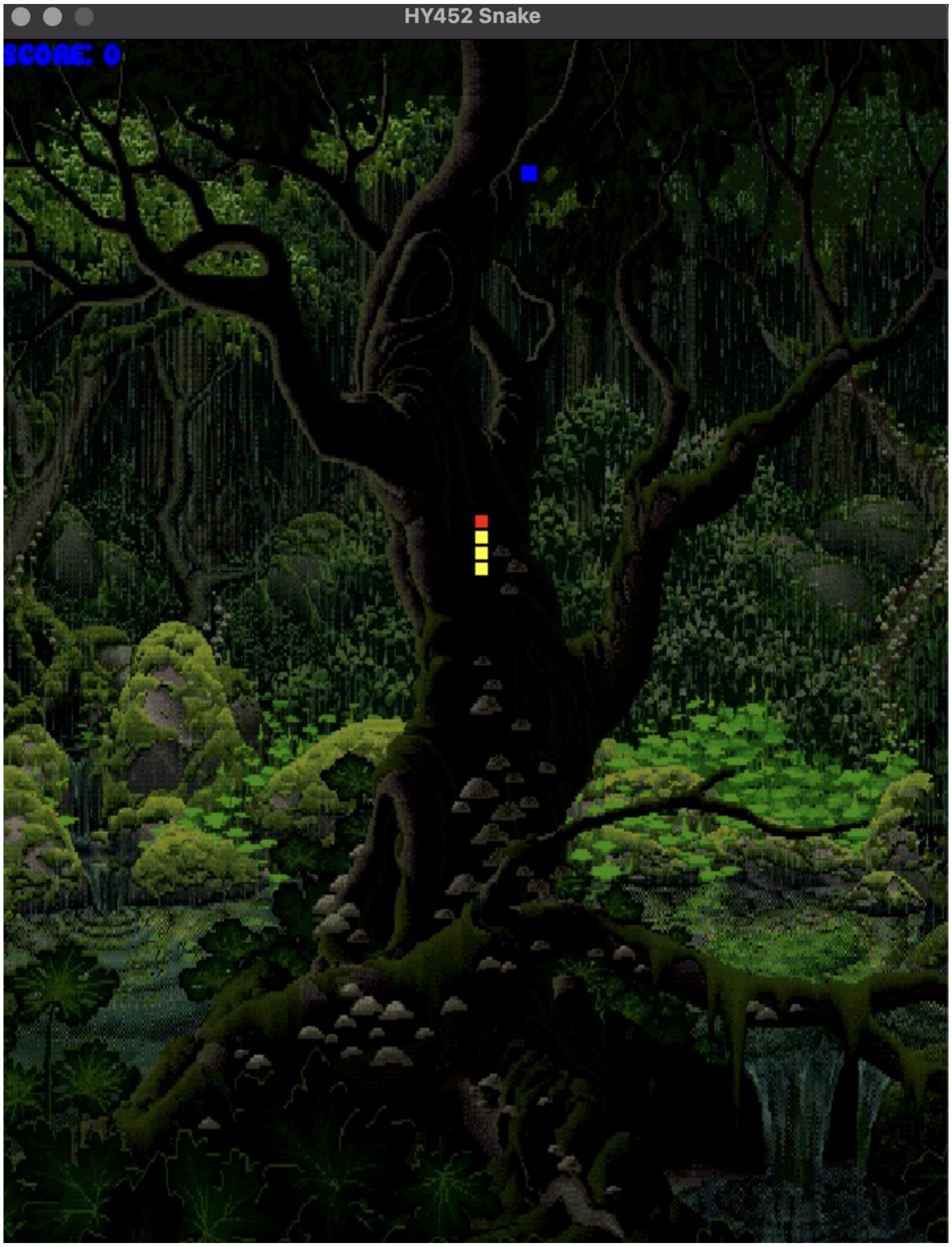
The user has more than the default background to choose from . Navigating the game from ‘Settings’ -> ‘Customize’ -> ‘Custom Background’ we arrive on the follow screen :



SERVER BACKGROUNDS

- FOREST_BACKGROUND.GIF
- BACKGROUND.GIF
- BACKGROUND_IMAGE_VAPOR.JPG
- BACKGROUND_IMAGE.JPG

Lets choose the first option . Now the background in game is different as demonstrated below .



These were all of the services I worked on for this project using AWS services !

Hope you enjoyed!