THANK YOU!

Q&A, easy questions please <3

DM2008 FINAL PROJECT PRESENTATION

MATH PYRAMID

- WONG KANG HONG

Background and Drive behind the project

WE LIVE IN A HIGHLY-DIGITALIZED WORLD TODAY, WHERE EVERY CHILD IS EASILY GIVEN ACCESS TO THE INTERNET. BUILDING ON THE INNATE COMPETITIVE NATURE OF THESE KIDS, AN ONLINE GAME THAT ENTICES AND INTRIGUES THE YOUTHFUL MIND WILL BRING ABOUT MANY BENEFITS.

THOUGH IN THE DEMO I MOSTLY USE MATH QUESTIONS, I WANT TO EMPHASIZE THAT QUESTIONS FROM OTHER SUBJECTS CAN BE USED AS WELL (THIS IS ALSO DEMONSTRATED LATER)

Concept :

TAKING INSPIRATION FROM CHILDHOOD GAMES OF MINE (E.G. TALES-RUNNER), I HAVE BUILT A 3D-PLATFORM GAME **MEANT TO ENCOURAGE KIDS** TO BE QUICK ON THEIR FEET (PUN INTENDED AS IT IS A RACING GAME) **WHEN IT COMES TO MATHEMATICAL COMPUTATIONS**

THE GAME IS HOSTED ON A PUBLIC DNS, ALLOWING FOR **EVERYONE AND ANYONE FROM ALL OVER THE WORLD** TO JOIN THE SAME GAME ROOM (SIMILAR TO THAT LIKE KAHOOT, BUT MORE **VISUALLY DYNAMIC AND ENGAGING**).

Why I chose to use these Technologies

Unity Game Engine

UNITY IS A CROSS-PLATFORM GAME ENGINE DEVELOPED BY UNITY TECHNOLOGIES, FIRST ANNOUNCED AND RELEASED IN JUNE 2005. THE ENGINE SUPPORTS A VARIETY OF DESKTOP, MOBILE, CONSOLE AND VIRTUAL REALITY PLATFORMS AND HAS BEEN USED FOR POPULAR GAMES SUCH AS POKEMON-GO AND CALL-OF-DUTY MOBILE.

HENCE, UNITY NOT ONLY PROVIDES HUGE SUPPORT AND RELIABILITY ACROSS ITS LARGE CODEBASE AND USERS, IT ALSO ALLOWS FOR PROJECT SCALABILITY.

Amazon Web Services

AWS IS A COMPREHENSIVE CLOUD COMPUTING PLATFORM THAT OFFERS SCALABLE SOLUTIONS FOR COMPUTE, STORAGE, DATABASES, ANALYTICS, AND MORE. BEING THE CURRENT MARKET LEADER IN THE CLOUD SERVICES MARKET, AWS HAS SHOWN TO HAVE PROVIDED RELIABLE SERVICES WHEN IT COMES TO THE CLOUD.

BY HOSTING OUR GAME VIA CLOUD, AWS NOT ONLY ALLOWS OUR GAME TO BE ACCESSIBLE ANYWHERE ACROSS THE WORLD, BUT ALSO ALLOWS FOR THE HYPER-SCALING OF OUR GAME APPLICATION, ACCOMODATING FOR THE POSSIBLITIES OF SUDDEN INFLUX IN USERS WHILE HAVING IMBUED, NATIVE SECURITY MEASURES IN PLACE.

Object Orientated Programming

BUILDING ON THE TECHNOLOGIES TAUGHT IN CLASS, OOP IS THE ARCHITECTURE THAT HOLDS EVERYTHING TOGETHER WHEN IT COMES TO THE COMMUNICATION OF ENTITIES WITHIN UNITY. FROM, ENCAPSULATION, INHERITANCE, COMPOSITION TO POLYMORPHISM, THE SCRIPTS ARE CODED WITH THESE CONCEPTS IN MIND.

Horizontally Scaling Workloads

AS AFOREMENTIONED, OUR GAME'S CLOUD INFRASTRUCTURE IS DESIGNED AND PROVISIONED WITH THE EMPLOYMENT OF AUTO-SCALING GROUPS, ALLOWING OUR GAME TO ACCOMMODATE TO A GROWING PLAYER-BASE AUTOMATICALLY WHILE KEEPING THE GAME RUNNING AT TIP-TOP CONDITIONS WITHOUT LOSING USER-EXPERIENCE.

AWS SETUP

within each Virtual Machine:

WE FIRST SPIN UP AN AWS VIRTUAL MACHINE RUNNING UBUNTU 18.04 LTS OS . FOLLOWING WHICH, WE INSTALL AN APACHE WEB SERVER ON TOP OF IT

APACHE HTTP SERVER (IS A FREE AND OPEN-SOURCE CROSS-PLATFORM WEB SERVER SOFTWARE) ALLOWS US TO HOST, PLAY AND COMMUNICATE WITH OUR GAME (BUILT ON WEBGL) OVER THE HTTP PORT 80/8080.

TO SIMPLY PUT, WE CONFIGURE APACHE TO POINT TO THE "BUILD" FOLDER OF OUR UNITY GAME, AND WHEN WE ACCESS THE VIRTUAL MACHINE VIA ITS IP ADDRESS OVER PORT 80/8080, WE WILL BE ABLE TO ACCESS THE GAME

```
GNU nano 2.9.3                                                    /etc/a

<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        #DocumentRoot /var/www/html
        DocumentRoot /var/www/Build9

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for pa
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error
        CustomLog ${APACHE_LOG_DIR}/acce

        # For most configuration files fro
        # enabled or disabled at a global le
        # include a line for only one parti
        # following line enables the CGI co
        # after it has been globally disabl
        #Include conf-available/serve-cgi-b
</VirtualHost>

  vim: syntax=apache ts=4 sw=4 sts=4 sr noe
```

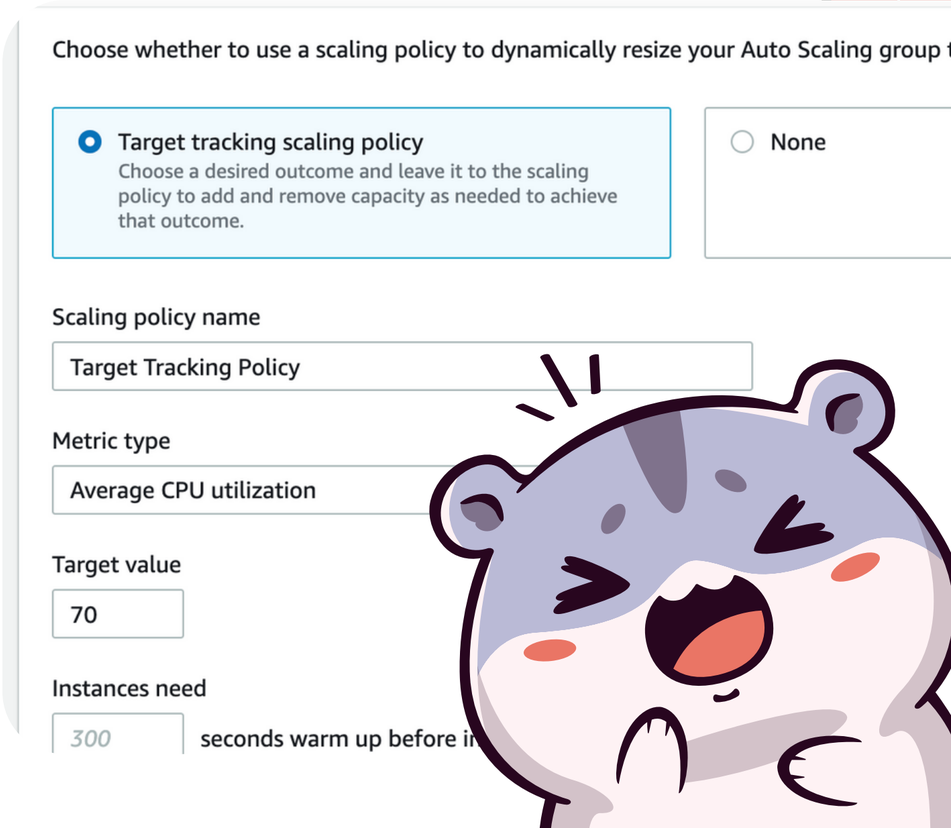
AWS SETUP

Scaling each Virtual Machine:

FOLLOWING WHICH, WE CREATE AN AMAZON MACHINE IMAGE FROM OUR RUNNING VIRTUAL MACHINE. AN AMI CAPTURES ALL THE CODE WITHIN THE MACHINE AT THAT POINT OF INSTANCE (IN CONTEXT, OUR UNITY GAME AND APACHE CONFIGURATIONS).

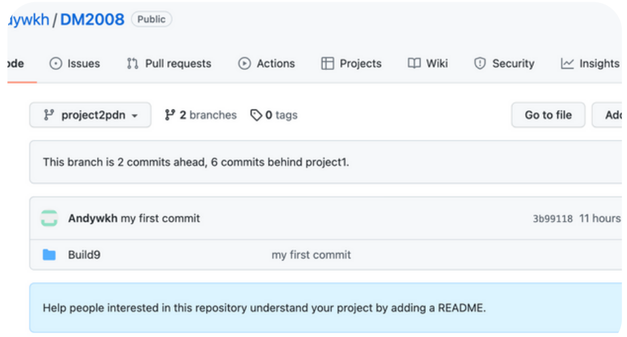
WE THEN CREATE A LAUNCH TEMPLATE TO TELL AWS WHAT VM CONFIGURATIONS (E.G. CPU, AMI) WE DESIRE WHEN WE INTEND TO SPIN UP NEW INSTANCES. A LAUNCH TEMPLATE ALLOWS US TO TEMPLATIZED OUR AWS VM DEPLOYMENTS (SIMILAR TO THAT OF UNITY'S PREFAB).

LASTLY, WE CREATE AN AUTO-SCALING GROUP, ATTACHED TO A LOAD BALANCER, BASED OFF THE LAUNCH TEMPLATE PREVIOUSLY CREATED. WE CONFIGURE THE AUTO-SCALING GROUP TO SPIN NEW INSTANCES WHEN THE WORKLOAD REACHES 70% CPU USAGE. TO ELABORATE, WHEN VM 1 REACHES A CPU USAGE OF 70%, AWS WILL SPIN UP ANOTHER VM 2 AND SIMULTANEOUSLY ROUTE TRAFFIC TO THE VMS AUTOMATICALLY.



Introduction to GitOps

Our Cloud Infrastructure employs GitOps to bring about effective communication and continuous integration of new features to our game



MODERN INFRASTRUCTURE NEEDS TO BE ELASTIC SO THAT IT CAN EFFECTIVELY MANAGE CLOUD RESOURCES THAT ARE NEEDED FOR CONTINUOUS DEPLOYMENTS. GITOPS IS USED TO AUTOMATE THE PROCESS OF PROVISIONING INFRASTRUCTURE.

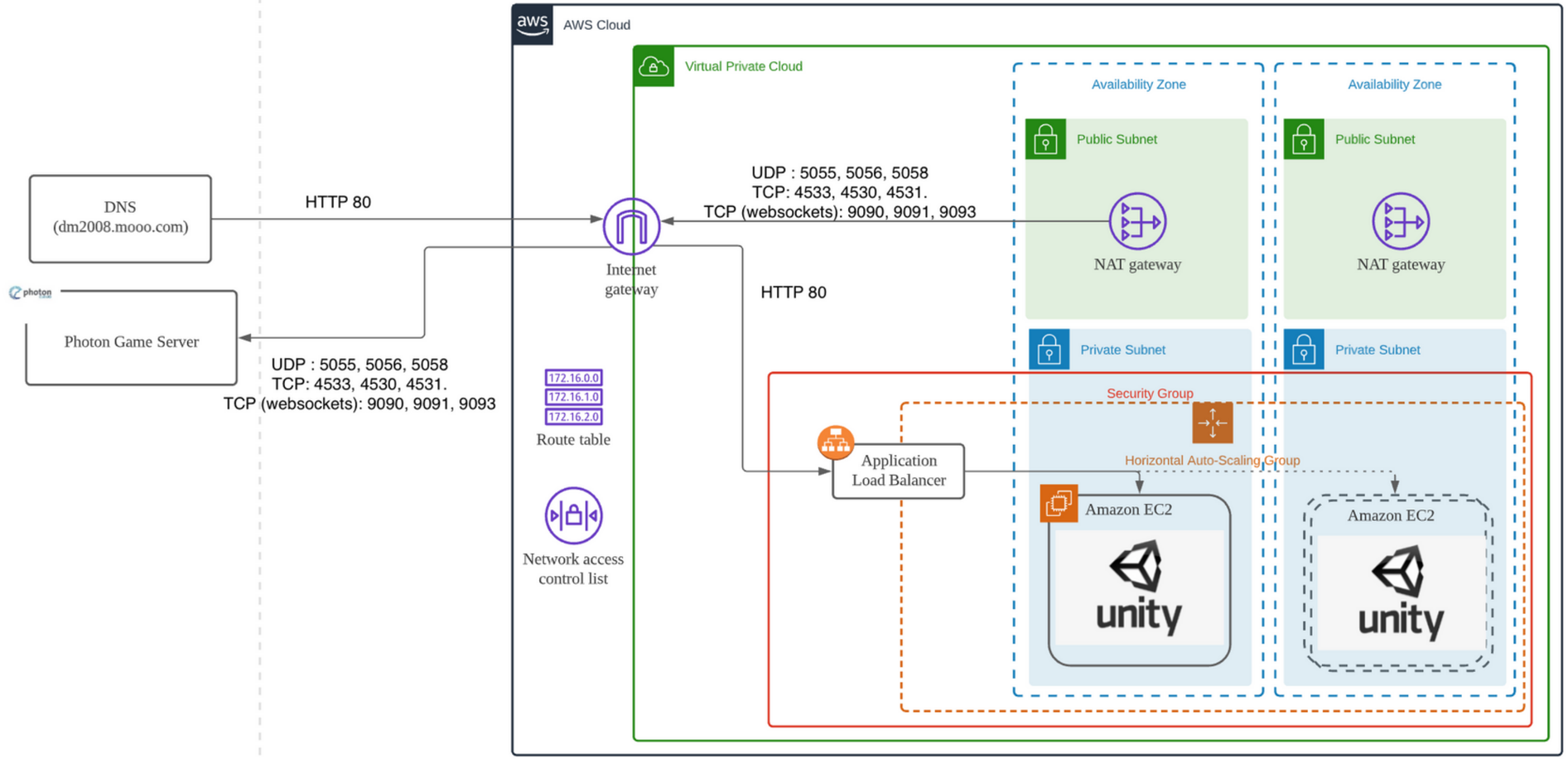
WITHOUT DIVING MUCH INTO THE INTRICACIES OF SHELL SCRIPTING, THE IDEA IS FOR GAME DEVELOPERS TO PUSH NEW BUILDS OF THE GAME INTO THE GITHUB REPOSITORY. AFTER WHICH, AWS IS SCHEDULED TO AUTOMATICALLY PULL THE NEWEST VERSION OF THE GAME FROM THE REPOSITORY, ALLOWING FOR THE HOSTING OF NEWEST VERSION OF THE GAME DAILY.

```
ong@Wongs-MacBook-Air-2 Build10 % git remote add origin
te origin already exists.
ong@Wongs-MacBook-Air-2 Build10 % git push -u origin proj
 objects: 19, done.
jects: 100% (19/19), done.
ession using up to 4 threads
 objects: 100% (19/19), done.
ects: 100% (19/19), 33.34 MiB | 2.03 MiB/s, done.
elta 0), reused 0 (delta 0), pack-reused 0

ate a pull request for 'project2dev' on GitHub by visiting
  https://github.com/Andywkh/DM2008/pull/new/project2dev

github.com/Andywkh/DM2008.git
nch]      project2dev -> project2dev
ect2dev' set up to track remote branch 'project2dev' fror
```

HIGH-LEVEL ARCHITECTURE



Demo

NO ENEMIES: **35.175.176.50**

PRODUCTION URL: **DM2008.MOOO.COM - (3.219.130.117)**

DEMO (VIDEO) [REACHING THE FINISHLINE]

CREDIBILITY FROM A CLOUD PROFESSIONAL

Here's a Recap:

(Not just a game, but a game that is hyper-scaled on the cloud)

MANY STUDIES HAVE SHOWN THAT GAMIFICATION OF ACTIVITIES LEADS TO COGNITIVE DEVELOPMENT IN ADOLESCENTS. THIS GAME IS THEREFORE PURPOSED TO HELP IMPROVE THE RATE AT WHICH THE BRAIN PROCESSES AND MAINTAINS INFORMATION.

IN ADDITION, THE MULTIPLAYER ASPECT IS ALSO A KEY COMPONENT OF THE GAME. STUDIES HAVE SHOWN THAT MULTIPLAYER GAMES PROMOTE TEAMWORK AND BUILD CONFIDENCE. THIS SHARED EXPERIENCE IS, THEREFORE, A GREAT WAY TO ALLOW PLAYERS TO COLLABORATE AND LEARN FROM EACH OTHER BRINGING ABOUT POSITIVE COMPETITIVE SPIRIT.

LASTLY, AS ELABORATED, OUR GAME'S CLOUD INFRASTRUCTURE IS DESIGNED AND PROVISIONED WITH THE EMPLOYMENT OF AUTO-SCALING GROUPS AND A PUBLIC URL TIED TO A DNS. THIS ALLOWS OUR GAME TO ACCOMMODATE TO A GROWING PLAYER-BASE AUTOMATICALLY WHILE KEEPING THE GAME RUNNING AT TIP-TOP CONDITIONS WITHOUT LOSING USER-EXPERIENCE. IT IS ALSO IMPORTANT TO NOTE THAT THE EASY TO MEMORIZE PUBLIC URL WILL BRING ABOUT A RECURRING PLAYER-BASE.

Prominent challenges faced and how I went about solving them

Challenge 1

With respect to Unity Game Engine, the hardest component and challenge faced was the integration of multiplayer. As a new player is meant to only control a new object instantiated from the character prefab, controls cannot be hard-coded into the character (when a new player joins, old characters are automatically disabled as all users are controlling the newest character).

Challenge 2

With respect to AWS cloud services, the hardest component and challenge faced was the design and implementation of security compliances (how to design a cloud infrastructure that encompasses security as anyone can use the cloud, but a good infrastructure must be hardened).

Solution 1

The solution integrated was such that within the player character instantiation script, the script creates a unique tag. and the player control searches for this tag and binds the movement controls to it. Hence, all player control scripts are uniquely acting upon their own object.

Solution 2

The final cloud infrastructure designed encompasses the services of an internet gateway, NAT gateways, NACL, and a routing table. The opening of ports and communication protocols between the cloud services took time and requires a deep understanding of the software and libraries required (e.g. ports listed in the architectural diagram for the integration of multiplayer via photon cloud servers.

WHAT CAN BE IMPROVED IF GIVEN MORE TIME, AND HOW I WOULD GO ABOUT DOING THEM

- IMPLEMENT MORE LEVELS (MULTI-LEVEL PYRAMID)
- USE SSL CERTIFICATES (HTTP(S) INTEGRATION)
- RANDOMIZED QUESTIONS FROM A DATABASE