# City Builder Game

team #31 Code Connoisseurs
partner - Project: Human City

## Introducing the Problem

- The Problem: How do we get people to go outside and interact with their city?

- The Solution: Make a game that rewards players for interacting with their city.
    - Finding different kinds of real world objects
    - Maintaining a game city that utilizes principles of sustainable development
    - Teach players about pollution and environmentally friendly solutions

## About The Game

- Web-based city-building game
- Start all from scratch
- Acheived the MVP of the game
- Future development: Link with AR camera for real-world resource gathering, diverse building options.

# Partner Introduction

Project: Human City

- Non-profit organization
- Focus on human inequality, social injustice and basic needs
- Initiatives for a more inclusive and equitable world

Organization Leader:

- James Rhule
- jamesrhule@projecthumancity.com
- Leading initiatives for global impact and fostering collaborations for social change.

Collaborative Members:

- Cheng-Ming Hsu: Spotstitch AR Camera Lead
- Dushyant Mehul Lunechiya: Frontend Product Lead
- Anupama Kadambi: Backend Development Lead
- Ali Hassan Amin: Spotstitch Frontend Developer

# Target Users – Understanding Diverse Player Types

Entertainment Enthusiasts: Enjoy building citis from scratch for fun.(Current Foucs)

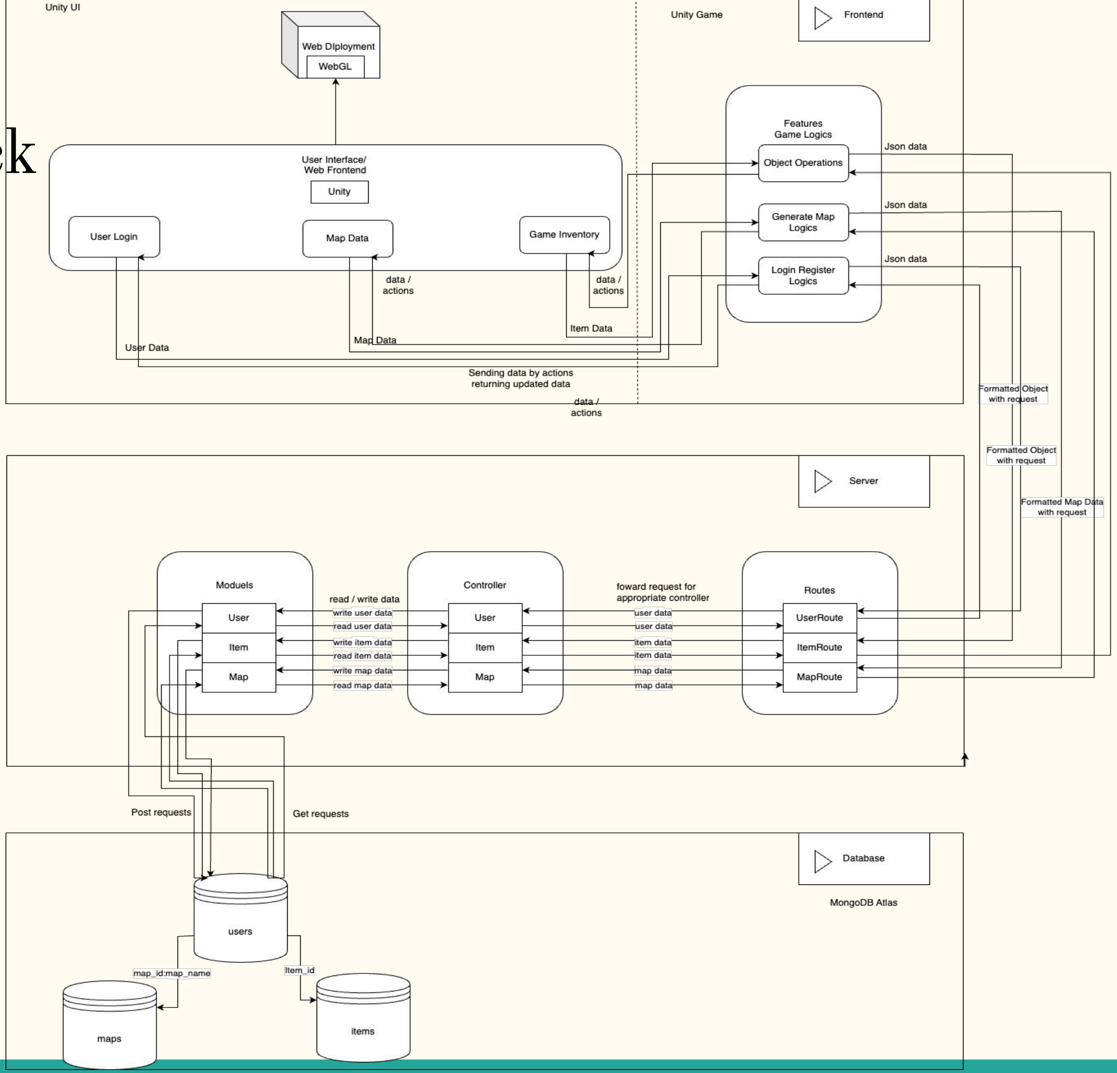
Professional City Designers: Utilize the game as a professional design tool.

Spotstich App(App already developed by our partner) Users: Engage for socializing within the Spotstitch community

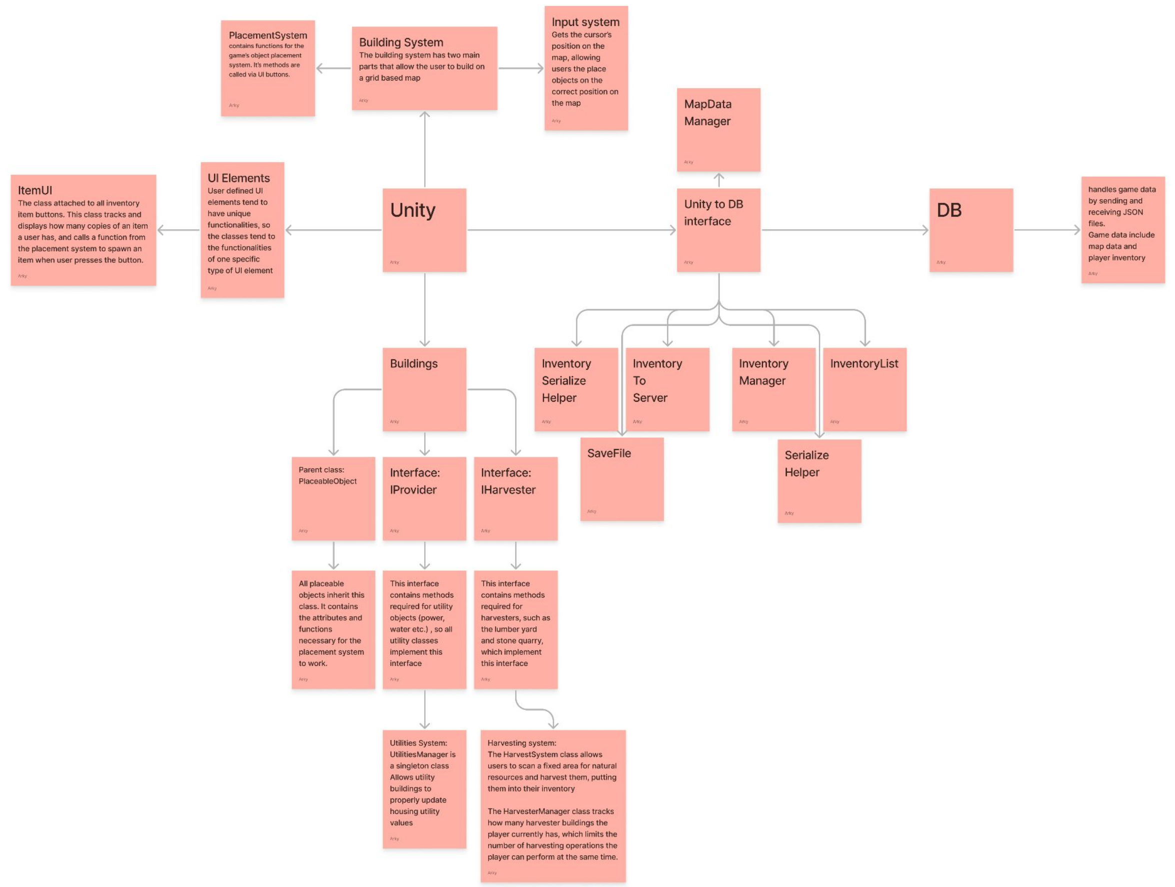
Demo

## Architecture & Tech Stack

- Frontend
  - Unity
- Server
  - Node.js
  - React
- Database
  - Mongo DB Atlas



# Unity Architecture Diagram



**PlacementSystem**
contains functions for the game's object placement system. It's methods are called via UI buttons.
Arky

**Building System**
The building system has two main parts that allow the user to build on a grid based map
Arky

**Input system**
Gets the cursor's position on the map, allowing users the place objects on the correct position on the map
Arky

**MapData Manager**
Arky

**ItemUI**
The class attached to all inventory item buttons. This class tracks and displays how many copies of an item a user has, and calls a function from the placement system to spawn an item when user presses the button.
Arky

**UI Elements**
User defined UI elements tend to have unique functionalities, so the classes tend to the functionalities of one specific type of UI element
Arky

**Unity**
Arky

**Unity to DB interface**
Arky

**DB**
Arky

handles game data by sending and receiving JSON files.
Game data include map data and player inventory
Arky

**Buildings**
Arky

**Inventory Serialize Helper**
Arky

**Inventory To Server**
Arky

**Inventory Manager**
Arky

**InventoryList**
Arky

**SaveFile**
Arky

**Serialize Helper**
Arky

**Parent class: PlaceableObject**
Arky

**Interface: IProvider**
Arky

**Interface: IHarvester**
Arky

All placeable objects inherit this class. It contains the attributes and functions necessary for the placement system to work.
Arky

This interface contains methods required for utility objects (power, water etc.) , so all utility classes implement this interface
Arky

This interface contains methods required for harvesters, such as the lumber yard and stone quarry, which implement this interface
Arky

Utilities System: UtilitiesManager is a singleton class Allows utility buildings to properly update housing utility values
Arky

Harvesting system:
The HarvestSystem class allows users to scan a fixed area for natural resources and harvest them, putting them into their inventory

The HarvesterManager class tracks how many harvester buildings the player currently has, which limits the number of harvesting operations the player can perform at the same time.
Arky

## Why Use MongoDB, Node.js, and Unity?

Mongodb:

- Partner Collaboration
- Uniformity and Intergration

Nodejs:

- Speed and Scalability
- Efficient Server Management
- Seamless Integration with MongoDB

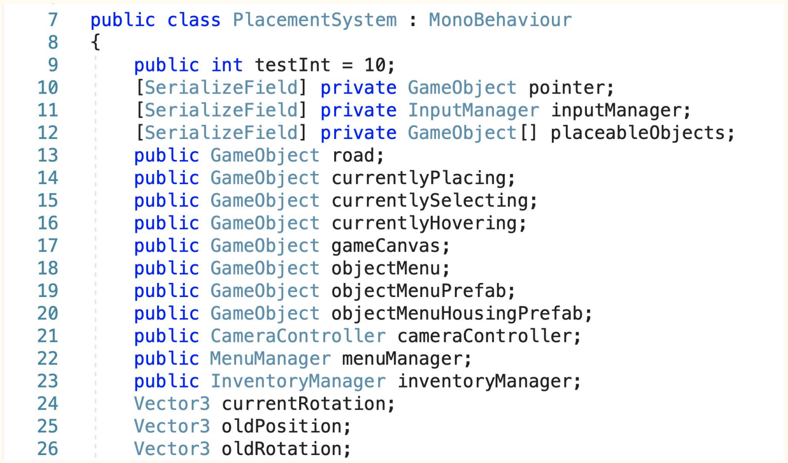
Unity:

- Partner Requirement
- Support for High-End Graphics
- Experienced Team Members

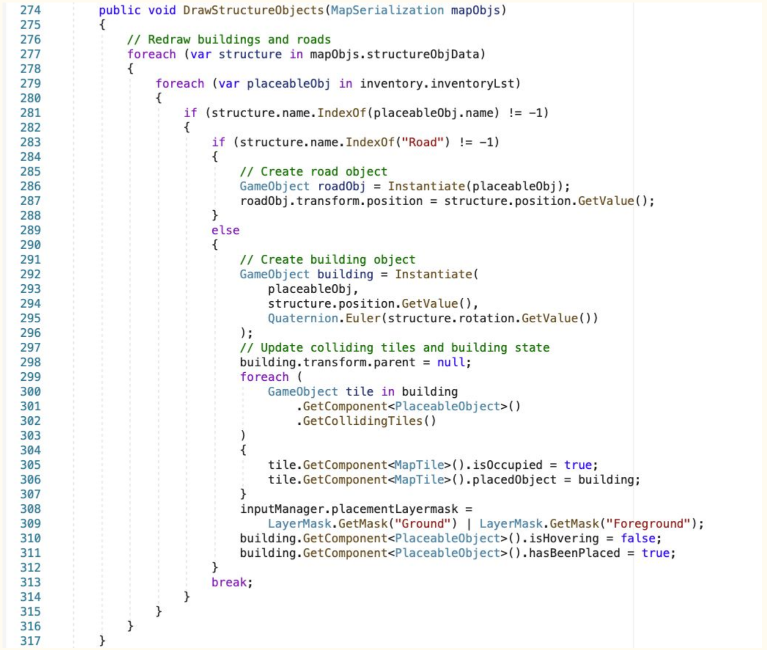
# Technical Discussion

## Coding styles & Clean Code Practices:

- **coding and naming conventions**
    - Google C# Style Guide, descriptive naming



```csharp
public class PlacementSystem : MonoBehaviour
{
    public int testInt = 10;
    [SerializeField] private GameObject pointer;
    [SerializeField] private InputManager inputManager;
    [SerializeField] private GameObject[] placeableObjects;
    public GameObject road;
    public GameObject currentlyPlacing;
    public GameObject currentlySelecting;
    public GameObject currentlyHovering;
    public GameObject gameCanvas;
    public GameObject objectMenu;
    public GameObject objectMenuPrefab;
    public GameObject objectMenuHousingPrefab;
    public CameraController cameraController;
    public MenuManager menuManager;
    public InventoryManager inventoryManager;
    Vector3 currentRotation;
    Vector3 oldPosition;
    Vector3 oldRotation;
```

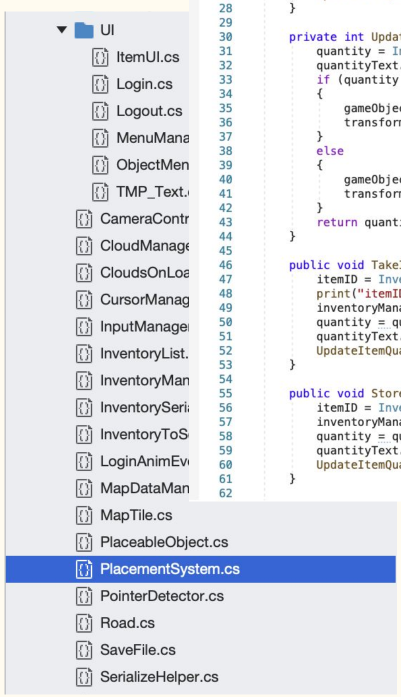


```csharp
public void DrawStructureObjects(MapSerialization mapObjs)
{
    // Redraw buildings and roads
    foreach (var structure in mapObjs.structureObjData)
    {
        foreach (var placeableObj in inventory.inventoryLst)
        {
            if (structure.name.IndexOf(placeableObj.name) != -1)
            {
                if (structure.name.IndexOf("Road") != -1)
                {
                    // Create road object
                    GameObject roadObj = Instantiate(placeableObj);
                    roadObj.transform.position = structure.position.GetValue();
                }
                else
                {
                    // Create building object
                    GameObject building = Instantiate(
                        placeableObj,
                        structure.position.GetValue(),
                        Quaternion.Euler(structure.rotation.GetValue())
                    );
                    // Update colliding tiles and building state
                    building.transform.parent = null;
                    foreach (
                        GameObject tile in building
                            .GetComponent<PlaceableObject>()
                            .GetCollidingTiles()
                    )
                    {
                        tile.GetComponent<MapTile>().isOccupied = true;
                        tile.GetComponent<MapTile>().placedObject = building;
                    }
                    inputManager.placementLayermask =
                        LayerMask.GetMask("Ground") | LayerMask.GetMask("Foreground");
                    building.GetComponent<PlaceableObject>().isHovering = false;
                    building.GetComponent<PlaceableObject>().hasBeenPlaced = true;
                }
                break;
            }
        }
    }
}
```

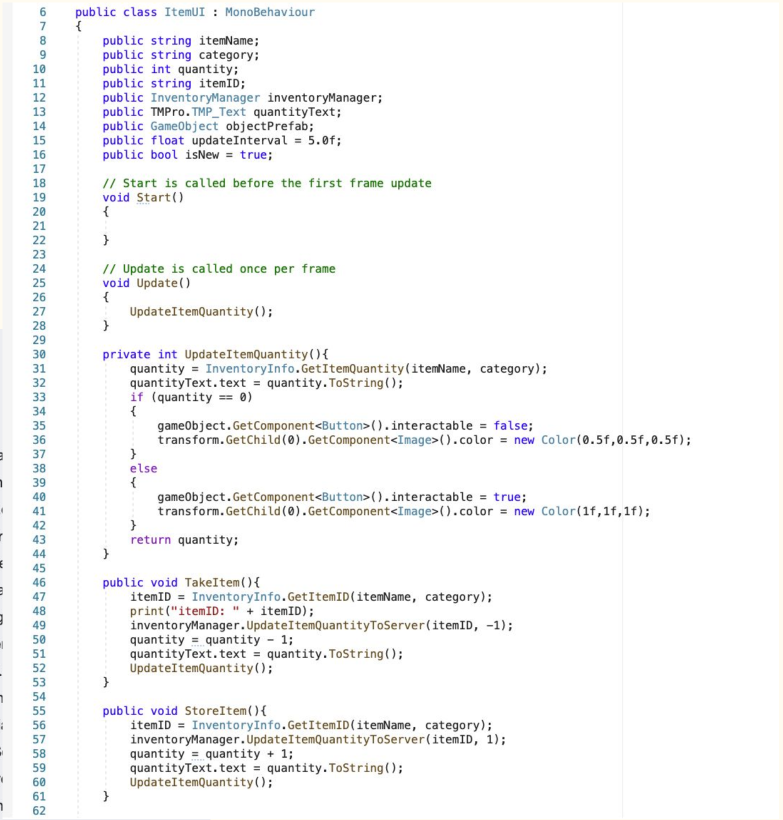
# Technical Discussion

## Coding styles & Clean Code Practices:

- **Modularity**
  - breakdown into modular files &classes
  - clear responsibilities for each functions



- UI
  - ItemUI.cs
  - Login.cs
  - Logout.cs
  - MenuMana
  - ObjectMen
  - TMP_Text.
- CameraContr
- CloudManage
- CloudsOnLoa
- CursorManag
- InputManage
- InventoryList.
- InventoryMan
- InventorySeri
- InventoryToS
- LoginAnimEv
- MapDataMan
- MapTile.cs
- PlaceableObject.cs
- PlacementSystem.cs
- PointerDetector.cs
- Road.cs
- SaveFile.cs
- SerializeHelper.cs



```csharp
public class ItemUI : MonoBehaviour
{
    public string itemName;
    public string category;
    public int quantity;
    public string itemID;
    public InventoryManager inventoryManager;
    public TMPro.TMP_Text quantityText;
    public GameObject objectPrefab;
    public float updateInterval = 5.0f;
    public bool isNew = true;

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        UpdateItemQuantity();
    }

    private int UpdateItemQuantity(){
        quantity = InventoryInfo.GetItemQuantity(itemName, category);
        quantityText.text = quantity.ToString();
        if (quantity == 0)
        {
            gameObject.GetComponent<Button>().interactable = false;
            transform.GetChild(0).GetComponent<Image>().color = new Color(0.5f,0.5f,0.5f);
        }
        else
        {
            gameObject.GetComponent<Button>().interactable = true;
            transform.GetChild(0).GetComponent<Image>().color = new Color(1f,1f,1f);
        }
        return quantity;
    }

    public void TakeItem(){
        itemID = InventoryInfo.GetItemID(itemName, category);
        print("itemID: " + itemID);
        inventoryManager.UpdateItemQuantityToServer(itemID, -1);
        quantity = quantity - 1;
        quantityText.text = quantity.ToString();
        UpdateItemQuantity();
    }

    public void StoreItem(){
        itemID = InventoryInfo.GetItemID(itemName, category);
        inventoryManager.UpdateItemQuantityToServer(itemID, 1);
        quantity = quantity + 1;
        quantityText.text = quantity.ToString();
        UpdateItemQuantity();
    }

```

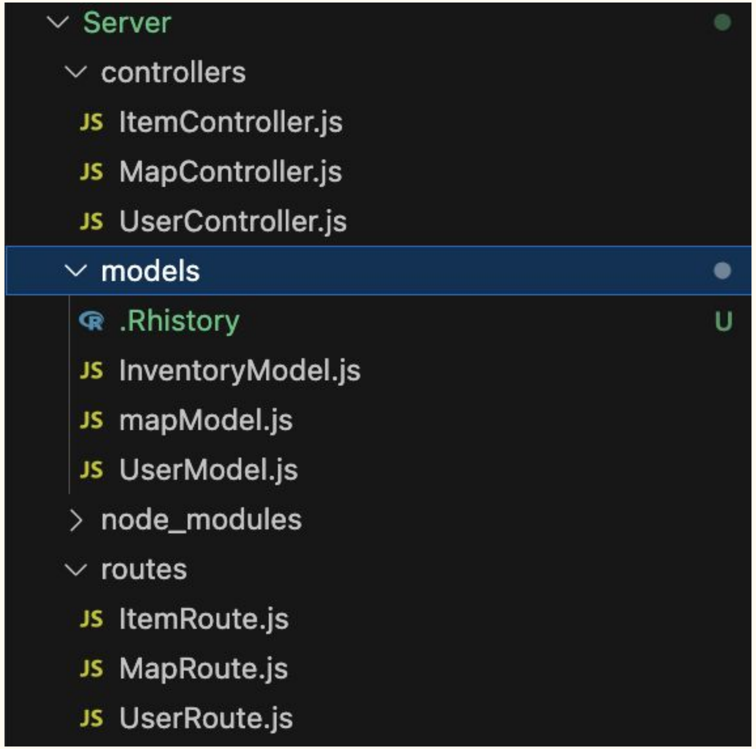
# Technical Discussion

**Coding styles & Clean Code Practices:**

- **Modularity**
    - breakdown into modular files &classes
    - clear responsibilities for each functions

e.g: For server part using Node.js

- models: define schema for mongodb
    - map, user, inventory
- controllers: functions that handle different requests
    - map, user, inventory
- routes: provides API endpoint to trigger functions in controller
    - map, user, inventory

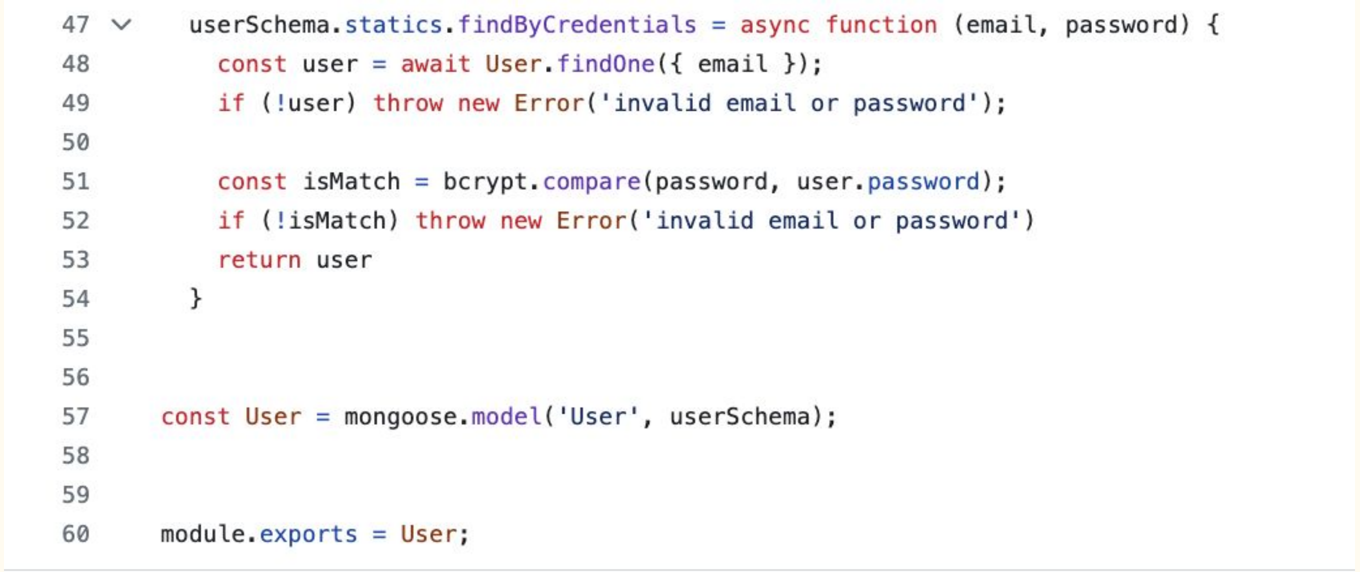


```
Server
  controllers
    ItemController.js
    MapController.js
    UserController.js
  models
    .Rhistory                U
    InventoryModel.js
    mapModel.js
    UserModel.js
  node_modules
  routes
    ItemRoute.js
    MapRoute.js
    UserRoute.js
```

## Technical Discussion

**Coding styles & Clean Code Practices:**

- **Error Handling:**

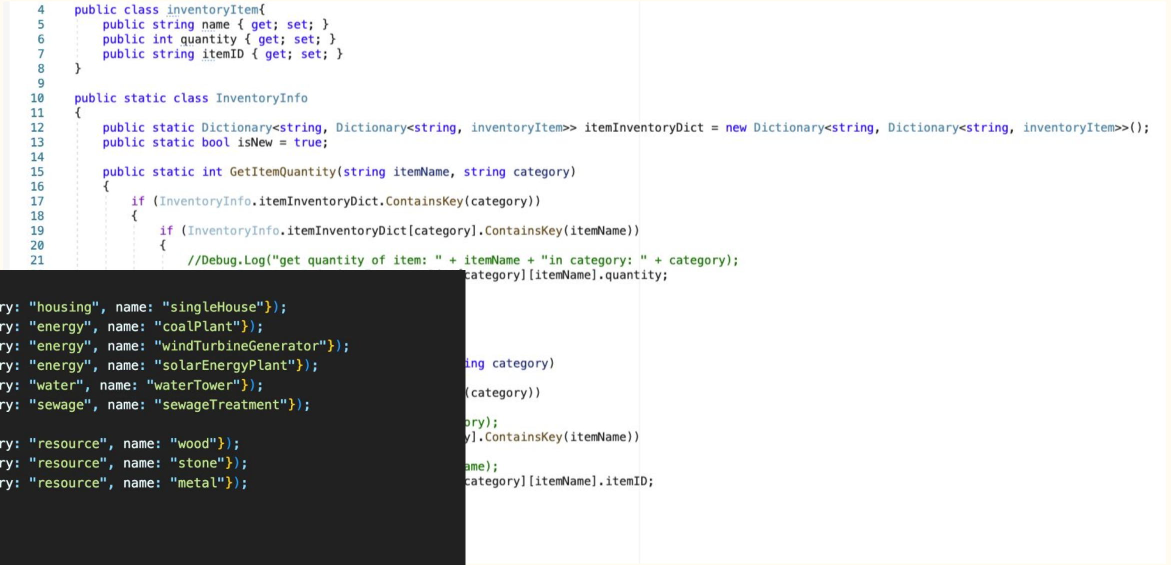


```
  userSchema.statics.findByCredentials = async function (email, password) {
    const user = await User.findOne({ email });
    if (!user) throw new Error('invalid email or password');

    const isMatch = bcrypt.compare(password, user.password);
    if (!isMatch) throw new Error('invalid email or password')
    return user
  }


const User = mongoose.model('User', userSchema);


module.exports = User;
```

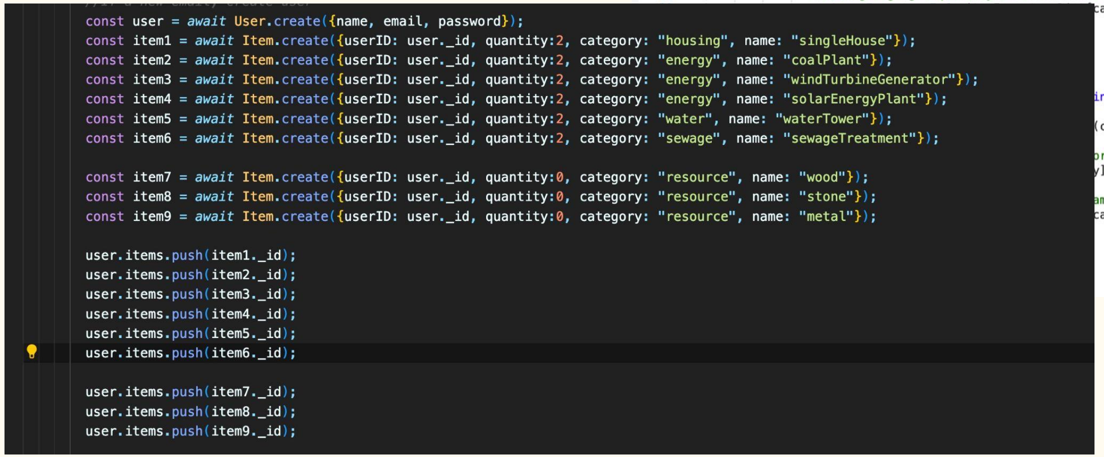
# Technical Discussion

## Coding styles & Clean Code Practices:

- **Optimization**
    - e.g.
      expand inventory easily



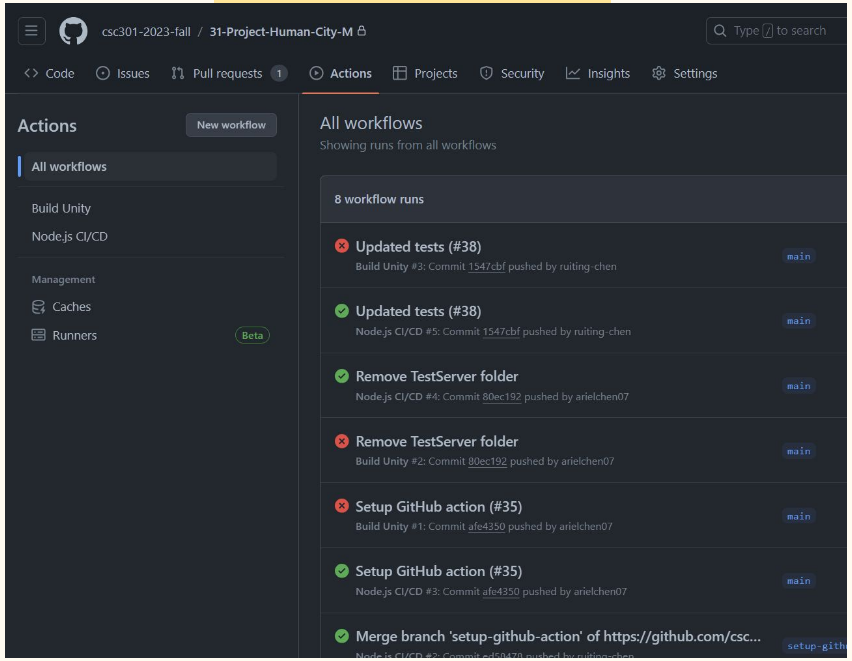
```
public class inventoryItem{
    public string name { get; set; }
    public int quantity { get; set; }
    public string itemID { get; set; }
}

public static class InventoryInfo
{
    public static Dictionary<string, Dictionary<string, inventoryItem>> itemInventoryDict = new Dictionary<string, Dictionary<string, inventoryItem>>();
    public static bool isNew = true;

    public static int GetItemQuantity(string itemName, string category)
    {
        if (InventoryInfo.itemInventoryDict.ContainsKey(category))
        {
            if (InventoryInfo.itemInventoryDict[category].ContainsKey(itemName))
            {
                //Debug.Log("get quantity of item: " + itemName + "in category: " + category);
                                                                   category][itemName].quantity;





                                                                   ing category)

                                                                   (category))

                                                                   ory);
                                                                   y].ContainsKey(itemName))

                                                                   ame);
                                                                   category][itemName].itemID;
```



```
const user = await User.create({name, email, password});
const item1 = await Item.create({userID: user._id, quantity:2, category: "housing", name: "singleHouse"});
const item2 = await Item.create({userID: user._id, quantity:2, category: "energy", name: "coalPlant"});
const item3 = await Item.create({userID: user._id, quantity:2, category: "energy", name: "windTurbineGenerator"});
const item4 = await Item.create({userID: user._id, quantity:2, category: "energy", name: "solarEnergyPlant"});
const item5 = await Item.create({userID: user._id, quantity:2, category: "water", name: "waterTower"});
const item6 = await Item.create({userID: user._id, quantity:2, category: "sewage", name: "sewageTreatment"});

const item7 = await Item.create({userID: user._id, quantity:0, category: "resource", name: "wood"});
const item8 = await Item.create({userID: user._id, quantity:0, category: "resource", name: "stone"});
const item9 = await Item.create({userID: user._id, quantity:0, category: "resource", name: "metal"});

user.items.push(item1._id);
user.items.push(item2._id);
user.items.push(item3._id);
user.items.push(item4._id);
user.items.push(item5._id);
user.items.push(item6._id);

user.items.push(item7._id);
user.items.push(item8._id);
user.items.push(item9._id);
```

# Deployment

GitHub Actions



csc301-2023-fall / 31-Project-Human-City-M

All workflows
Showing runs from all workflows

8 workflow runs

Updated tests (#38)
Build Unity #3: Commit 1547cbf pushed by ruiting-chen — main

Updated tests (#38)
Node.js CI/CD #5: Commit 1547cbf pushed by ruiting-chen — main

Remove TestServer folder
Node.js CI/CD #4: Commit 80ec192 pushed by arielchen07 — main

Remove TestServer folder
Build Unity #2: Commit 80ec192 pushed by arielchen07 — main

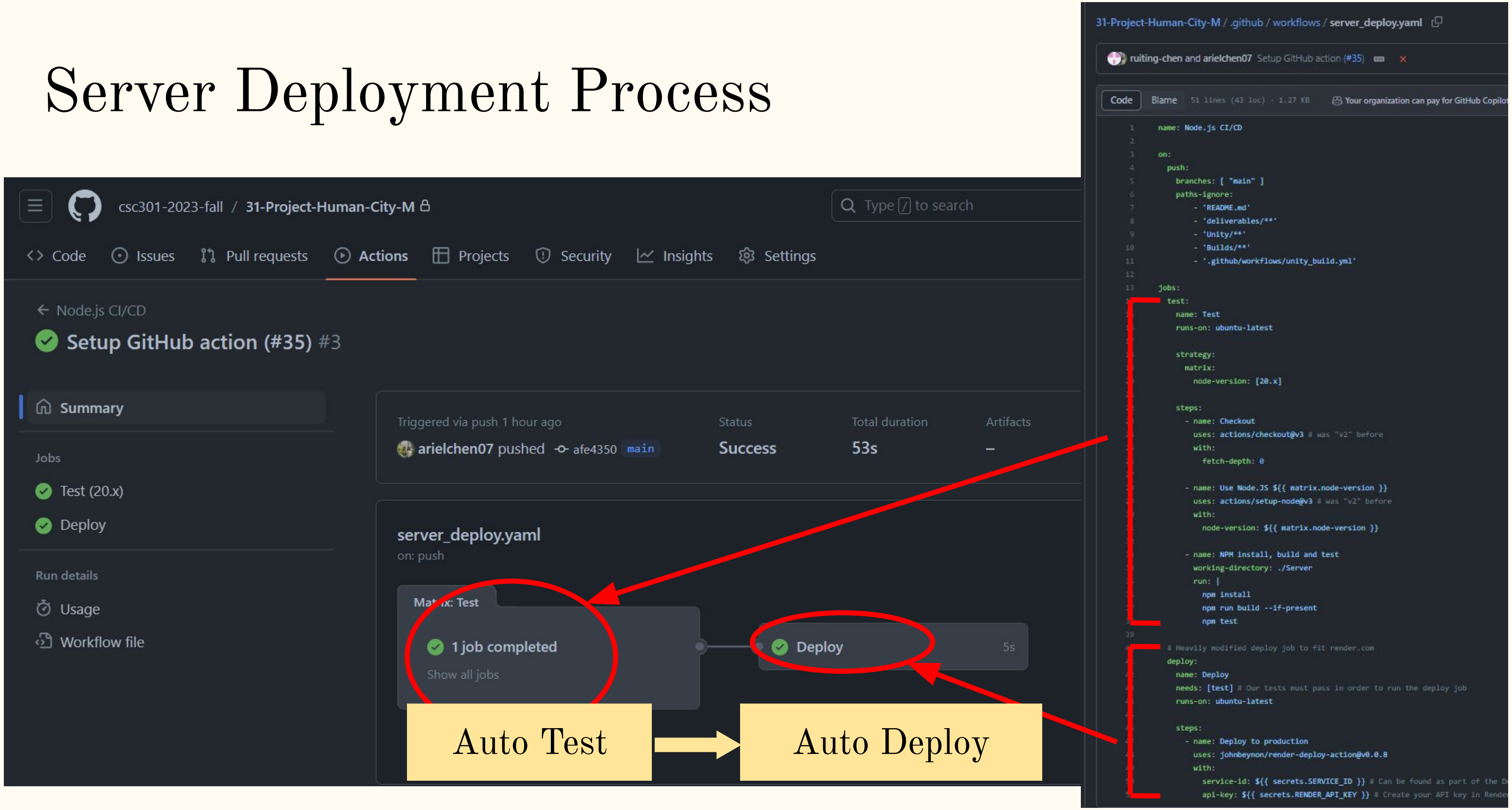
Setup GitHub action (#35)
Build Unity #1: Commit afe4350 pushed by arielchen07 — main

Setup GitHub action (#35)
Node.js CI/CD #3: Commit afe4350 pushed by arielchen07 — main

Merge branch 'setup-github-action' of https://github.com/csc...
Node.js CI/CD #2: Commit ed50470 pushed by ruiting-chen — setup-gith

- Triggered on each commit onto Main branch (ingnore some files ie. README updates)
- Automated testing, then deployment on success test
- Frontend (Unity): desktop exe
- Backend (server): Render

# Server Deployment Process



csc301-2023-fall / 31-Project-Human-City-M

Code · Issues · Pull requests · Actions · Projects · Security · Insights · Settings

← Node.js CI/CD

Setup GitHub action (#35) #3

Summary

Jobs
- Test (20.x)
- Deploy

Run details
- Usage
- Workflow file

Triggered via push 1 hour ago — arielchen07 pushed afe4350 main — Status: Success — Total duration: 53s — Artifacts: –

server_deploy.yaml
on: push

Matrix: Test — 1 job completed — Show all jobs

Deploy 5s

31-Project-Human-City-M / .github / workflows / server_deploy.yaml

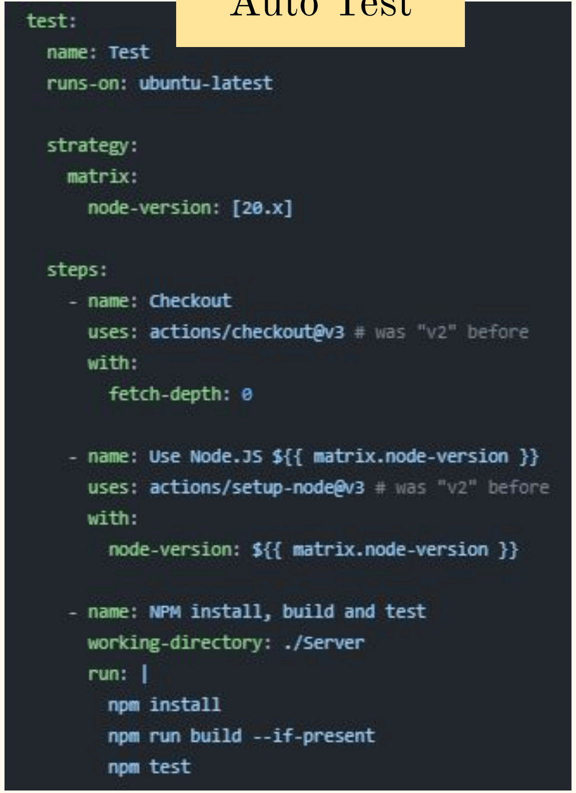
ruiting-chen and arielchen07 Setup GitHub action (#35)

Code Blame 51 lines (43 loc) · 1.27 KB Your organization can pay for GitHub Copilot

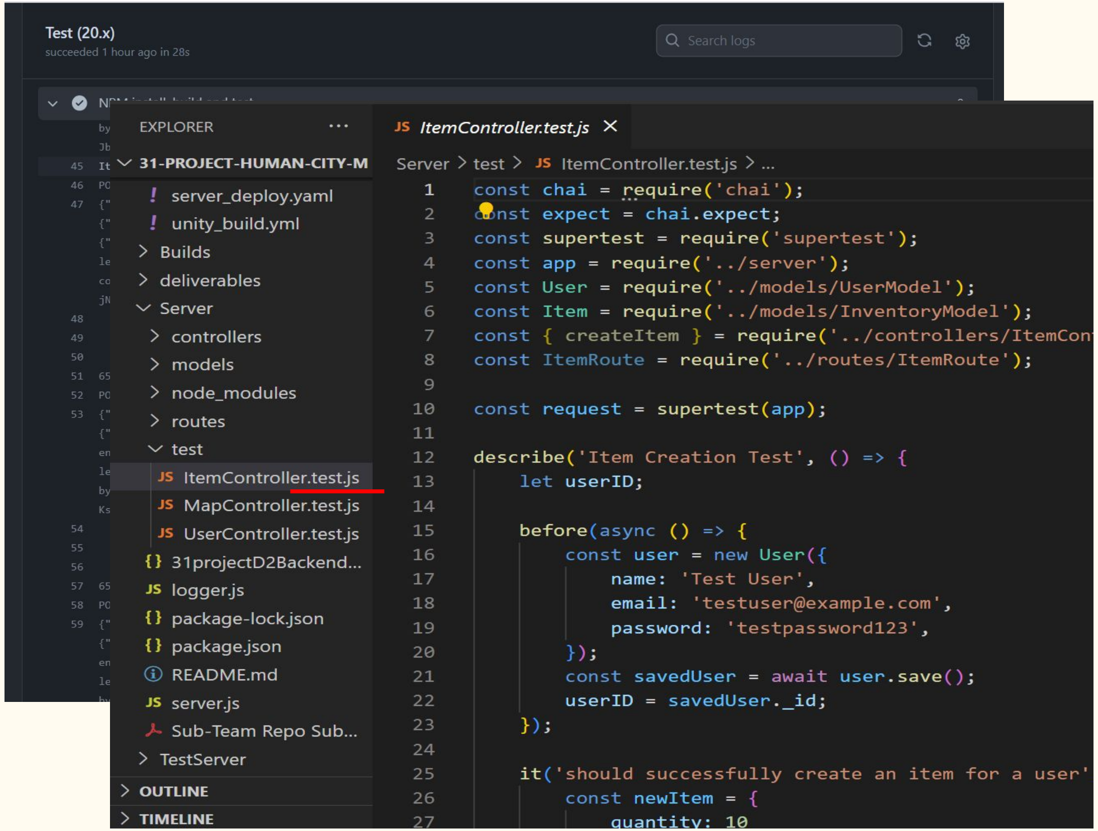
```yaml
name: Node.js CI/CD

on:
  push:
    branches: [ "main" ]
    paths-ignore:
        - 'README.md'
        - 'deliverables/**'
        - 'Unity/**'
        - 'Builds/**'
        - '.github/workflows/unity_build.yml'

jobs:
  test:
    name: Test
    runs-on: ubuntu-latest

    strategy:
      matrix:
        node-version: [20.x]

    steps:
      - name: Checkout
        uses: actions/checkout@v3 # was "v2" before
        with:
          fetch-depth: 0

      - name: Use Node.JS ${{ matrix.node-version }}
        uses: actions/setup-node@v3 # was "v2" before
        with:
          node-version: ${{ matrix.node-version }}

      - name: NPM install, build and test
        working-directory: ./Server
        run: |
          npm install
          npm run build --if-present
          npm test

  # Heavily modified deploy job to fit render.com
  deploy:
    name: Deploy
    needs: [test] # Our tests must pass in order to run the deploy job
    runs-on: ubuntu-latest

    steps:
      - name: Deploy to production
        uses: johnbeynon/render-deploy-action@v0.0.8
        with:
          service-id: ${{ secrets.SERVICE_ID }} # Can be found as part of the De
          api-key: ${{ secrets.RENDER_API_KEY }} # Create your API key in Rende
```

# Server Deployment Process

Auto Test



```yaml
test:
  name: Test
  runs-on: ubuntu-latest

  strategy:
    matrix:
      node-version: [20.x]

  steps:
    - name: Checkout
      uses: actions/checkout@v3 # was "v2" before
      with:
        fetch-depth: 0

    - name: Use Node.JS ${{ matrix.node-version }}
      uses: actions/setup-node@v3 # was "v2" before
      with:
        node-version: ${{ matrix.node-version }}

    - name: NPM install, build and test
      working-directory: ./Server
      run: |
        npm install
        npm run build --if-present
        npm test
```

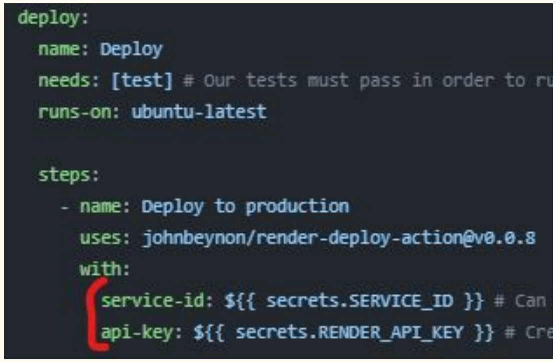


Test (20.x)
succeeded 1 hour ago in 28s

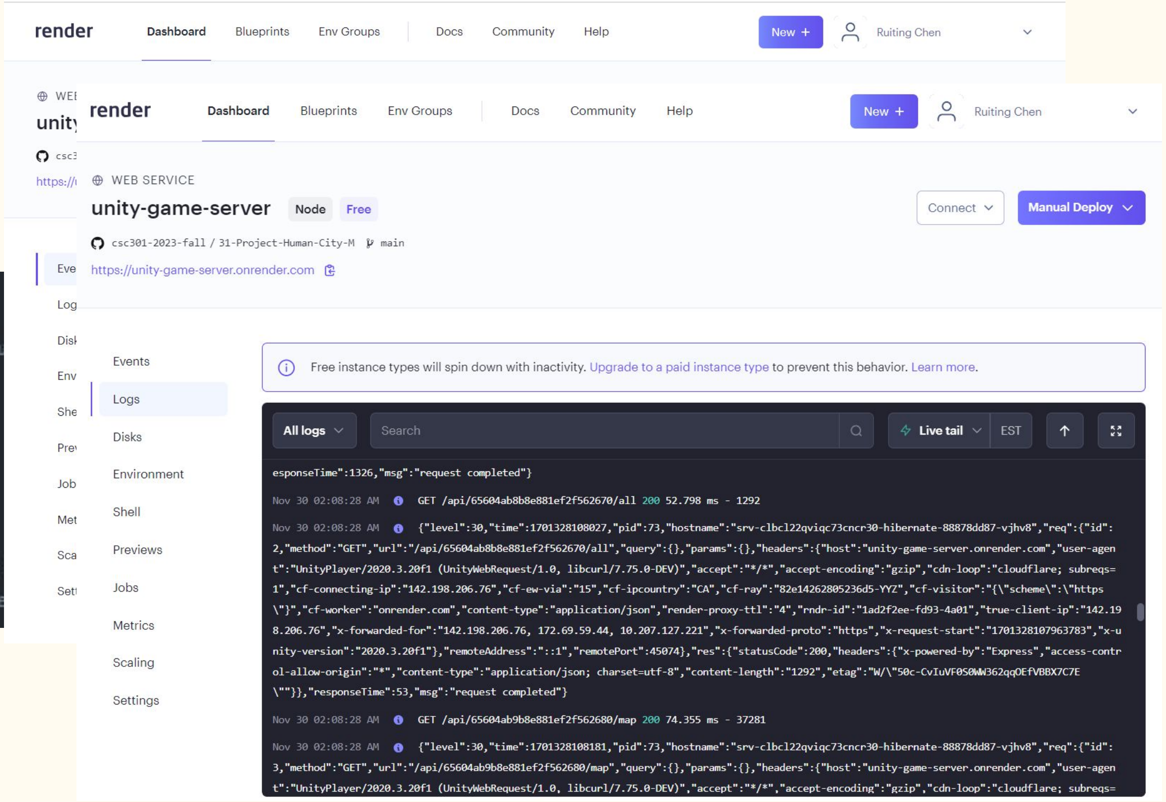
```javascript
const chai = require('chai');
const expect = chai.expect;
const supertest = require('supertest');
const app = require('../server');
const User = require('../models/UserModel');
const Item = require('../models/InventoryModel');
const { createItem } = require('../controllers/ItemCon
const ItemRoute = require('../routes/ItemRoute');

const request = supertest(app);

describe('Item Creation Test', () => {
    let userID;

    before(async () => {
        const user = new User({
            name: 'Test User',
            email: 'testuser@example.com',
            password: 'testpassword123',
        });
        const savedUser = await user.save();
        userID = savedUser._id;
    });

    it('should successfully create an item for a user'
        const newItem = {
            quantity: 10
```

# Server Deployment Process

Auto Deploy



```
deploy:
  name: Deploy
  needs: [test] # Our tests must pass in order to ru
  runs-on: ubuntu-latest

  steps:
    - name: Deploy to production
      uses: johnbeynon/render-deploy-action@v0.0.8
      with:
        service-id: ${{ secrets.SERVICE_ID }} # Can
        api-key: ${{ secrets.RENDER_API_KEY }} # Cre
```



WEB SERVICE

unity-game-server Node Free

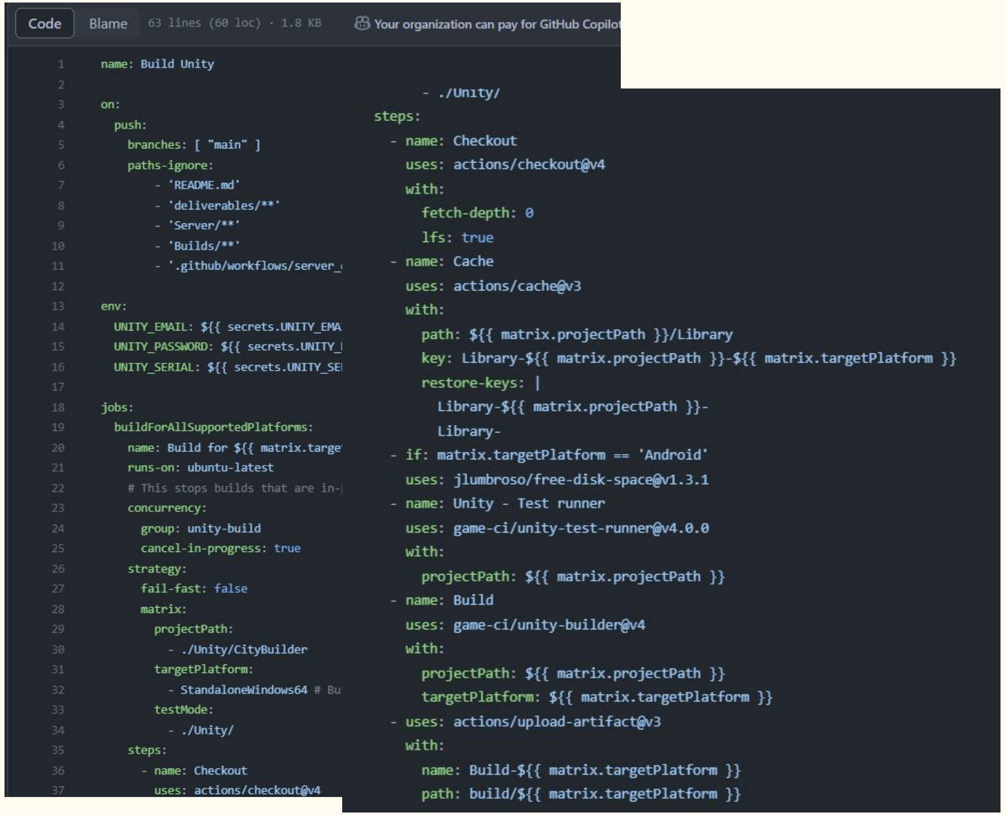
csc301-2023-fall / 31-Project-Human-City-M main

https://unity-game-server.onrender.com

Free instance types will spin down with inactivity. Upgrade to a paid instance type to prevent this behavior. Learn more.

```
esponseTime":1326,"msg":"request completed"}

Nov 30 02:08:28 AM  GET /api/65604ab8b8e881ef2f562670/all 200 52.798 ms - 1292

Nov 30 02:08:28 AM  {"level":30,"time":1701328108027,"pid":73,"hostname":"srv-clbcl22qviqc73cncr30-hibernate-88878dd87-vjhv8","req":{"id":
2,"method":"GET","url":"/api/65604ab8b8e881ef2f562670/all","query":{},"params":{},"headers":{"host":"unity-game-server.onrender.com","user-agen
t":"UnityPlayer/2020.3.20f1 (UnityWebRequest/1.0, libcurl/7.75.0-DEV)","accept":"*/*","accept-encoding":"gzip","cdn-loop":"cloudflare; subreqs=
1","cf-connecting-ip":"142.198.206.76","cf-ew-via":"15","cf-ipcountry":"CA","cf-ray":"82e14262805236d5-YYZ","cf-visitor":"{\"scheme\":\"https
\"}","cf-worker":"onrender.com","content-type":"application/json","render-proxy-ttl":"4","rndr-id":"1ad2f2ee-fd93-4a01","true-client-ip":"142.19
8.206.76","x-forwarded-for":"142.198.206.76, 172.69.59.44, 10.207.127.221","x-forwarded-proto":"https","x-request-start":"1701328107963783","x-u
nity-version":"2020.3.20f1"},"remoteAddress":"::1","remotePort":45074},"res":{"statusCode":200,"headers":{"x-powered-by":"Express","access-contr
ol-allow-origin":"*","content-type":"application/json; charset=utf-8","content-length":"1292","etag":"W/\"50c-CvIuVF0S0WW362qqOEfVBBX7C7E
\""}},"responseTime":53,"msg":"request completed"}

Nov 30 02:08:28 AM  GET /api/65604ab9b8e881ef2f562680/map 200 74.355 ms - 37281

Nov 30 02:08:28 AM  {"level":30,"time":1701328108181,"pid":73,"hostname":"srv-clbcl22qviqc73cncr30-hibernate-88878dd87-vjhv8","req":{"id":
3,"method":"GET","url":"/api/65604ab9b8e881ef2f562680/map","query":{},"params":{},"headers":{"host":"unity-game-server.onrender.com","user-agen
t":"UnityPlayer/2020.3.20f1 (UnityWebRequest/1.0, libcurl/7.75.0-DEV)","accept":"*/*","accept-encoding":"gzip","cdn-loop":"cloudflare; subreqs=
```

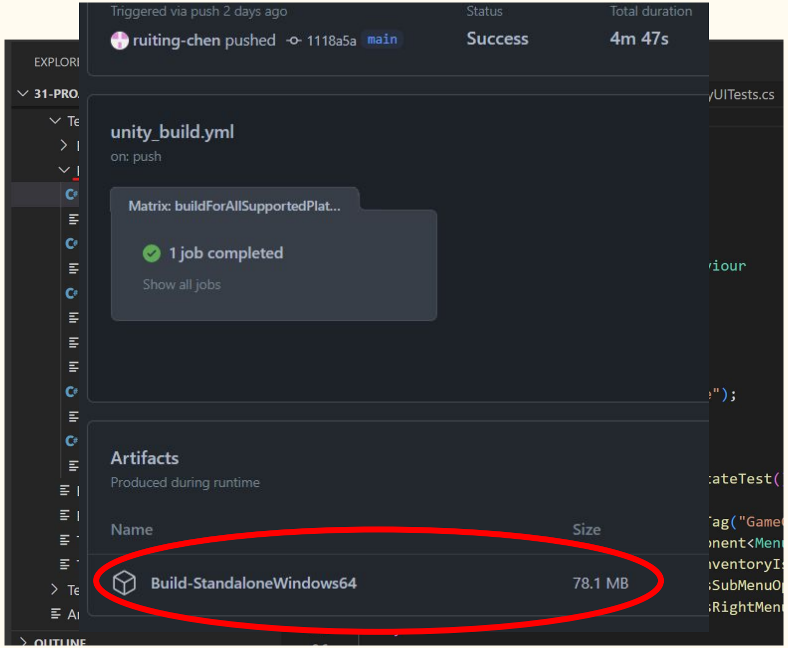
# Unity Deployment Process



Code
Blame
63 lines (60 loc) · 1.8 KB
Your organization can pay for GitHub Copilot

```yaml
name: Build Unity

on:
  push:
    branches: [ "main" ]
    paths-ignore:
        - 'README.md'
        - 'deliverables/**'
        - 'Server/**'
        - 'Builds/**'
        - '.github/workflows/server_

env:
  UNITY_EMAIL: ${{ secrets.UNITY_EMA
  UNITY_PASSWORD: ${{ secrets.UNITY_
  UNITY_SERIAL: ${{ secrets.UNITY_SE

jobs:
  buildForAllSupportedPlatforms:
    name: Build for ${{ matrix.targe
    runs-on: ubuntu-latest
    # This stops builds that are in-
    concurrency:
      group: unity-build
      cancel-in-progress: true
    strategy:
      fail-fast: false
      matrix:
        projectPath:
          - ./Unity/CityBuilder
        targetPlatform:
          - StandaloneWindows64 # Bu
        testMode:
          - ./Unity/
    steps:
      - name: Checkout
        uses: actions/checkout@v4
```

```yaml
          - ./Unity/
    steps:
      - name: Checkout
        uses: actions/checkout@v4
        with:
          fetch-depth: 0
          lfs: true
      - name: Cache
        uses: actions/cache@v3
        with:
          path: ${{ matrix.projectPath }}/Library
          key: Library-${{ matrix.projectPath }}-${{ matrix.targetPlatform }}
          restore-keys: |
            Library-${{ matrix.projectPath }}-
            Library-
      - if: matrix.targetPlatform == 'Android'
        uses: jlumbroso/free-disk-space@v1.3.1
      - name: Unity - Test runner
        uses: game-ci/unity-test-runner@v4.0.0
        with:
          projectPath: ${{ matrix.projectPath }}
      - name: Build
        uses: game-ci/unity-builder@v4
        with:
          projectPath: ${{ matrix.projectPath }}
          targetPlatform: ${{ matrix.targetPlatform }}
      - uses: actions/upload-artifact@v3
        with:
          name: Build-${{ matrix.targetPlatform }}
          path: build/${{ matrix.targetPlatform }}
```



Triggered via push 2 days ago — ruiting-chen pushed 1118a5a main

Status: Success

Total duration: 4m 47s

unity_build.yml
on: push

Matrix: buildForAllSupportedPlat...
1 job completed
Show all jobs

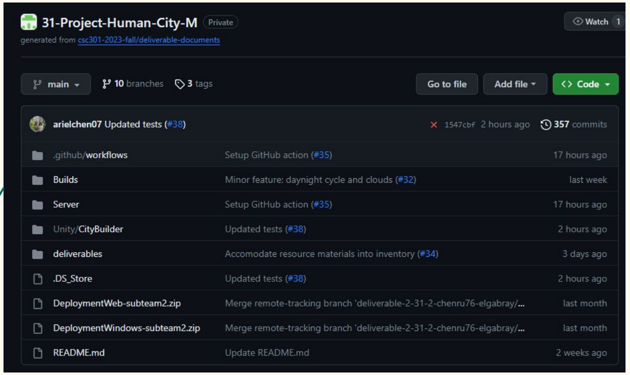
Artifacts
Produced during runtime

| Name | Size |
|---|---|
| Build-StandaloneWindows64 | 78.1 MB |

# Accessing application

## Hand-off plan:

- Clone repo to organization:
  - Code follows guidelines
  - Simplified deployment
  - README
- Upload documentation to drive:
  - Game Design Document
  - Technical Document
  - Architectural Diagrams
  - Demo video
- Deployment accounts by email

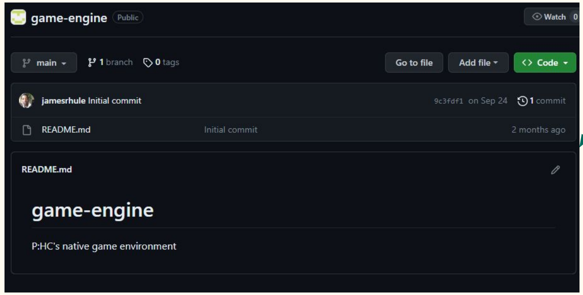


31-Project-Human-City-M Private
generated from csc301-2023-fall/deliverable-documents

main · 10 branches · 3 tags

arielchen07 Updated tests (#38) ✗ 1547cbf 2 hours ago 357 commits

| | | |
|---|---|---|
| .github/workflows | Setup GitHub action (#35) | 17 hours ago |
| Builds | Minor feature: daynight cycle and clouds (#32) | last week |
| Server | Setup GitHub action (#35) | 17 hours ago |
| Unity/CityBuilder | Updated tests (#38) | 2 hours ago |
| deliverables | Accomodate resource materials into inventory (#34) | 3 days ago |
| .DS_Store | Updated tests (#38) | 2 hours ago |
| DeploymentWeb-subteam2.zip | Merge remote-tracking branch 'deliverable-2-31-2-chenru76-elgabray/... | last month |
| DeploymentWindows-subteam2.zip | Merge remote-tracking branch 'deliverable-2-31-2-chenru76-elgabray/... | last month |
| README.md | Update README.md | 2 weeks ago |



game-engine Public

main · 1 branch · 0 tags

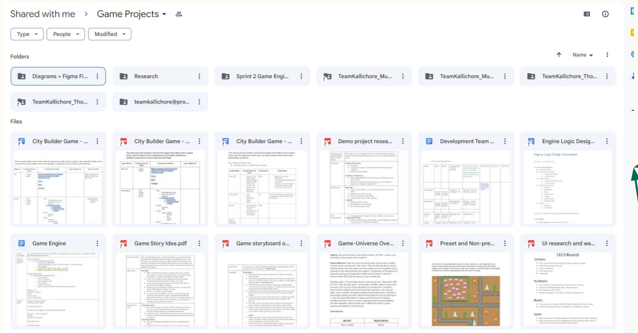
jamesrhule Initial commit 9c3fdf1 on Sep 24 1 commit

| | | |
|---|---|---|
| README.md | Initial commit | 2 months ago |

README.md

game-engine

P:HC's native game environment



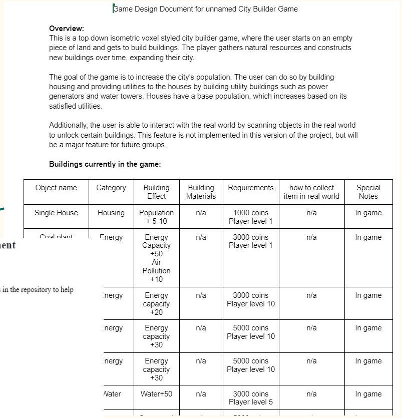
Shared with me > Game Projects

Folders

Diagrams + Figma Fi... · Research · Sprint 2 Game Engi... · TeamKallichore_Mu... · TeamKallichore_Mu... · TeamKallichore_Tho... · TeamKallichore_Tho... · teamkallichore@pro...

Files

City Builder Game - ... · City Builder Game - ... · City Builder Game - ... · Demo project resea... · Development Team ... · Engine Logic Desig... · Game Engine · Game Story Idea.pdf · Game storyboard o... · Game-Universe Ove... · Preset and Non-pre... · UI research and wa...



Game Design Document for unnamed City Builder Game

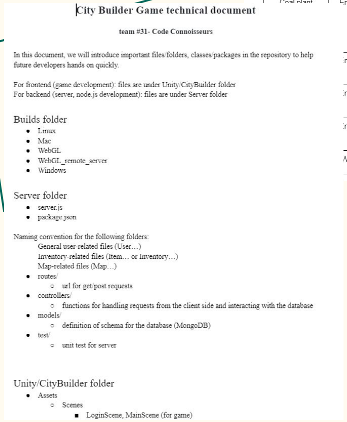
**Overview:**
This is a top down isometric voxel styled city builder game, where the user starts on an empty piece of land and gets to build buildings. The player gathers natural resources and constructs new buildings over time, expanding their city.

The goal of the game is to increase the city's population. The user can do so by building housing and providing utilities to the houses by building utility buildings such as power generators and water towers. Houses have a base population, which increases based on its satisfied utilities.

Additionally, the user is able to interact with the real world by scanning objects in the real world to unlock certain buildings. This feature is not implemented in this version of the project, but will be a major feature for future groups.

**Buildings currently in the game:**

| Object name | Category | Building Effect | Building Materials | Requirements | how to collect item in real world | Special Notes |
|---|---|---|---|---|---|---|
| Single House | Housing | Population + 5-10 | n/a | 1000 coins Player level 1 | n/a | In game |
| Coal plant | Energy | Energy Capacity +50 Air Pollution +10 | n/a | 3000 coins Player level 1 | n/a | In game |
| [illegible] | nergy | Energy capacity +20 | n/a | 3000 coins Player level 10 | n/a | In game |
| [illegible] | nergy | Energy capacity +30 | n/a | 5000 coins Player level 10 | n/a | In game |
| [illegible] | nergy | Energy capacity +30 | n/a | 5000 coins Player level 10 | n/a | In game |
| [illegible] | ater | Water+50 | n/a | 3000 coins Player level 5 | n/a | In game |



City Builder Game technical document

team #31- Code Connoisseurs

In this document, we will introduce important files/folders, classes/packages in the repository to help future developers hands on quickly.

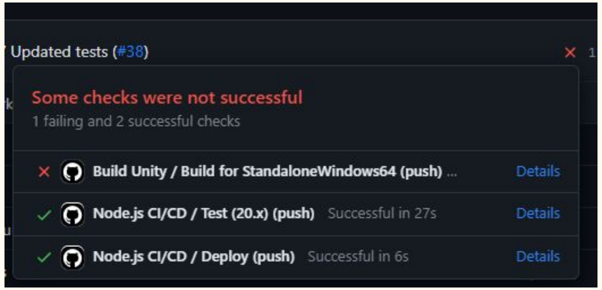
For frontend (game development): files are under Unity/CityBuilder folder
For backend (server, node.js development): files are under Server folder

Builds folder
- Linux
- Mac
- WebGL
- WebGL_remote_server
- Windows

Server folder
- server.js
- package.json

Naming convention for the following folders:
General user-related files (User…)
Inventory-related files (Item… or Inventory…)
Map-related files (Map…)
- routes/
  - url for get/post requests
- controllers/
  - functions for handling requests from the client side and interacting with the database
- models/
  - definition of schema for the database (MongoDB)
- test/
  - unit test for server

Unity/CityBuilder folder
- Assets
  - Scenes
    - LoginScene, MainScene (for game)



Updated tests (#38) ✗

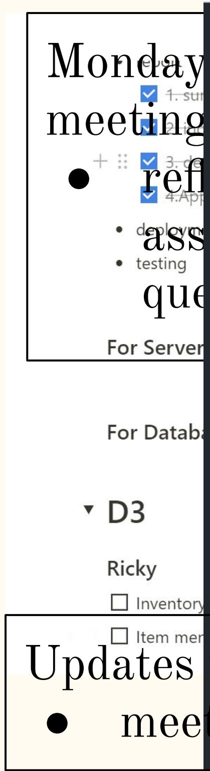
Some checks were not successful
1 failing and 2 successful checks

| | | |
|---|---|---|
| ✗ | Build Unity / Build for StandaloneWindows64 (push) ... | Details |
| ✓ | Node.js CI/CD / Test (20.x) (push) Successful in 27s | Details |
| ✓ | Node.js CI/CD / Deploy (push) Successful in 6s | Details |

## Working Process

Monday meeting:

- ref
- ass
- que



~~1. sur~~

~~3. de~~

4.App

- deployme
- testing

For Server

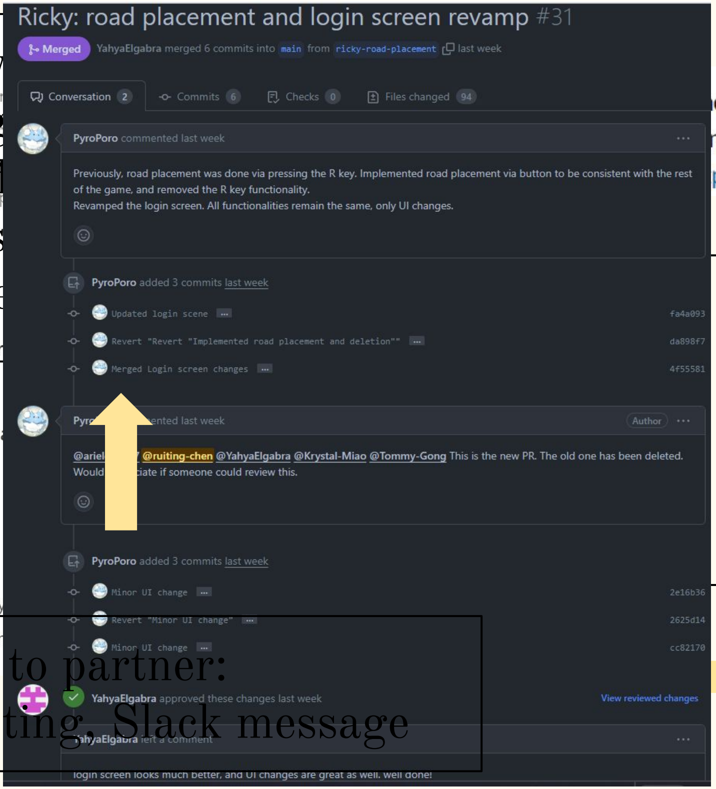
For Datab

▾ D3

Ricky

☐ Inventory

☐ Item mer



Ricky: road placement and login screen revamp #31

Merged YahyaElgabra merged 6 commits into `main` from `ricky-road-placement` last week

Conversation 2 | Commits 6 | Checks 0 | Files changed 94

PyroPoro commented last week

Previously, road placement was done via pressing the R key. Implemented road placement via button to be consistent with the rest of the game, and removed the R key functionality.
Revamped the login screen. All functionalities remain the same, only UI changes.

PyroPoro added 3 commits last week

```
Updated login scene                                    fa4a093
Revert "Revert "Implemented road placement and deletion""   da898f7
Merged Login screen changes                            4f55581
```

Pyro ented last week — Author

@ariel @ruiting-chen @YahyaElgabra @Krystal-Miao @Tommy-Gong This is the new PR. The old one has been deleted. Would iate if someone could review this.

PyroPoro added 3 commits last week

```
Minor UI change               2e16b36
Revert "Minor UI change"      2625d14
Minor UI change               cc82170
```

YahyaElgabra approved these changes last week — View reviewed changes

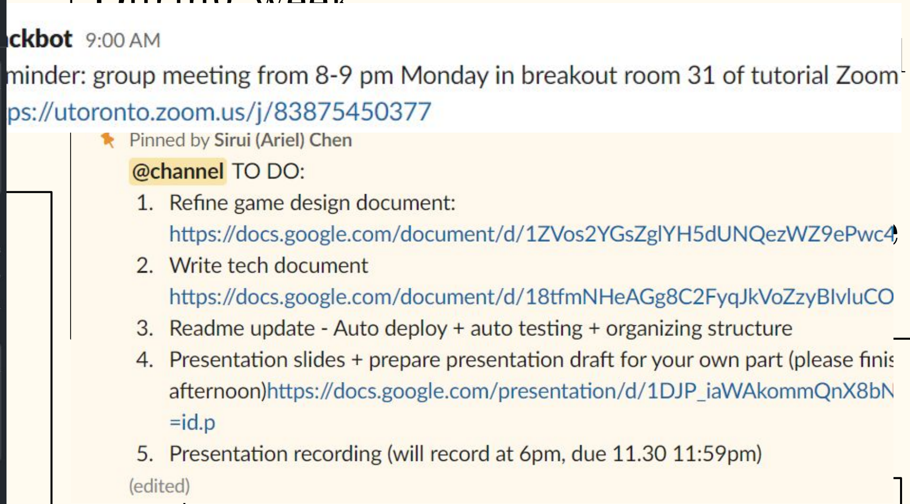
YahyaElgabra left a comment

login screen looks much better, and UI changes are great as well. well done!

Updates to partner:

- meeting, Slack message

During week:



ckbot 9:00 AM

minder: group meeting from 8-9 pm Monday in breakout room 31 of tutorial Zoom
ps://utoronto.zoom.us/j/83875450377

Pinned by Sirui (Ariel) Chen

@channel TO DO:

1. Refine game design document:
   https://docs.google.com/document/d/1ZVos2YGsZglYH5dUNQezWZ9ePwc4
2. Write tech document
   https://docs.google.com/document/d/18tfmNHeAGg8C2FyqJkVoZzyBlvluCO
3. Readme update - Auto deploy + auto testing + organizing structure
4. Presentation slides + prepare presentation draft for your own part (please finis afternoon)https://docs.google.com/presentation/d/1DJP_iaWAkommQnX8bN =id.p
5. Presentation recording (will record at 6pm, due 11.30 11:59pm)

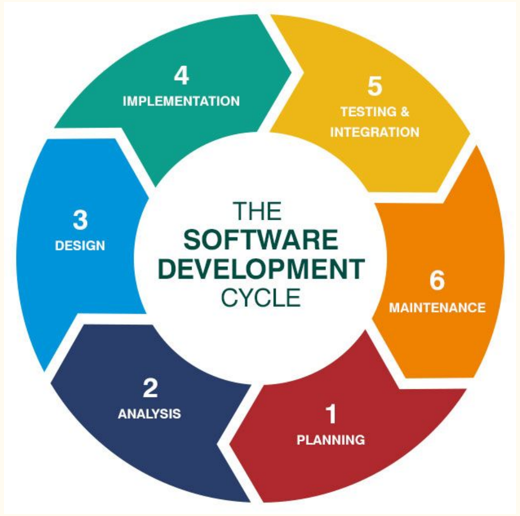
(edited)

Task completion:

- Pull Request reviews, at least one other member to approve to merge

## Reflection

- **What did we learn?**
  - **technical skills**
    - Unity
    - Node.js,
  - **soft skill**
    - communication with partner
    - team organization
  - **Takeaway message:**
    - Software engineering is more than just coding; it's a holistic approach that involves user experience considerations, the art of seamlessly connecting diverse software components and deployment strategies.



## Reflection

- **What would we do differently next time?**
  - **Unity version control or alternatives**
    - Consistent merging problem in github throughout the whole project
  - **Team work**
    - use project management tools: e.g. Jira
    - setup sprints
    - estimation of deadline

Unity DevOps
Version Control

# Individual Contribution

Sirui Chen - Team Organizer, Software developer (Unity): Inventory Manager scripts and connections to UI display, resource inventory manager, setup automated testing Github actions for Unity

Ruiting Chen - Save/load map from Unity to server, Update/Load inventory and resource item from Unity to Server, Setup deployment GitHub actions for server and Unity, Setup Render service for server deployment

Krystal Miao- Backend User registration&login , map creation&deletion. Unity login&register system(including UI and connection to server)

Tommy Gong - Backend Inventory, item creation, increment & decreament. Unity login loadmap logic( requestes betweeen Unity and server)

Yahya Elgabra - Save/load map and decorations between Unity and server, Generate new maps, Integrate decorations into maps, Implement harvest system

Ricky Wen- Partner Liason, Unity Developer, Game Designer: All models and UI sprites, building system, inventory UI, all building prefabs and UI buttons. Login transition, settings UI, population logic and utilities system, login scene background, clouds and day-night cycle system