

## **Project Milestone 7**

### **Group name:**

203-5 Trouble Twist

### **Title of the project:**

TroubleTwist Games

### **Who:**

Jake Tracy - jakewoodtracy

James Bohn - jbohn3353

Alexey Yermakov - yyexela

Conor Simmons - conorsim

James Ryan - jary7635

Blake Peery - BlakeP32

### **Project Description:**

For this project we created an online multiple-choice game. All you need to play is a single client running the Unity executable and players connect via a web browser at `[ttgames.us](http://ttgames.us)`. We had to implement a server running NodeJS to handle communications between users and clients and vice-versa. The server is hosted on an Amazon EC2 container through AWS. The website is hosted through github pages with a custom domain. The website was written in HTML, CSS, and Javascript. The Unity application was primarily written with C# scripts which implemented the game logic. The architecture that we decided to use allows us to handle variable amounts of players and games all running at the same time; furthermore, there is no need for accounts for people to use to join the game since we ensure nothing malicious is done on the back-end (much like other online multiplayer games). We tested our game through manual testing since the best way to test random user input is to have real users mess with the product to find bugs (QA testing). QA testing was performed with nearly every code addition to ensure a smooth experience.

### **Project Tracker:**

<https://csci-3308-fa20-203-5.atlassian.net/secure/RapidBoard.jspa?rapidView=1&projectKey=T25&atlOrigin=eyJpIjoiMTRIN2YyNWFiNDI5NDUzY2I4ZmNkMWY1MGFjMDVhNWUiLCJwIjoiY9>

### **Optional Video (viewable by @colorado.edu addresses):**

<https://drive.google.com/drive/u/0/folders/1oUisC91bAEfm9bmcnFwVUnDnbPjqFqGy>

## Backlog

Share ...

Q



Only My Issues

Recently Updated

VERSIONS  
EPICS



Plan your team's work

The backlog is your team's to-do list. Create 3 issues, and rank them in order of priority.

Backlog 0 issues

Create sprint ...

What needs to be done?  
New Story in Backlog

Cancel ...

## VSC:

Github Repo: [https://github.com/CSCI-3308-CU-Boulder/203\\_5\\_F20](https://github.com/CSCI-3308-CU-Boulder/203_5_F20)

- Source code is in main folder
- Test cases are described in `README.md` (manual testing)
- `README.md` explains what the project is and how to run it
- Milestone7\_203-5.pdf is in the `MILESTONES` folder

## Contributions:

[https://github.com/CSCI-3308-CU-Boulder/203\\_5\\_F20/tree/master/GitLogs](https://github.com/CSCI-3308-CU-Boulder/203_5_F20/tree/master/GitLogs)

- Please see contributions at the link above, formatted by first name and first letter of last name.

James Bohn:

- Server architecture and message handling, web client interface with server communications. Setup github pages originally and facilitated the switch to our production repo once we gained ownership. (Node.js, ws library, http, github pages, google domains)

Alexey Yermakov:

- Website UI (HTML, CSS, JS), initial server development (Creation of AWS account, EC2 instance, EBS storage, EC2 and EBS integration, other Linux server initialization), meeting management (Jira), and websocket communication development between server and clients (NodeJS).

Jake Tracy:

- Initial Unity development, project documentation, wireframe layout of Unity application, initial development of logic mapping of Unity application, quote database work, support and research for final Unity development, Demo presentation and management explanation

James Ryan:

- Initial Unity development, project/milestone documentation, initial development of logic mapping of Unity application, development of multiple choice game format for questions in Unity and file input/output utilizing a document of quotes to generate random questions and four corresponding answer choices in a random order for the gameplay

Conor Simmons:

- Initial server development, exploratory research and ideation to pivot to a different architecture (AWS), established C# websocket communication in Unity, set up the node.js server to run perpetually on AWS, maintained and updated integration between server and Unity

Blake Peery:

- Unity game host lobby, Unity-side server communication, Unity UI, created a document with quotes for our unity game, unity and C# game logic, graphics on Unity, displaying users that are connected, end of game scoreboard logic.

### **Deployment:**

Website/Client page: [ttgames.us](http://ttgames.us)

Unity application:

- [https://github.com/CSCI-3308-CU-Boulder/203\\_5\\_F20](https://github.com/CSCI-3308-CU-Boulder/203_5_F20)
- Follow instructions in the README file for building and running it