Engineering Logbook

Category

[1. Websites 2](#_Toc37445363)

[2. Literature 3](#_Toc37445364)

[a) Literature: 3](#_Toc37445365)

[b) Thesis: 3](#_Toc37445366)

[3. Notes 4](#_Toc37445367)

[(1) Summary Before 2020/04/07 4](#_Toc37445368)

[(2) 04.04: Server Programming 4](#_Toc37445369)

[(3) 04.05: Server Programming 4](#_Toc37445370)

[(4) 04.07: Meeting 4](#_Toc37445371)

[(5) 04.09: Bug Fix 4](#_Toc37445372)

[(6) 04.10: Learning 4](#_Toc37445373)

[(7) …… 5](#_Toc37445374)

1. Websites
   1. GitHub: <https://github.com/JYATMorz/BachelorThesis>
   2. GitKraken: <https://app.gitkraken.com/glo/board/XlUkZKMmogAQi2U6>
   3. Phaser3 API Document: <https://photonstorm.github.io/phaser3-docs/index.html>
   4. Phaser3 Examples: <http://phaser.io/examples>
   5. Socket.IO API Document:
      * <https://socket.io/docs/server-api/>
      * <https://socket.io/docs/client-api/>
   6. Tutorial (Chinese):
      * JavaScript: <https://www.runoob.com/js/js-tutorial.html>
      * Node.js: <https://www.runoob.com/nodejs/nodejs-tutorial.html>
      * HTML: <https://www.runoob.com/html/html-tutorial.html>
   7. More...
2. Literature
   1. Literature:
      1. <https://phaser.io/tutorials/making-your-first-phaser-3-game/part1>
      2. <https://gamedevacademy.org/create-a-basic-multiplayer-game-in-phaser-3-with-socket-io-part-1/?a=13>
      3. <http://buildnewgames.com/real-time-multiplayer/>
      4. <https://phasertutorials.com/how-to-create-a-phaser-3-mmorpg-part-1/?a=13>
      5. <https://phaser.discourse.group/t/phaser-3-real-time-multiplayer-game-with-physics/1739>
      6. <https://www.dynetisgames.com/2017/03/06/how-to-make-a-multiplayer-online-game-with-phaser-socket-io-and-node-js/>
      7. <https://gamedevacademy.org/phaser-3-tutorial/>
      8. <https://itnext.io/making-a-multiplayer-blockchain-game-using-phaser-nodejs-and-ethereum-pt-2-702b0c667e44>
      9. <https://medium.com/@tglaiel/how-to-make-your-game-run-at-60fps-24c61210fe75>
      10. <https://www.codementor.io/@amoslaber/build-a-paper-plane-with-phaser-3-and-arcade-physics-s8so8u831>
      11. More...
   2. Thesis:
      1. <https://scholar.google.com/scholar?start=30&q=distributed+systems+html5&hl=zh-CN&as_sdt=0,5>
      2. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.452.1731&rep=rep1&type=pdf>
      3. <https://dl.acm.org/doi/abs/10.1145/2305484.2305527>
      4. <https://ieeexplore.ieee.org/abstract/document/6703321>
      5. <https://ieeexplore.ieee.org/abstract/document/6598536>
      6. <https://link.springer.com/article/10.1007/s11042-017-5242-4>
      7. <https://dl.acm.org/doi/abs/10.1145/2695664.2695987>
      8. More...
3. Notes
   1. Summary Before 2020/04/07

Waiting for sorting and editing

* 1. 04.04: Server Programming
     1. Introducing Socket.IO into the server
     2. Add socket.io events emitter and listeners to handle issues of synchronizing clients => The clients can perform same action at the same time now
     3. Turn shooting logic into a broadcast event with data checking on server
     4. Current Issues:
        + Server don’t know the exact order and all action is only fine when working on the first client
        + More than 4 users can connect which will cause errors
        + Random player position for each client due to built-in game logic => Ask first client to share its players’ position
        + No editable user names
  2. 04.05: Server Programming
     1. The server now can store the user data and delete them when users connect and disconnect. => using array and setting attributes to *null* or *splice* the array
     2. Using socket.io id as the unique id for each client and check id before every broadcast event (e.g. shooting ball)
     3. The 5th and later user will not connect to the existing game
     4. Current Issues:
        + Only the team with a client can be controlled => AI || single player
        + The 5th and later user will not being notice that they won’t start the game.
        + The later client’s action won’t be broadcasted to earlier client.
  3. 04.07: Meeting
     1. Add comment (the SHA of the commit) in cards of GitKraken when it’s done & Add description (the URL of the card) when committing updates in GitHub
     2. Upload and continue fulfill logbook (Catalog needed)
     3. Try to fill the content into the template of the thesis paper
     4. Local Network Test (ASAP)
     5. Next Meeting: 04/28 (Complete the Application with certain distributed systems architecture => Peer-To-Peer & Clients-Server; Basic test design)
  4. 04.09: Bug Fix
     1. The existing clients don’t receive the event of adding new user and deleting disconnected user. => Add new socket.io event, “*newUser*” & “*playerDown*”, to handle the stored socket id in existing clients.
  5. 04.10: Learning
     1. JavaScript ES6: Promise Object
        + <https://www.runoob.com/w3cnote/es6-promise.html> (Chinese)
        + Try to force new client to load latest position only when all players and ball are still. (The function keeps waiting until the require is met.) => Event Emitter won’t work due to asynchronous callback function.
  6. ……