Project Notes

This is a log of First Practical Project work and contents do not reflect work from other sessions. Information may have changed.

Trello, Google Drive & Github used for internal team management and collaboration.

Private Github: https://github.com/wpw503/ENG1-Team-12

Attendees: Richard Liiv, Umer Fakher, William Walton, James Frost, Olly Wortley

Ice Breaking & Skills

Richard Liiv - Python

Umer Fakher - Python, Java
Computer Science With AI. Interested in machine learning/ data science

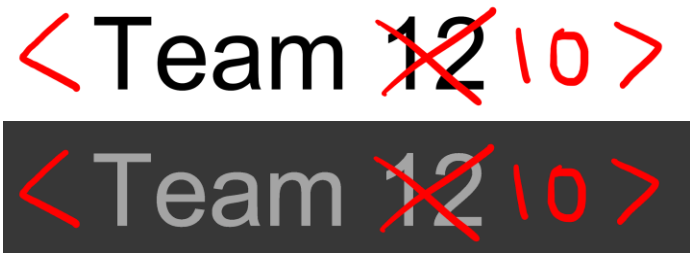
William Walton - Python, Java, C/C++

James Frost - Python, Java, GML

Olly Wortley - Python, Java

Identity

Old as this is from first practical (Changed now on Github):
Team name: “Team 12”
Logos:



Activities assigned:

- Set up communications:
    - Discord for general communication
    - Google Drive for file sharing
    - GitHub for collaborative work
- Ice Breaking Activities - All
    - Shared personal interests
    - Got to know each other
- Discussing Team identity - All
    - Agreed on team name
    - Designing Team Logo - William
- Useful links
- Schedule Interview - Umer
- Start Interview Notes
    - Create initial interview questions - will
    - Research user requirements methods in requirements engineering- All

Roles:

“Leader” and Chief Communications Officer: Umer
Resource Investigator:The people
Implementer:for the people

Useful Links

Start the shared page by outlining your team identity. If you decided on a new (internal) team name, note your agreed team identity as well. Include any image or logo you selected
Your main task is to find and justify as many websites as possible that may be of use to you (or other teams) in working out how to develop software in the ENG1 context.

- Your links must be clickable (links that we cannot open won't be counted).
- For each link, write a single sentence explaining what it is about and why it is likely to be relevant to ENG1.

Points will be awarded for the information given under team identity, and for each useful, distinct suggestion (e.g. UML notations would be one suggestion; UML class diagrams and UML sequence diagrams are not distinct suggestions!).

Yahtzee's Dev Diary - A dude makes 12 small games (one every month) and talks about some useful things.

https://github.com/wpw503/ENG1-Team-12 - Useful for collaborating with multiple people at once, keeping track of changes, and version management.

https://drive.google.com/drive/u/1/folders/17FzinMTtPQCnmvuOA3RsrAzTDCp6jkdl - A shared file sharing system by Google that can be used to work collaboratively for example on Google Docs in real-time.

https://trello.com - A board for planning out what needs to be done, what is being done, and what is done

https://en.wikipedia.org/wiki/Software_engineering - A summary of what Software Engineering may entail.

https://news.ycombinator.com/show - Various web applications and tech demos for inspiration.

https://www.gamedesigning.org/learn/java/ - Java game development tutorials, may help with coding.

Product Brief

Cohort 2 Product Brief: York Dragon Boat Race

Dragon boat racing has a rich history of ancient ceremonial and ritualistic traditions, which originated in southern central China more than 2500 years ago. Since 2003, York has its own annual Dragon Boat Challenge on the River Ouse, on a course from Scarborough Bridge to Lendal Bridge. Every year, many teams compete to be the fastest dragon on the river and win the gold medal!

Competition is structured as follows: All teams will race four times. This includes three legs and a final. The fastest of the three leg times will be used to place teams into the finals. In the "Championship Final", teams are racing for the title of the York Dragon Race (gold medals), the second place team will receive silver medals, and the third place team will receive bronze medals.

You are to build a single-player game that involves racing with one of the teams/boats in the DBR competition. Specific features that are required include:

- Every boat must have a unique spec in terms of speed, acceleration, maneuverability, and robustness.
- Over time, paddlers in the team get tired, so speed, acceleration and maneuverability decrease progressively during every leg.
- Every boat must remain in its lane for the duration of the race. Leaving the lane may result in a penalty at the discretion of the chief race official.
- Teams may find obstacles in the river during the race, like clueless ducks and geese, or tree branches floating down the river.
- Colliding against obstacles will progressively reduce the robustness of the boat, until it breaks down (resulting in the end of the game).
- Every subsequent leg will increase in difficulty level.

Constraints

You are building a game that should be playable and enjoyable by your ENG1 cohort. However, there are two stakeholders that you must accommodate.

- **The customer:** one of your lecturers (Dr. Javier Cámara - javier.camaramoreno@york.ac.uk) will play the role of a customer who is interested in eventually trying to market and sell your game. Ultimately, the customer is the person you must convince of the validity of your assumptions and decisions. This stakeholder can be contacted as often as you need and at any time (but do not expect an instant reply!).
- **The University of York Communications Office:** who is interested in using your game for its own promotional activities, e.g., at Open Days, UCAS Days. Please note that you can only communicate with this stakeholder through the lecturers.

Note: For some inspirational footage of the real event, please follow the link: https://youtu.be/-qHmlkvv9jM

Interview Preparation

Preliminary questions & thoughts to ask

Do we need to ask questions like “3d vs 2d, viewpoint of the player, real-time vs turn-based, etc? Or is that what they’re gonna give us in the ‘written requirements’.

Good question… Guess we will see when they release the lecture tomorrow.

Umer
Will
James

Things to ask

- Ask for what they would determine to be the Single Statement of Need (SSON)
- What should the user be able to do
- How should the user interact with the game

- What should the game look like (graphics)
- How should the game play (control scheme / difficulty)
- Which programming language / game engine / documentation convention should be used?
- What level of performance is expected (fps on modern computers) ?
    - Maybe specific frame times on our computers?

- What do you define as maneuverability, turning or moving side to side

- “Over time, paddlers in the team get tired, so speed, acceleration and maneuverability decrease progressively during every leg”
    - Should all teams have the same rate of tiredness and should this be dependent on how much a team has manoeuvred?

- When shall we schedule a first prototype? (which week)

Module question (out of customer role):

- Do we need to submit evidence of planning and prep for the module? A log or something of what tasks allocated to who etc or don’t need to.

- If so for customer meetings we could do it freeflow OR have roles like 3 people focus on questioning and 3 focus on jotting down interview notes into google drive

Interview Notes

Things to ask

- Ask for what they would determine to be the Single Statement of Need (SSON)

- What should the user be able to do

- How should the user interact with the game

- What should the game look like (graphics)

- How should the game play (control scheme / difficulty)

- Which programming language / game engine / documentation convention should be used?

- What level of performance is expected (fps on modern computers) ?
    - Maybe specific frame times on our computers?

    - Example for a Fit criteria:
        - Description: “Game response shall be fast”

        - Fit criterion: “90 % of the time, on average the system should have above threshold x (units of time) for y function” e.g. where y is maybe response time from when command is received to when it is executed on screen or something like that

- How much documentation should be made?

- When shall we schedule a first prototype? (which week)

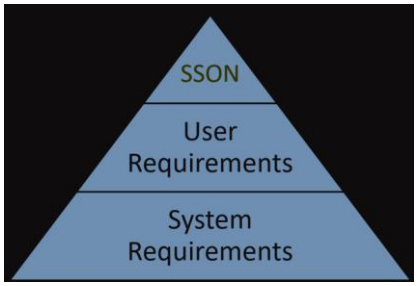
- Do we need to submit evidence of planning and prep for the module? A log or something of what tasks allocated to who etc or don't need to.

Answers

- Single player game, on dragon boat race
- Should be playable w/ a keyboard & Mouse
- Java
- 2d or 3d
- 30+ fps generally
- Easy at first builds up to harder difficulty
- Rotation or Pan movement
- No. of Boats should be so that the player can see their own boat, but enough to make it "interesting" 3-6 ish boats
- Static and Dynamic obstacles, more dynamic later on
- Onscreen numbers for tiredness, integrity ect.
- Tiredness effected by the amount of movement of the boat + degrades over length of race
- Progressive increase in difficulty through legs of race and races
- 36fps (smooth action)
- Some boats more responsive than others
- 3-7 teams per race
- Obstacles - branches, geese, rocks, bridge pillars
- Time added at the end for leaving lane
- All information on screen (e.g. position, distance, tiredness, damage etc.)

Requirements Engineering

Key Points from Lecture 5



SSON

Description:

A clear, concise statement about what is the overall goal of the system and how it will accomplish this goal

Usually determined by the client/sponsor early on

- **Although it is often poorly done (This could be a hint?)**

Example:

“The system shall enable customers to check out and pay for their shopping with minimal staff involvement.”

User Requirements

Description:

Statements regarding the tasks that users should be able to carry out using the system

- Users as stakeholders should be engaged as early as possible in the requirements process

Written for non-technical people involved in the requirements process

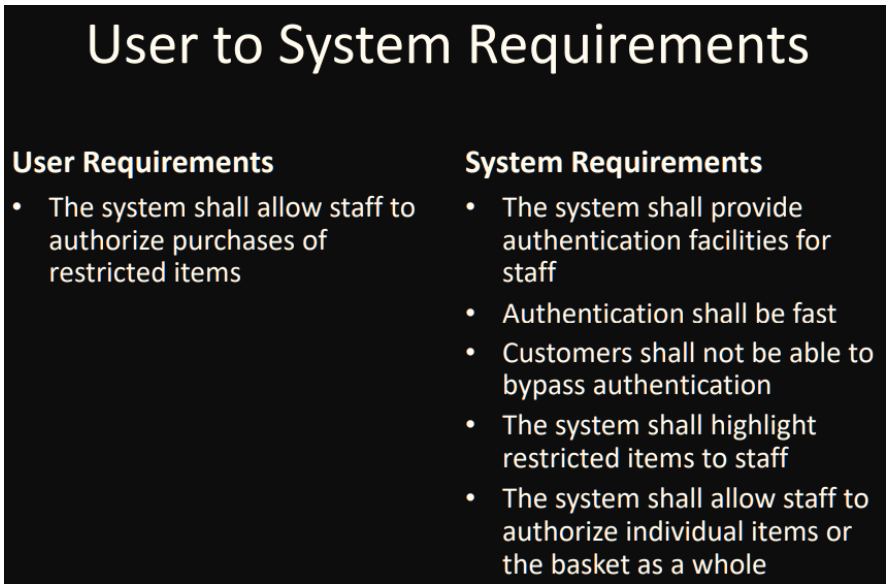
- Technical jargon should be avoided

System/Software Requirements

Description:

- A description of how the system will deliver on the needs of the users
- Detailed descriptions of functionality, services and constraints
- Written for technical implementations
- Multiple system requirements will often satisfy a single user requirement

Example for User and System Requirements:



Requirement Types

Functional requirements

- Things a system must do
- An action that the system has to take if it is to provide useful functionality for its user

Non-functional requirements (NFRs)

- Qualities a system must have
- Constraints on functional requirements
- Often critical to a system's success

Constraint requirements

- Global issues that shape the requirements
- Preferably defined at the beginning of gathering requirements

Requirements vs. Design

Most requirements can be met in more than one way:

- Requirement: “The system shall provide authentication facilities for staff”
- Solution: Fob, Username/password or Numeric passcode that changes every few hours

Try to avoid making **design/implementation decisions** during **early stages of requirements elicitation** (hint?)

- Can constraint creativity and potential for innovation during design/implementation