Team-Customer Meeting: Questions

Team 19

1. What platforms would you like the game to be compatible with? Desktop? Mobile?
    a. "Cross-platform because it's developed in Java, but the primary target platform is a standard desktop PC with a display ranging between 15-24/27 inches [check this is what he said?], with standard peripherals like keyboard and mouse."
2. How should the other boats be controlled? What kind of AI?
    a. "Different boats will have different sets of attributes. Constraints: fixed set of boats, with each having its own tradeoffs [speed, maneuverability, robustness]. Boats must be balanced.
    b. CPU boats respect the rules of the game. There is no concept of lap, only a concept of leg. The stamina etc is reset at the start of each leg, but during one the reduction of stats are continuous."
3. Should there be easy / medium / hard difficulty levels?
    a. "No, there will be no initial difficulty selected by the player. But each leg will become subsequently difficult (like heats in running races).
    b. More difficult = AI boats smarter in terms of avoiding obstacles, or adding [non-static / faster] obstacles which are more difficult to avoid in later legs."
4. Do you have a preference for the names of the competing teams, or should these names be imaginary?
    a. "You can make them up. You can take inspiration from the real event, but this shouldn't constrain your design in any way; artistic license that does not fit reality is fine."
5. Do you have a preference for what the obstacles in the river should be?
    a. "You can be creative: the minimum set of obstacles is fully-static obstacles e.g. rocks, moving obstacles (with simple trajectories) like a tree branch which floats down the river in a straight line (easy to avoid), or moving obstacles with more complex trajectories like geese which move around -- crossing the river horizontally, diagonally, etc (harder to avoid).
    b. Other obstacles can be added -- artistic license applies."
6. What is a "penalty"? Should a team be disqualified for breaking the rules? Time restrictions (for how long allowed outside of a lane)? Etc.
    a. "The idea is a fixed penalty whenever the boat gets out of its lane. For every second the boat spends out of its line a fixed penalty of X seconds will be applied (added to the total time the boat took to complete that leg).
    b. The penalty remains constant regardless of leg (and leg difficulty).

7. Should we balance the boats' properties like speed, acceleration, maneuverability? Should some boats be better in some respects but worse in others?
    a. "The player chooses which boat they'd like to use -- from a set of pre-made (by us) configurations which have a fixed acceleration, speed, maneuverability, and robustness. Again, these pre-made configurations should be balanced so as to make the game fun.
    b. The rest of the boats that were not chosen by the human player will be used by the CPU."

8. How should we model paddlers in each boat? Do they have individual speed, acceleration, how quickly do they get tired?
    a. “No, we model a boat and its crew as a singular unit. The stamina/robustness/acceleration/speed of the boat is boat + crew. All boats get tired at the same rate.”
9. Should there be a tutorial?
    a. “No, but the first leg should be easy so as to slowly ease the player into the mechanics of the game. Presenting the first static obstacle, then slowly introducing more elements of the game and increasing difficulty.
    b. Splash dialogue (click to continue) JUST before starting the first leg mentioning the controls (to control the boat), and introducing the concept of obstacles / winning.
    c. Avoid interruptions to the game: no “press space to avoid the obstacle”. Natural learning of mechanics through consequences.
10. Does the river have to look like Scarborough bridge to Lendal Bridge?
    a. “No. Feel free to experiment with the graphics of the game, you can deviate from reality.”
11. How many teams should compete in each leg?
    a. “Minimum of 3 teams -- no maximum in mind, but consider that many teams would seriously clutter the screen and reduce the playability of the game (narrow lanes, difficult to control / avoid obstacles).
    b. Only 3 teams == lanes too wide & game too easy == not enjoyable, so find a good balance. Maybe sometimes the player is forced to invade another lane, but this shouldn’t happen all the time: experimentation and testing while developing to find out what is fun.
    c. Could be 6, 8, depends on concrete design of the game. The player cannot change the number of teams.”
12. Should there be any power ups that speed up the boats or repair the boats?
    a. “Not necessary to implement these.”
13. Should it be 2D or 3D?
    a. “Preferred to be 2D, no hard constraint here but preferred.”
14. Ask about the type of music or if we should put any in the game?
    a. “Music would be nice but it’s not necessary, not a must. With respect to the kind of music if you do include it -- don’t use anything that is copyrighted, or ‘hardcore’; consider that the target audience is prospective students and their families during Open Days. Also applies to graphics: no gore, no blood. Family friendly, PG.
    b. No people getting run over by boats and exploding like in The Boys Ep1 (rip robin).” press F in the chat(Also is a spoiler)
        i. F

Note: if he ever says “this thing isn’t required but would be nice” this is a hint towards prioritising resources and time. We won’t get any extra points for things that would be nice but aren’t required: prioritise the things that are required *first* and then you can implement them later if you wish.

If we only have one or two additional questions send them by email -- this will be faster than scheduling another customer meeting. Questions can also be answered if we call them to the breakout room during a practical session.

But if there’s many questions, schedule another customer meeting.

Requirements

1. Having a working game
2. Good security that does not result in someone accidentally downloading a virus instead of our project
3. A easily understandable UI that users like
4. Good graphics, like actually good graphics that don't make me want to bleach my eyes
5. Intuitive controls that people can use (arrows VS wasd)
6. Will use UNICODE so Elon Musk's son can at least try and put his name in
7. a Cheat code for practical purposes(just to look if everything is working fine)[thinkit as a debugging feature]
8. several(eleven should suffice) image check to see if you're not a robot

Risks

1. Game not compatible with all hardware.
2. Github breaks down (Failure of working platform). Alternatively, Microsoft fucks it up.(usually what happens)
3. Disney raids(Basically any super evil corporation trying to conquer the world)
4. Everyone's laptops/PCs break down
5. The internet, regular ass Wifi issues
6. Data centre of our cloud hosting company (AWS or otherwise) violently explodes.
7. Obsolete technology (NEW AI overturning the world)
8. How to pronounce the name of Elon Musk newborn son (if he was to input it into our game, it may crash).
9. Our game is so inefficient that it results in extreme heat (Halt and Catch Fire, HCF) and an exploding device. (Overcharging)
10. The team that takes over our code falls out with us because it’s so terrible.
11. We put off doing the project for so long that we all die of sleep deprivation and hunger/thirst in the last few days before the deadline. Staff fatigue and **stress**
12. “misinterpretations” from customer cause he’s not paying us a dime
13. Stakeholders WANT MOONIE and they ain't getting any
14. The project is so poor that the customer decides to drop down from paying us £0.00 to £0.00.
15. We unionise (militantly) and demand better pay