THE FINAL FINAL DESIGN THING:

THE NAME: Beat Back The Horde

THE TEAM: The Tala Huhe, The Zach Davis, The Kayle Gishen, The Justin Ardini, The Nathan Partlan, and The Evan Wallace

THE GAME:

- There is a map. From the top come enemies, controlled by your opponent (or by AI, if you do single-player). They are trying to get to the bottom, where your base is. Your screen scrolls down with the farthest-along enemies.

- You have a side-bar with a scrolling chart of beats. At each beat, you can tap to create a defending NPC. There are also longer hold parts, where you can create a wall.

- You get combos if you tap on the beat correctly many times in a row. If you get a combo, you can create an explosion or some other cool effect. Combos get better if you keep waiting to use this, but they go away as soon as you miss.

- You can change which type of NPC you create, by clicking on the correct one when you're not needing to tap for a beat.

- There is a menu system. It could control speed, enemy strength, new/load game, etc.

- Achievements will be achievable. You can has them (if you can find them).

THE ENGINE:

- Lockstep Networking (scope: moderate): The engine will use the lockstep networking protocol, common in many games with large numbers of objects that need to be synced. This will send user input between the devices at set intervals. The networking will use Bluetooth.

- MIDI sound engine (scope: moderate): The engine will be able to load and play MIDI charts using some format.

- Level loading (scope: large): The engine will be able to load and display map files in some format, including which enemies will appear where, how fast, and where there are obstacles.

- Particles (scope: small): There will be support for creating and displaying particle emitters.

- Camera (scope: small): The map can be larger than the screen, and the camera should show only part of it, and be able to move.

- Advanced AI/Pathfinding (scope: large): There will be support for individual AI types for various enemies. There will be pathfinding, using grid-based A*, and the ability to recalculate it for a changing map. For this purpose, there will be a new data structure for the priority queue, that is better for updating values than the built-in Java priority queue. Finally, (though this may turn out to be a game feature rather than an engine feature) there will be the ability to have behaviours that follow the standard cohesion, separation, and alignment rules for flocking.

- UI widgets (scope: huge): There will be full menu and UI widget support, with multiple types of graphical widgets, fading, modal screens, and possibly layers. If a player enters a menu during gameplay, the game will continue (except for re-drawing).

- Haptic feedback (scope: small): There will be the ability to make the device vibrate, at a specific time, and for a specific duration, with specific power.

THE EXTRAS:

- Achievements (stored using shared preferences)
- Advanced difficulty level control
- Spacial partitioning, for performance improvement?
- Content generation (automagically!)
- Saving state
- Networking (multipreyer!)
- Smell-o-vision
- 4D