Final Project

Setup

Github Classroom
One group member should navigate to the GitHub Classroom invitation and create a team. All other group members should then follow the same link and join that team.

Meet with your TA
Your TA should have created a private slack channel with all of your group members. They will reach out to schedule a meeting with you. Here’s what you should prepare:

- A rough summary of your game design/idea
- A general idea of which features you want to implement
- You will work with the TA at the meeting to determine which team member will complete which feature and at which checkpoint

Timeline

Final I: November 18
- Engine features should be mostly done

Final II: November 25
- Engine should be done
- Game is playable (barebones is fine)
- 2 playtests per group member from people not in cs1950n

Final III: December 1
- Game should be more done
- 2 more playtests per group member from people not in cs1950n

Final IV: December 8
- Game should be done
- Fix bugs, polish up gameplay
● 3 more playtests per group member from people not in cs1950n
● Record a demo video!

Submission

For each final project checkpoint, create a new release the same way you’ve been doing throughout the semester. There was no stencil code included in this final project repository, but make sure to create the following three files:

● README
  ○ List of your group members’ Banner IDs
  ○ The current checkpoint’s expected deliverables (as discussed with your mentor TA), and an explanation of where we can find them
● Instructions
  ○ How do we play your game?
● Playtests
  ○ Starting from Final II, a list of playtesting reports (see Timeline above)
  ○ Each report should clearly indicate which group member interviewed them, and what the playtester’s name is (or their relation to the group member).

Demos

One person from your group should upload a demo onto Slack, just like the previous solo projects. For the Final I checkpoint, this can just be a paragraph reflecting on what engine features you added (if you don’t have a demo-able game yet).

Late Submissions

You can turn in a checkpoint late, but only if every member of the group still has a late pass remaining. However, Final IV must be submitted on-time, due to deadlines set by the university. Please reach out to the professor if you need an extension.

Grading

You and your mentor TA should have developed a list of deliverables for each checkpoint. You will be graded according to that document. Make sure you both have the same copy for reference!

The deliverables are not completely binding, and you can certainly move some items up or some times later, if you’re encountering unexpected difficulties during development.
However, we expect you to demonstrate you’ve put in a similar amount of effort/work, such as completing another feature ahead of time instead. Reach out to your mentor TA if you want to make any adjustments to these deliverables.

Feature Ideas

Each member of your team is responsible for 10 points. If a member is taking this class as a capstone, they are responsible for 20 points. Here are some ideas to get you started. Feel free to come up with your own and get them approved with your assigned TA.

Networking
5 pts - Naive networking / sending a serialized game state
15 pts - Deterministic peer-to-peer lockstep synchronization
25 pts - Client-side prediction with rewinding
10 pts - Matchmaking or lobby system

Embedded Scripting Languages
5 pts - Auto-marshalling game objects
10 pts - Extensive with well-considered API

Cutscene Scripting
5 pts - Video support
5 pts - Text box sequences
10 pts - In-game rendered animation with fixed actions

Sound
5 pts - Real-time sound manipulation (reverb and echo)
5 pts - Positional sound

Pathfinding
5 pts - String pulling when traversing waypoint graphs
5 pts - Navigation meshes

Artificial Intelligence
5 pts - Combined behavior trees and GOAP
5 pts - Inter-entity communication/squad tactics

**Graphics**
5 pts - Viewport rotation
5 pts - Smooth, polished, animated menus and transitions
10 pts - Skeletal animations
5 pts - Replace Java2D drawing with OpenGL
5 pts - Spline drawing
5 pts - Conic fog of war

**Physics**
10 pts - Physically correct rotation during collision response
5 pts - Friction
15 pts - Constraints (e.g., pin, spring, axis)

**Procedural Generation**
5 pts - Procedurally generated levels
10 pts - Value noise
10 pts - Procedural animation

**Persistence**
5 pts - Saving and loading settings/key bindings
5 pts - Saving and loading levels

**Spatial Acceleration**
5 pts Quadtree/K-D tree collision detection

**Others**
20 pts - Custom map/level editor
10 pts - Complete, polished UI system
5 pts - High scores
10 pts - High scores with anti-cheating protection
? pts - Anything you want! Just pitch it to your mentor TA.