436D Project Milestones

Project Pitch (mini milestone): Jan 12, 2018

Produce an *individual/mini-group* write-up/sale pitch which should contain a condensed game proposal.

- Story: Overall game description with possible background story or motivation.

- Core game design elements: Identify how the game satisfies the core technical requirements: rendering, geometric/sprite assets, 2D transformations, gameplay logic (passive path planning, adversarial action), basic physics, collision handling, and sound effects.

Proposal: Jan 19, 2018

Produce a detailed write-up which lays out your game story and technical elements.

- Story: Detailed game description with possible background story or motivation.

- Core game design elements: Identify how the game satisfies the core technical requirements: rendering, geometric/sprite/other assets, 2D transformations, gameplay logic, basic physics, collision handling, and sound effects.

- Advanced game design elements: List the more advanced/additional technical elements you intend to include in the game. Prioritize them based on likelihood of inclusion. Identify alternatives/impact on gameplay of skipping each of those. Explain which input devices you plan on supporting and how they map to in-game controls.

- Concepts: Produce basic sketches of the major game screens/states. These should be consistent with the game design elements, and help you assess the amount of work to be done.

- Development plan: Provide a list of weekly to bi-weekly tasks that you will work on with associated deadlines. Do account for some testing and delays and consider plan B options. Include all the major features you plan on implementing (no code). These should be consistent with the course milestones, but can pre-empt those. Provide a tentative division of labor for these tasks.

- Tools: Specify and motivate the libraries and tools that you plan on using except for C/C++ and OpenGL.

Note: **the proposal is NOT a contract but rather a roadmap, you can change the different parts as you go, BUT you need to specifically note this in your reporting, and provide an updated proposal when this happens.**

Basic Graphics: Feb 2, 2018

For this milestone you should have the core rendering functionality and basic 2D content in place including:

- Loading and rendering of non-trivial geometry
- Loading and rendering of textured geometry with correct blending
- Working 2D Transformations
- Basic shaders as needed for rendering: Similar to what was seen in the first assignment
- Response to user input (mouse, keyboard): including changes in the set of rendered objects, object geometry, position, orientation, textures, colors, and other attributes.
- Basic key-frame/state interpolation (smooth movement from point A to point B in Cartesian or angle space).

Core Game Logic & AI: Feb 16, 2018

For this milestone you should incorporate pre-planned gameplay

- Rudimental game logic: Define game states and implement principled transitions between states and interaction between the different entities.
- You should have some (possibly randomized) decision tree in place.
- Incorporate a shortest path computation into your gameplay (it can be a path in actual Cartesian space or on a state graph).
- Implement basic planning under uncertainty (e.g. A*).

Physics & Collisions: Mar 2, 2018

For this milestone you should incorporate one or more physical effects into your game including:

- A subset of the game entities should now possess non-trivial physics properties such as linear momentum, angular momentum, and acceleration and act based on those.
- You should have working collision detection and response

- Some form of simulation, which can be either background effects (e.g. water, smoke implemented using particles) or active game elements (throwing a ball, swinging a rope, etc…)
- The physical effects should be correctly integrated in time and should not be locked to the machine's speed by correctly handling the simulation timestep and integration.

Robust Play: Mar 16, 2018

For this milestone your game should be playable by others:

- The game should not terminate early – allow infinite even if repetitive gameplay
- Monkey proofing: Your game should robustly handle any user input. The game should not have any undefined behavior, memory leaks or random crashes. Unexpected inputs or environment settings should be correctly handled and reported.
- The game should not hog memory even after extended play time.
- The gameplay should be real-time (no lag). This included improving your collision handling using effective detection strategies.
- The game should allow for some form of state saving: scores, game sessions, replays...

The Grand Finale, April 14, 2018

At this stage you should incorporate s many technical and story elements as you have time for.
Note: **make sure these are consistent with the game story and do not overwhelm the user.**

- Implement the advanced features in the original proposal.
- Sound: There should be audio feedback for all meaningful interactions in the game
- User Interface: consider ways to improve playability. E.g. provide interactive help options, evaluate choice of user gestures, ease of navigation, etc.
- Gameplay: Add more advanced decision making mechanisms based on goals, state machines, behavior trees or some form of group behavior if applicable to the game.
- Physics: More complex physic interactions with the environment (e.g. gravity, bouncing, complex dynamics)
- Rendering: Implement additional shaders and visual effects (e.g. Particle Systems, 2.5D(3D) elements)
- Modeling: Incorporate additional static and dynamic assets – including rigging-based controlled deformation.