CPSC 436D - Video Game Programming
Spring 2018/19

Project Milestones

(Please note that this document may be updated as the course proceeds)

Milestone Submission:

Excluding the first game proposal pitch due on Jan 14, student teams will submit the code and proposal document for each of the milestones on the course Github repository. The repository is hosted on the UBC servers and will be accessible only to enrolled students at the address specified on the course Piazza (or via email).

For each milestone, students are expected to submit:

- *Group: [Github]* A working version of the code + a document detailing how the code aligns with the milestone requirements and the game proposal features.
- *Individual: [Piazza (private response post in appropriate thread)/email]* A small write-up describing their individual contribution
- *Individual: [Piazza (private response post in appropriate thread)/email]* A feedback report on the performance of *each* other team member

Note: Development **should not** be done on the course Github repository, student teams are expected to independently manage sharing and versioning of their game source code.

Game Proposal Pitch - **Jan 14, 2019**

Produce an *individual/mini-group* one page write-up/sale pitch which should contain a condensed game proposal. Please submit (private or public) via response post in appropriate Piazza thread.

- *Story*: Describe the overall game structure with a possible background story or motivation.

- *Core game design elements*: Identify how the game satisfies the core technical requirements: rendering; geometric/sprite/other assets; 2D geometry manipulation (transformation, collisions, etc.); gameplay logic/AI; physics.

Game Proposal - **Jan 21, 2019**

Produce a detailed design document for the game your group has decided to produce. The document should include a compelling story, describe the technical elements and a development plan aligned with the course project milestones. A minimal template is provided which includes the following sections

- *Story*: Describe the overall game structure with a possible background story or motivation.

- *Core technical game elements*: Identify how the game satisfies the core technical requirements: rendering; geometric/sprite/other assets, 2D geometry manipulation (transformation, collisions, etc.), gameplay logic/AI, physics.

- *Advanced technical game elements*: List the more advanced and additional technical elements you intend to include in the game prioritized on likelihood of inclusion. Describe the impact on the gameplay in the event of skipping each of the features and propose an alternative.

- *Devices*: Explain which input devices you plan on supporting and how they map to in-game controls.

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

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

- *Development plan*: Provide a list of tasks that your team will work on for each of the weekly deadlines. Account for some testing time and potential delays, as well as describing alternative options (plan B). Include all the major features you plan on implementing (no code). These should be consistent with the course milestones, but can pre-empt those.

Please submit (private) via response post in appropriate Piazza thread or via handin.

Note: **The proposal is NOT a contract but rather a roadmap, you can change the different parts at any point in time during the course, BUT you need to specifically note this in your reporting, and provide an updated proposal when this happens.**

Milestone: Skeletal Game - **Feb 1, 2019**

For this milestone you should have a basic game of comparable complexity to the “Hello World” assignment. This should include basic rendering, input-driven response, basic 2D motion, basic event-driven/random response, and a minimal set of assets.

(75%) *Milestone requirements*:

- Working application rendering code and shaders for background and sprite assets
- Loading and rendering of textured geometry with correct blending.
- Working basic 2D transformations.
- Keyboard/mouse control of one or more character sprites. This can include changes in the set of rendered objects, object geometry, position, orientation, textures, colors, and other attributes.
- Random or hard-coded action of (other) characters/assets.
- Basic key-frame/state interpolation (smooth movement from point A to point B in Cartesian or angle space).
- Stable game code supporting continuing execution and graceful termination.

(25%) *Creative*: You should implement one or more additional creative elements. These can include additional integrated assets or rendering effects or pre-emptive implementation of one or more features from subsequent milestones.

Your submission should align with your proposed development plan: Provide a write-up explaining how your milestone aligns with the plan. Explain all discrepancies.

Milestone: Minimal playability - **Feb 22, 2019**

For this milestones you should continue to support all required skeletal game features. You should augment those with core gameplay logic, incorporate additional assets and features that allow for non-repetitive gameplay, introduce basic user help, and perform playability testing.

(75%) *Milestone requirements:*

- Sustain progressive, non-repetitive gameplay using all required features for **2 min or more** (assume that you can provide users with oral instruction).
- You should implement state and decision tree driven (possibly randomized) response to user input and game state (create a decision tree data structure and reuse it for multiple entities)
- Provide extended sprite and background assets set as well as corresponding actions.
- Provide basic user tutorial/help.
- Stable game code supporting continuing execution and graceful termination.

(25%) *Creative:* You should implement one or more additional creative elements. These can include additional assets, rendering effects, complex gameplay logic, or pre-emptive implementation of one or more features from subsequent milestones.

Your submission should align with your proposed development plan: Provide a write-up explaining how your milestone aligns with the plan. Explain all discrepancies.

Milestone: Playability - **Mar 8, 2019**

For this milestones you should continue to support all features from prior milestones. You should also include detailed geometry, non-linear motion, collision detection and response, and time-stepping based physics. You should test the playability of all new features and ensure alignment with game development plan.

(75%) *Milestone requirements:*

- Provide complete playable prior-milestone implementation
- Sustain progressive, non-repetitive gameplay for **4min or more** including all new features (with minimal tutorial)
- Implement time stepping based physical animation: A subset of the game entities (main or background) should now possess non-trivial physics properties such as linear momentum or angular momentum, and acceleration and act based on those. Specifically, you should implement some form of physical 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…)
- Incorporate one or more polygonal geometric assets
- Implement smooth non-linear motion of one or more assets or characters
- Implement an accurate and efficient collision detection method (include multiple moving assets that necessitate collision checks)
- Stable game code supporting continuing execution and graceful termination.

(25%) *Creative:* You should implement one or more additional creative elements. These can include additional assets, rendering effects, complex gameplay logic, or pre-emptive implementation of one or more features from subsequent milestones.

Your submission should align with your proposed development plan: Provide a write-up explaining how your milestone aligns with the plan. Explain all discrepancies.

Milestone: Robust Game - **Mar 29, 2019**

For this milestone you should have a complete playable game. You should continue to support all features from prior milestones. You should support robust continuous play with no memory leaks, crashes or glitches, and be able to extensive playability testing, alignment with game development plan.

(75%) *Milestone requirements:*

- Include complete playable prior-milestone implementation
- Sustain progressive, non-repetitive gameplay for **6min or more** including all new features. The game should not terminate early and should allow infinite even if repetitive gameplay
- Support real-time response rate (i.e. lag-free input).
- Include proper memory management (no excessive allocation or leaks). The game should not have any undefined behavior, memory leaks or random crashes. The game should not hog memory even after extended play time.
- The game should robustly handle any user input. Unexpected inputs or environment settings should be correctly handled and reported.
- The gameplay should be real-time (no lag). This included improving your collision handling using effective detection strategies. You should support dozens simultaneously moving main or background assets.
- The game should allow for some form of state saving for play “reload”. Users should be able to pause and restart in a graceful (if not perfect) manner.
- The physical effects should be correctly integrated in time and should not be locked to the machine’s speed by correctly handling the simulation time step and integration.
- Stable game code supporting continuing execution and graceful termination.

(25%) *Creative:* You should implement one or more additional creative elements. These can include additional assets, rendering effects, complex gameplay logic, or pre-emptive implementation of one or more features from subsequent milestones.

Your submission should align with your proposed development plan*:* Provide a write-up explaining how your milestone aligns with the plan. Explain all discrepancies.

Milestone: The Grand Finale - **April 19, 2019**

The final version of your game should support robust and continuous gameplay as well as integrate advanced game elements created using either a game engine or alternative tools. You should implement one or more advanced gameplay features (AI, physics, geometry, or other). The game should fully comply with your game development plan.

(100%) *Milestone requirements:*

- *Development*: All features implemented in the previous milestones should be working, or improved upon if it's the case.
- *Robustness*: Sustain progressive, non-repetitive gameplay across one or more levels for **10min** including all new features. **No verbal explanation should be required at any point** during the gameplay.
- *Usability*: Include a self-explanatory tutorial introducing the player to the game mechanics.
- *External Integration*: Include integration of one or more external tools or libraries (physical simulation (PhysX, Bullet, ODE, etc …), game engines, or alternatives)
- *Advanced Graphics*: Implementation of one or more advanced graphics features including visual effects (Particle Systems, 2.5D(3D) lighting, 2D dynamic shadows) and/or advanced 2D geometric modifications (2D deformations, rigged/skinned motion).
- *Advanced Gameplay*: Implementation of one or more advanced gameplay features including advanced decision making mechanisms based on goals (path planning, A*, or similar), or some form of group behavior if applicable to the game; more complex physic interactions with the environment (e.g. gravity, bouncing, complex dynamics).
- *Accessibility*: evaluate and optimizing user-game interactions (choice of user gestures, ease of navigation, etc …).
- *Audio*: There should be audio feedback for all meaningful interactions in the game as well as a background music with tones reflecting the current state of the game.

Your submission should align with your proposed development plan: Provide a write-up explaining how your final project aligns with the plan. Explain all discrepancies.