CPSC 427 - Video Game Programming

Fall 2019/20

Milestone 2: Minimal playability – October 18, 2019

For this milestone, 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, refactor your code to support best design practices, and perform playability testing.

(75%) *Milestone requirements:*

- **Playability** (15%): Sustain progressive, non-repetitive gameplay using all required features for **2 min or more** (assume that you can provide users with oral instruction). During these 2 minutes, the player should be able to interact with the game and see **new** content for most of the time.
- **Improved Gameplay** (35%):
    - Game Logic: You should implement state and decision tree driven (possibly randomized) response to user input and game state (create a simple decision tree data structure and reuse it for multiple entities). Check out https://www.gamedev.net/articles/programming/artificial-intelligence/the-total-beginners-guide-to-game-ai-r4942/ for discussion and ideas (15%).
    - Animation: Implement sprite sheet animation or an equivalent animation system (10%).
    - Assets: Introduce new sprite and background assets as well as corresponding actions to enable interesting gameplay (10%).
    - Help: Provide basic user tutorial/help. (5%)
- **Game Design** (15%): Refactorize your code following the ECS design pattern. Define objects as game entities, and construct the actionable components for each of them.
- **Robustness** (10%): Continue to support stable gameplay with continuing execution and graceful termination. In particular, you need to ensure that
    - the game runs without severe lagging;
    - the game resolution and aspect ratio are consistent across different machines/displays;
    - the game code supports continuing execution and graceful termination.

(25%) *Creative Component*: To get the full creative credits, the game should have at least two features outside the core requirements. These can include additional assets, rendering effects (e.g. basic physics, parallax scrolling background), complex gameplay logic, a significant number of manually created sophisticated (not bought) assets, or pre-emptive implementation of features from subsequent milestones. Grading here will necessarily be subjective: more complex features or those better fitting into the overall game will be rewarded with more points.

Note: You will receive full credit for features only if they are **fully** operational. Points will be deducted for buggy and incomplete implementation.

Your submission should align with your proposed development plan: Provide a write-up explaining how your milestone aligns with the plan. Explain all discrepancies and **submit an updated proposal** when such discrepancies occur.

Submission: Submit the code and associated documents using the course Git that's been set up for you at https://github.students.cs.ubc.ca/CPSC427/team#. The repository is hosted on the UBC servers and will be accessible only to enrolled students. *Note that each team member is also expected to submit their individual progress & feedback report via 'handin'.*