# CPSC 427 - Video Game Programming

## Milestone 2: Minimal playability

For this milestone, you should continue to support **all required** skeletal game features (please fix any latent bugs). You should augment those features with core gameplay logic, incorporate additional assets and play elements that allow for non-repetitive gameplay, introduce basic user help, fix all known bugs, and perform playability testing.

**(80%)** *Mandatory Requirements:*

**Improved Gameplay** (50%):

- 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 (15%).
- Assets: Introduce new sprite and background assets as well as corresponding actions to enable interesting gameplay (10%).
- Mesh-based collision detection (e.g. mesh and straight wall) (10%)
- Help: Provide basic user tutorial/help. (5%)

- **Playability** (10%): 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.
- **Stability** (15%): Continue to support stable gameplay. 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.

**(20%)** *Creative Component*: To receive full creative credits, the game should have either *two basic features or one advanced feature* outside the mandatory requirements. Examples of tasks

you should be able to complete at this stage include adding more advanced rendering effects (e.g. basic physics and parallax scrolling backgrounds), complex gameplay logic, or a significant number of manually created sophisticated (not bought) assets. Check out the **MilestoneSubmissionForm.pdf** for ideas of other creative tasks you can complete and classifications of basic vs advanced features (please make sure you have the background knowledge to implement chosen features).

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. You will receive points for creative components only if the mandatory ones are fully operational. Points will be deducted for buggy and incomplete implementations.

**Documentation**:

- Provide a README.md providing entry points to each of the implemented features and explain them where necessary.
- **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.
- **Game Design Documentation:** Document the ECS design pattern used in your game. Enumerate the game entities and actionable components used. Draw a diagram of the interaction between entities and components. *Highlight any changes versus the previous milestone.*
- Please submit a filled **MilestoneSubmissionForm.pdf** with this and all subsequent milestones.

**Submission:** Submit the code and associated documents using the course Git repository. 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 Canvas*.