

## 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.
- 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.*

## Milestone 1 Game Design Documentation:

In this milestone, we implemented the basic gameplay features outlined in our initial development plan. We have basic collision between the player and the platform, as well as the movement and related physics implemented. The player can jump (and double jump) between the platforms, and can drop through if the “down” key is pressed. The player sprite flips based on the direction the player is facing, and the player can shoot or throw a projectile in that direction using F and P (these keybindings are subject to change). We have also met the goals of adding assets such as textured geometry for the player sprites and implemented a test map.

The following world asset functions use a standalone entity to represent them then use `emplace` and `insert` to associate with the appropriate Motion, Color components etc.

- `createBackground()`
- `createPlayer()`
- `createPlatform()`
- `createBullet()`

## Entities:

- Background
- Platform
- Player
- Bullet

## Components:

- Color
- Motion
- Collision
- Friction
- Gravity
- Platform
- Player
- Bullet

## Systems:

- Render
- Platform
- World

