**Project 03 – Mechanical Analysis**

**A Hat in Time – Character Controller**

I intend to recreate the character movement controller for Hat Kid from *A Hat in Time*, specifically mechanics available when the player has the sprint hat (allows sprinting) equipped.

Core Elements:

1. Walking: Player can move in any direction
   1. Can only be done while on ground
   2. Turns while walking, doesn’t strafe, can immediately change to opposite direction with short skip
2. Jumping: Player can jump when they are touching the ground
   1. player is able to make fine movement adjustments in air
   2. Holding jump makes the jump last longer.
3. Double Jump: Player can jump once while in the air for extra height/airtime
4. Sprinting: Player can hold a button to sprint which increases forward speed
   1. Sprint Jump: Jumping while sprinting maintains forward momentum, but loses ability to double jump; less fine control over in air adjustments
5. Dive: Can dive at any time which gives an instant forward speed
   1. Player has barely any in-air control, but can slightly adjust left/right rotation
   2. Dive preserves forward momentum from sprint
   3. Can jump out of a dive (even if already double jumped)
      1. Cancels dive momentum and returns in-air control
      2. Cannot dive again until touching ground
      3. Cannot double jump after a dive cancel, even if double jump wasn’t used before diving
   4. Landing while still in the dive causes the player to slide forward while quickly decelerating
      1. No left/right control while sliding
      2. Once fully decelerated, stay in prone dive state until player exits dive
6. Wall Jump: Jumping into a wall causes the player to run up the wall for a short while
   1. Player can only run up a short distance
   2. Reaching top of wall causes player to hop up and forward.
7. Player can rotate camera around character and can zoom in/out
   1. Clamp zoom in and out

Implementation:

1. Implement standard character controller through rigid body forces
   1. Include serialized variables for different types of movement speeds, rotations
   2. Include Booleans to determine movement state
   3. Code methods that perform the various movements and transitions between states
   4. Execute movement methods upon specific button presses
2. Setup a simple 3d environment for the character to perform each mechanic
3. Setup a simple 3d character with hitboxes to detect collision with various other objects
4. Include checks within the character controller to confirm that the character is in a valid position to perform a movement action (i.e. enabling jump when on the ground/ resetting double jump when on the ground.
5. Setup a basic camera controller which rotates around the player character in line with horizontal mouse movements, and moves closer/farther in line with vertical movements

Stretch Goals:

1. Hat Kid’s melee
2. Simple emote button (wave)
3. Camera collision with environment (doesn’t go through walls)