

*\*Items that are crossed out were removed from the final implementation*

- Made UML to streamline classes
- Scaled movement speed and attack delay based on player scale
- Rethought Shrinking Mechanic:
  - Need to inspire player to use attacks for movement
    - Idea: constantly shrink when timer reaches 0, constant rate of shrinkage (time and amount); rate of shrinkage is really fast but the timer is fairly long; timer scales to player scale, but still remains fairly long (> 3 secs)
- Added minimum scale field
- Added private “atMinScale” and “atMaxScale” functions in playercontroller; use in shrink() and grow(); change shrinktimer bool method to not check player scale
- Made buildings significantly harder to destroy
  - Player scale has to match building level to deal damage
- Added attack combos
  - ~~— Using three unique attacks in a row makes the player dash at the end~~
  - ~~— Sealing cooldown for dash?~~
  - Idea: record at most last three attack key presses
    - If next key press is already recorded, reset record but not attack delay
    - If the record is *full* (max size of 3), reset record but not attack delay
    - If next key press is not already recorded, reset attack delay and add key press to the record
      - If record fills after adding the next key press, activate dash
  - Made ComboList class
    - ComboList has a list under the hood (field), but controls what can be added to it, and what amount of things can be held
      - Add() method checks if the list already contains something or is full
    - Issue: reset means you can use the same attack twice instantaneously; need to require A DIFFERENT ATTACK for reset to apply
      - Solution: Checked before attacking if key is unique; if it's not unique, don't make the attack but still add it to the list after clearing the list
- 6/22/21
  - Issue: combo continues if you use the same key twice (key is just replaced with the same key)
    - Solution: Break combo, so you have to start it over again (Combo starts every other attack) but retain attack delay; just clear list when adding same attack and implement thing below:
  - Made first attack recorded in combo have attack delay, aka return false when adding to ComboList

- 6/22/21 - 7/18/21
  - Made it so holding down the attack button will repeat the same attack over and over again (move attack methods call to Attacks script)
    - Used Input.GetButton() so you can hold down the button (<https://youtu.be/THnivyG0Mvo?t=772>)
    - Just pressing the button will use a single attack
  - <https://youtu.be/THnivyG0Mvo?t=750> - changed attack delay variables to attack rates; more intuitive
    - Changed ScaleAttackDelay() to ScaleAttackRate()
    - Calculated nextAttack in Attack() function
- 7/18/21
  - Made scaling timer with min and max base times
    - Make shrink timer inherit from it
    - Use it for combo cooldown
    - Renamed timer class to basictimer
  - Added summary comments above classes
  - Assigned default values for public fields in scripts! Can change in editor if needed
- 7/21/21
  - Made script component fields private in all player scripts
    - Used GetComponent<>()
  - Added GetShrinkTimer() getter function to PlayerController so BuildingBehavior could reset the timer
  - Put initialization of PlayerController fields in Awake() to avoid bugs
  - Froze player's rigidbody on z axis, removed angular drag
- 7/26/21
  - Added enum and enum field to ScalingTimer to determine whether it scales up or down with player scale
- 8/3/21
  - Added priority for attacks when multiple attack keys are pressed
    - Kick > Left Punch > Right Punch
  - Made Kick only be used once per key press; does not repeat if key is held down
  - Made attack buttons/behavior less hard coded
    - added public keycode fields to attacks script with variable names associated with matching attack names; a map was unnecessary for this behavior
- 8/4/21
  - Added CanAttack() method to Attacks script for improved code readability
  - Made attack functions private
  - Improved building damage system
    - Added private float field representing player's scaled attack damage to Attacks
    - Renamed ScaleAttackRate() to Scale(), and made it scale player's attack damage as well
    - Turned building's health and scale fields into floats

- Made PlayerController's Grow() function take the building's scale in the form of a float rather than an int
- Removed Attacks field from ComboList script
- Made player movement framerate independent
  - Added private inputX and inputY float fields to PlayerController
    - Gathered these inputs from Input system in Update()
  - Set player rigidbody velocity in FixedUpdate()
- Replaced PlayerController's ScaleMovementSpeed() with Scale(), which scales the player's speed and size their current scale
- Reworded many comments
- Renamed UIButtonBehavior interface to fit naming conventions
- 8/11/21
  - Added ComboCooldownTimer() so that there is a cooldown between combos