

Progress Log (Mar)

👤 Created by	👤 Omar Owis
🕒 Created time	@March 1, 2023 9:40 PM
🏷 Tags	

Focus: Combat system

March 1st 2023

- Implemented toggleable sprinting
- Implemented dashing ability

BUGS TO FIX:

- ~~Can climb slopes greater than maxSlopeAngle with diagonal movement~~
- ~~Gravity feels like moon~~
- ~~Player sticks to walls~~
- ~~Toggling from sprinting to walking mid air does not toggle until player lands~~

March 2nd 2023

- Explored different types of shooting styles and combat systems

March 3rd 2023

Nothing was done on this day

March 4th 2023

- Wrote combat system preliminary design

- Implemented simple projectile shooting with collision detection and player rotation
- Monthly summary was added for February

March 5th 2023

- Implemented first attack cycle in fire character auto attack

March 6th 2023

- Implemented ability scriptable object
- Implemented ability controller
- Implemented cooldown handler script
- Generalized and refactored bullet shooting script

March 7th 2023

Nothing was done on this day

March 8th 2023

- Implemented combo abilities system

March 9th 2023

- Bug Fixes:
 - Player no longer sticks to walls
 - Player no longer can walk up slopes greater than maxSlopeAngle with diagonal movement
- Implemented movement and camera movement using new unity input system (jump & dash not implemented yet)
 - Controller support (movement: left joystick, cam control: right joystick, sprint: LB)

- All controls contained in one place (easier to change controls)
- Easier to add controls for other platforms (no extra code needed, just add input)
- Implemented functional humanoid player idle, walk, run animations. Animations can be used for any type of humanoid character model
- Changed character to Jammo
- Changed 3rd person camera configurations

March 10th 2023

Nothing was done on this day

March 11th 2023

- Bug Fix: Toggling from sprinting to walking mid air does not toggle until player lands
- Implemented jump & dash functionality with new unity input system
- Implemented Hierarchical State Machine for player movement (no dash state) (maybe add slope state for better performance)



BUGS TO FIX:

- ~~Sprinting state does not activate animations~~
- ~~Grounded state does not change drag of Rigidbody~~
 - ~~Player continues to move when movement inputs are not pressed~~
- ~~Moving in the air is very fast~~
- ~~Can walk on slopes greater than max slope~~

- ~~Walking/Sprinting off an edgy causes the player to fall slower than gravity const.~~

March 12th 2023

- Bug Fixes:
 - Sprinting state does not activate animations
 - Moving in the air is very fast
 - Can walk on slopes greater than max slope
 - Walking/Sprinting off an edgy causes the player to fall slower than gravity const.
 - Gravity feels like moon
- Added dash state to state machine (no momentum functionality)
- Added fall state to state machine

BUGS TO FIX:

- ~~Releasing movement keys during dash, cancels dash~~
- ~~Dashing on slopes is very fast~~
- ~~Jumping mid dash keeps player dashing until movement keys are released~~
- ~~Jumping on slopes lands the player mid air (slightly above slope)~~

March 13th 2023

Nothing was done on this day

March 14th 2023

- Bug Fixes:
 - Jumping mid dash keeps player dashing until movement keys are released
 - Walking backwards on slopes greater than maxSlopeAngle no longer pushes players up the slope (push player back relative to slope rather to player)
 - Releasing movement keys during dash, cancels dash

- Dashing on slopes is very fast
- Jumping on slopes lands the player mid air (slightly above slope)
- Player transitions to sprint state after dash (before player would standstill after dash)
- Implemented deceleration duration for dash: when exiting dash, slowly reduce the speed to sprint speed over the time period given.

March 15th 2023

Nothing was done on this day

March 16th 2023

Nothing was done on this day

March 17th 2023

Nothing was done on this day

March 18th 2023

- Finalized ability software system design
- Implemented simple shoot ability

March 19th 2023

- Implemented left hand/right hand option for simple shoot

March 20th 2023

Nothing was done on this day

March 21th 2023

- Implemented reworked combo ability system
- Switched ability controller from old input system to new input system

March 22th 2023

Nothing was done on this day

March 23th 2023

Nothing was done on this day

March 24th 2023

- Implemented global ability cooldown
- Added combat camera view

March 25th 2023

- Implemented camera switching functionality. If tab is pressed switch to other camera (temp). In the future, camera will automatically switch to combat/freestyle camera in the appropriate scenarios

March 26th 2023

- Added camera movement with combat camera
- Implemented strafing and backwards movement with combat camera

March 27th & 28th 2023

- Implemented the use of observer pattern in UI elements: Main inventory, Resource Inventory, Hot-bar, Item Tooltip, Drop Interface
 - Removes unnecessary dependences with UI
 - Game functions without UI

- Less spaghetti code
- Optimize performance
- Fixed tooltip hover bug: tooltip remains open when inventory is closed
- Fixed item drag bug: releasing the dragged item in-between slots causes item to drop
- Inventory/item code refactoring
- TODO:
 - ~~Bug: Inventory slots are initially grayed out until it is interacted with~~
 - ~~(*) Bug: When main inventory expands, position of resource inventory does not adjust~~
 - Split UI controls from Character script into UI Controller script
 - Use new input system with UI controls
 - (*) Project folder organization

March 29th 2023

- Bug fix: Inventory slots are no longer initially grayed out until it is interacted with
- Bug fix: position of resource inventory adjusts with main inventory expansion/shortening

March 30th 2023

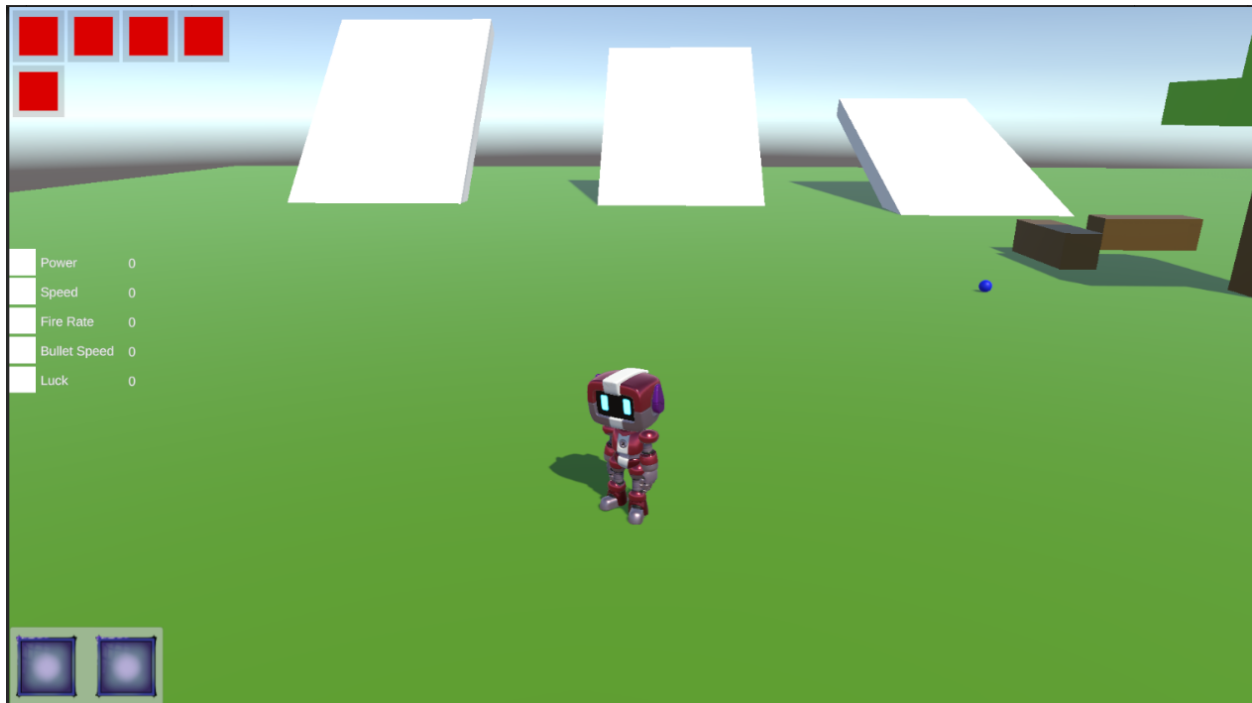
Nothing was done on this day

March 31th 2023

Nothing was done on this day

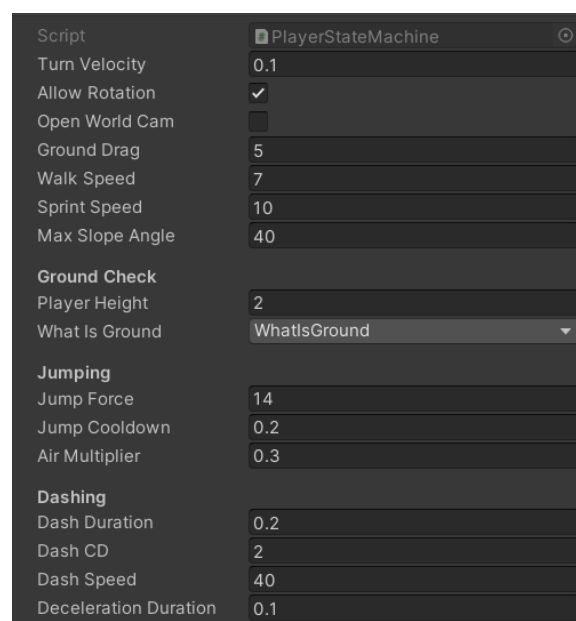
Monthly Summary:

Days without progress: 12 days out of 31

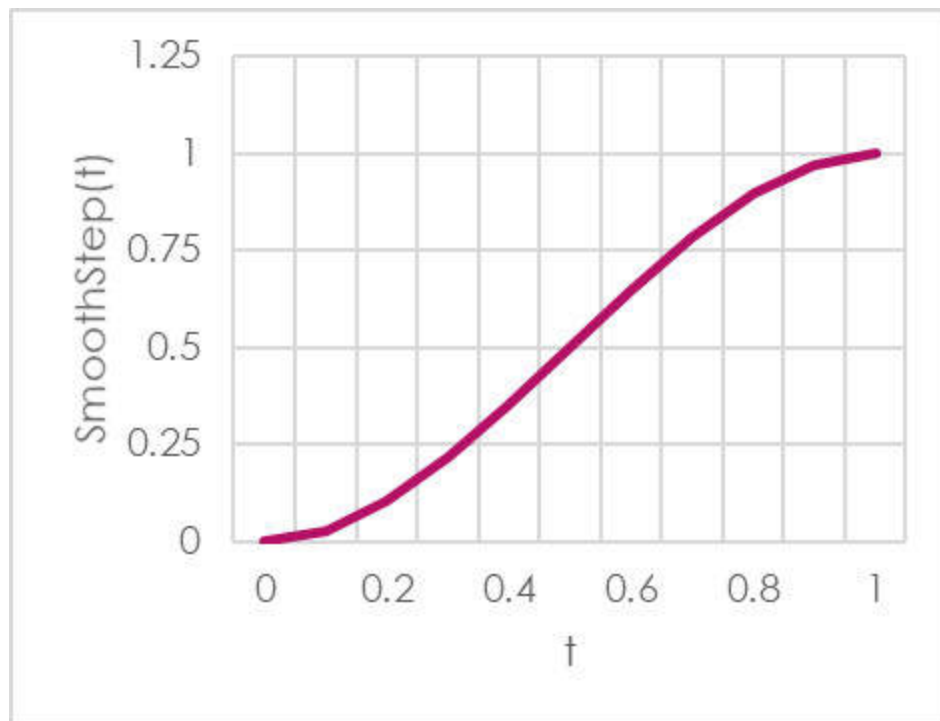


The focus of this month was suppose to be the development of the combat system. However, due to my uncontrollable ADHD and buggy implementation of current systems, the focus of turned out to be bug fixes, code refactoring/optimizations, and movement system.

Movement system:

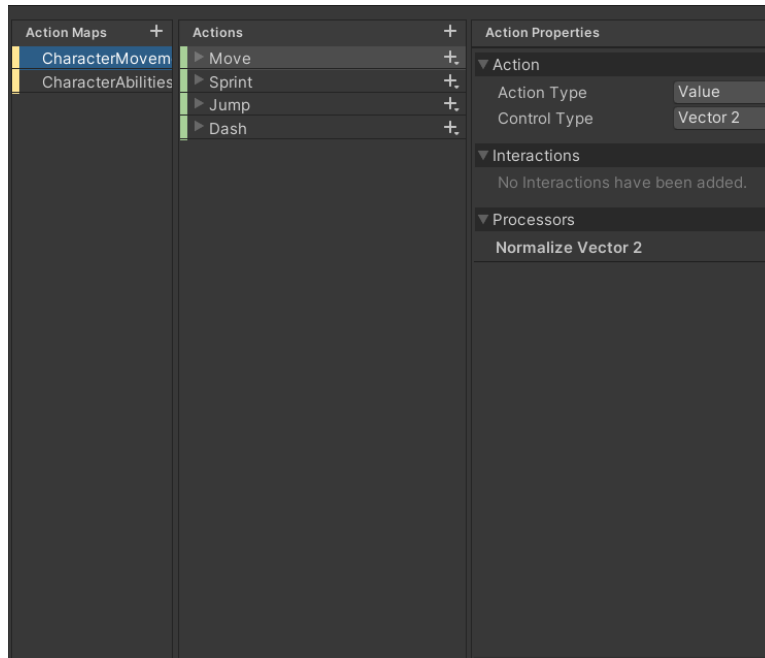


- Switched old movement script with a Hierarchical State Machine and factory pattern for player movement
 - This system allows for an easier future implementation of other movement abilities
 - Better code readability
 - Optimized performance
 - Easier debugging
- Implemented togglable sprinting with controllable speeds through editor/scripts
- Implemented dashing movement ability with ending deceleration option
 - Dash Duration: the duration in which the player is the dashing state
 - Dash CD: the cooldown time in between dashes
 - Dash Speed: the speed in which the player's RB velocity is set during the dash state
 - Deceleration Duration: the time taken for the player's RB velocity lerps to the sprint speed. below shows lerp graph with time



- Other movement variables:
 - Allow Rotation: if set to true the player rotates with movement, otherwise the player's rotation is locked
 - Open World Cam: if set to false, player looks always aiming in the direction of the camera and player is allowed to strafe. Otherwise, player moves in freelook camera mode.
 - Ground Drag: the drag force applied to the player when grounded
 - Walk Speed: the speed in which the player's RB velocity is set during the walking state
 - Sprint Speed: the speed in which the player's RB velocity is set during the sprinting state
 - Max Slope Angle: the maximum slope angle the player can move on (in degrees)
 - Player Height: the height of the player GFX model
 - What Is Ground: the layers in which are classified as ground. If the player is standing on one the selected layers, the player is considered grounded in the Hierarchical State Machine
- Implemented simple animation support (due to change)
 - Idle Animation State
 - Forward Walking State
 - Forward Sprint State

New Input System:



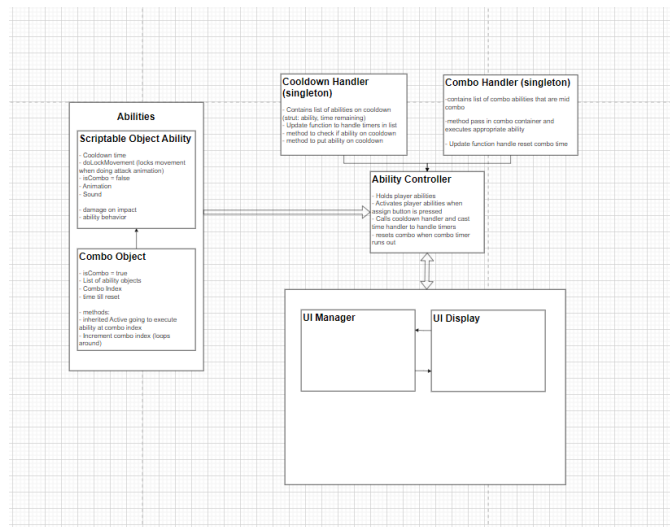
- Switched to new unity input
 - WASD/Left Joystick: controls movement
 - Left Control/LB: toggle sprint
 - Mouse Input/Right: Joystick controls camera
 - Space/South Button (X): jumps
 - Left Shift/East Button (O): dash
 - Left Click/RB: auto attack
 - Q/RT: heavy attack
 - C/West Button (square): obtainable ability 1
 - V/North Button (Triangle): obtainable ability 2

Inventory/UI Optimizations:

- Implemented the use of observer pattern in all UI elements
 - To add item to inventory, call AddItem with appropriate item and amount. No need to inventory UI, appropriate slots are automatically updated on item addition to inventory object

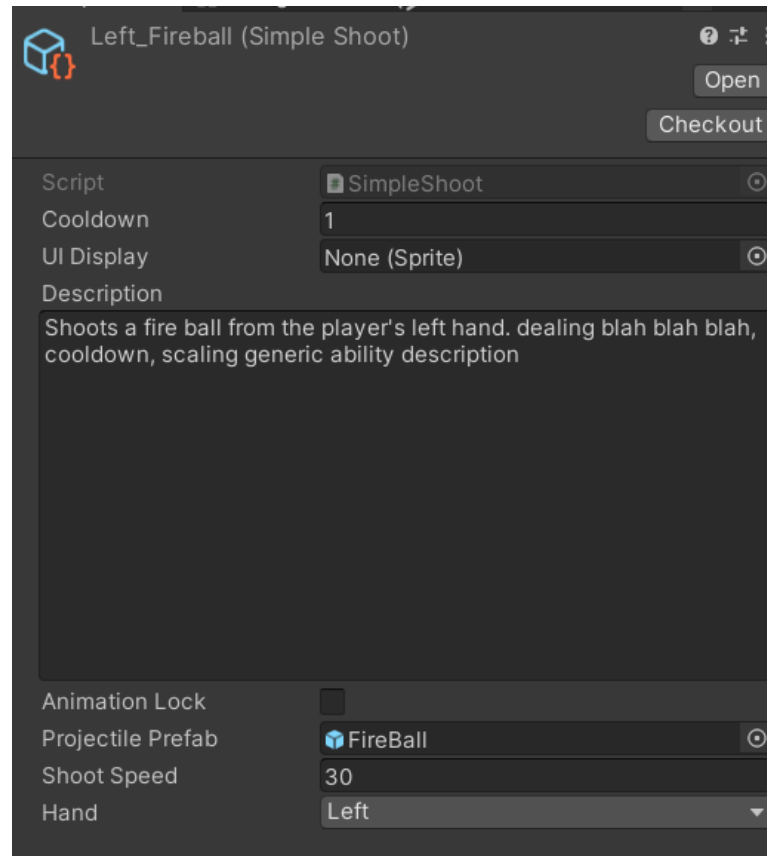
- Stat panel automatically updates when an trinket is remove/added
- Scripts no longer need to reference UI elements
 - Game will still run if UI is not added
 - Less unnecessary dependencies
 - Better code readability
 - Easier additions to inventory system
- Refactoring of inventory system code (better code readability)
- Fixed tooltip hover bug: tooltip no longer remains open when inventory is closed while hover over item
- Fixed item drag bug: releasing the dragged item in-between slots no longer causes item to drop

Combat Abilities (Incomplete)



- Implemented ability scriptable object. The ability object has the following properties, more properties can be added once inherited:
 - Cooldown (float): the cooldown time in between each ability cast
 - UI Display (sprite): the image shown on the ability bar (not implemented yet)
 - Description (string): description of the ability

- Animation Lock (bool): if true, player movement is locked until animation is finished playing (not implemented yet)
- Activate (function): when called the ability is casted
- Init (function): must be called ability pick-up/initialization



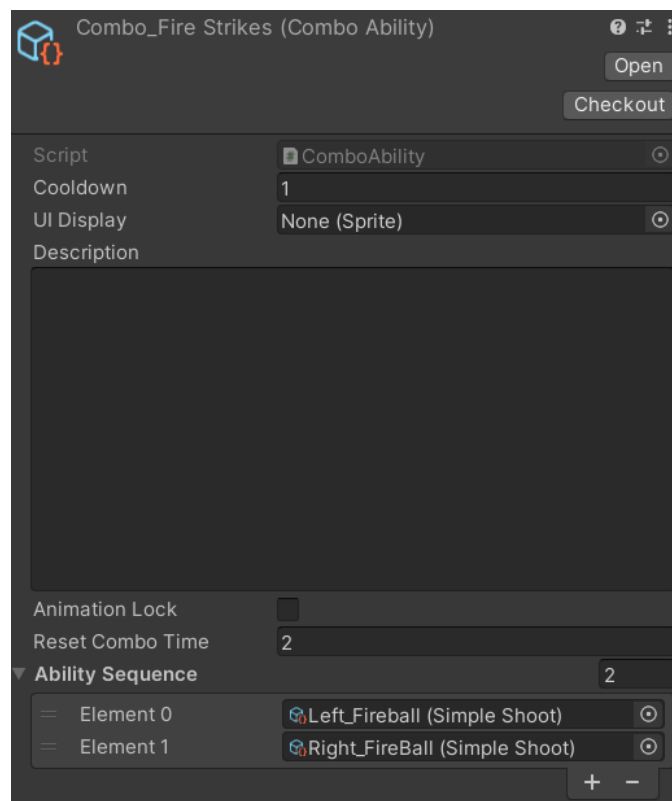
- Implemented concept simple shoot ability. Extra properties:
 - Projectile Prefab (GameObject prefab): the projectile that is shot when ability is activated
 - Shoot Speed (float): the velocity at the projectile's RB travels at
 - Hand (enum): the hand which the projectile is shot from.

Note: The projectile collision logic is handled within the project prefab itself. However, the path of the projectile is handled within the ability object itself

- Implemented combo ability scriptable object. Inherits from ability scriptable object. The combo ability object has the following properties:
 - Reset Combo Time (float): the time required for the combo ability to reset to start of combo sequence if the player stops activating combo
 - Ability Sequence (list<ability object>): the sequence of abilities activation

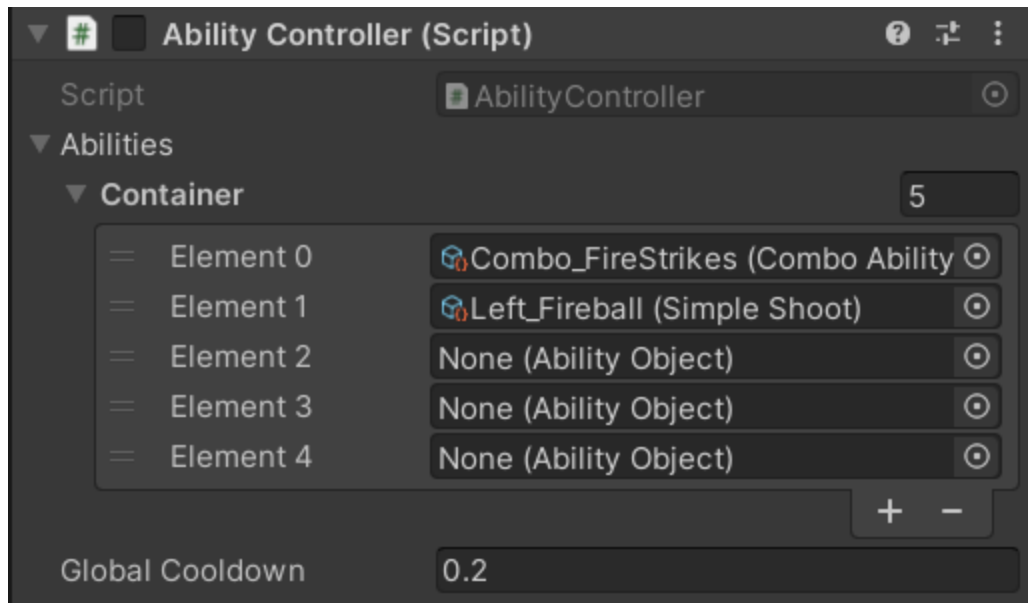
Note: The cooldown of the abilities in the sequence are ignored

- Implemented Fire Strike ability (combo)



- Implemented Cooldown handler (singleton): contains a list of all abilities on cooldown and their time remaining. Also handles the ticking of cooldown timers
- Implemented Combo Ability handler (singleton): contains a list of all combo abilities mid sequence and their reset combo timer. Returns ability to active in the combo sequence. Also handles the ticking of rest combo timer

- Implemented Ability controller. Holds the current abilities the player has. Handles ability initialization and activation. Calls cooldown handler to place ability on cooldown. Calls combo handler to get the ability to active of combo abilities



- Implemented global ability cooldown

Camera

Implemented combat camera view. Currently, pressing tab switches between open world camera and combat camera. The camera switch has a transition time. In the combat camera view, the player looks in the camera direction and the rotation of the player is handled with the mouse. Below shows the combat camera view

