Test Level Design

# Persistent Level

## Transitions

1. Ensure level transition persists properly
   1. Interactable state saved (wardrobes, door locked, and pickups)
   2. Enemies reset (for now)
   3. Trigger states should be saved
   4. Game state updates (clock and vision check) should be paused
   5. Ensure the game forcefully deselect units and reset the cursor
   6. Enemies that are not visible shouldn’t be hoverable or clickable until they come in vision sphere range (even if they are not detected via vision trace)
2. Camera moves to first member of the party (party leader) on level transition

## Main UI

1. UI should be hidden when entire party is over a level transition zone
2. UI reappears after party steps off level transition (except for settings menu)
3. Check to see that we cannot open other menus when opening a blocking menu (character stats)
4. Test to make sure that we cannot click in the backdrops of modals
5. **TBD needs fixing** **for sure** - Test save and load functionality

## Controls

1. Test out every control in the control list UI helper (bottom left)

# Quest Room

## Quest UI

1. Ensure proper goal description formatting
2. Ensure quest sidebar widget gets properly updated based on what quests users have marked to be tracked
3. Ensure quest sidebar status counts get updated
4. Ensure quest sidebar tracker works (to be reimplemented)
5. Test quest journal level indicator
6. Ensure quest journal has quest state properly recorded
7. Make sure the entries inside the quest journal are correct

## Quest Pipeline

1. Talk to an NPC to get a quest
   1. Also test out trigger quest setup
2. Test out the quest goals
   1. Test meeting collection goal
   2. Test out talking with NPC with dialog topic
   3. Test out talking with NPC with no dialog topic
   4. Test killing certain amount of enemies
   5. Test custom goals completed via trigger
   6. Test out locked goals with subgoals
   7. Test out goals with multiple “additional names”
   8. Test out goal indicator
   9. Test out before and after goal triggers
3. Test out quest completion
   1. Test out quest rewards
      1. Make sure quest has item rewards
   2. Test quest failure
   3. Ensure quest is properly updated after completion
4. Ensure automatic quest mapping works by adding a new quest tag to the quest manager
   1. Make sure the blueprints have the correct names corresponding to the gameplay tags
5. Test out quest creation pipeline (dev test)
   1. Check if QuestManager properly updates list when we attempt to add a new gameplay tag

# Combat Room

## Tasks

1. Test out every task

## AI Options

1. Test out all the different player AI settings
   1. Passive
   2. Defensive
   3. Neutral (for now it stops any custom trees)
   4. Offensive
   5. Aggressive
2. Ensure different chasing mechanics work

## Vision

1. Try to test out possible data races in the vision system

## Skills

1. Ensure system to use skills via keys and clicks work properly
2. Ensure system to add skills to action bar works by testing out spellbook UI.
   1. Have proper levels, learn a spell, and drag it on
   2. Have improper level, and attempt to learn spell but fail
   3. Attempt to drag unlearned spells
3. Ensure the actionbar reacts properly to skills being casted
   1. Should show CD’s for skills after they are activated
   2. Skill images should not change when a skills are on CD, it should just have aa red radial effect
   3. Using items should not trigger this effect (even though items cast spells unless the same spell is equipped in the action bar which wouldn’t happen in the real game since we’ll use a separate set of spells)
4. Test out actionbar drag and drop amongst its own slots
5. Use console commands to check various spell behaviors
   1. Ensure buffing correctly updates stats
   2. Ensure damage formula correctly applied
6. Ensure proper targeting for spell that falls in each category. For example:
   1. Fire bullet (single target enemy)
   2. Firestorm (aoe spell enemy)
   3. Healing Circle (aoe spell ally)
   4. TBD: Flamethrower (aoe cone)
   5. Cure (single target ally)
   6. Iron Maiden (single target both)
   7. Winter blast (line targeting)
   8. Unlock (single target interactables)

## Enemies

1. Take control of enemies and chase allied units around with console commands
2. Make sure that if enemies kill all our heroes, we reach a GameOver state (so far only a UI displays that takes us back to the main menu)

## Stats

1. Test out everything that can modify stats
   1. Spells
   2. Equipment
   3. Character Upgrades
   4. Triggers???
2. Stat Clamping
   1. Ensure vitals do not overflow
   2. Ensure vitals do not get regenerated when equipping items that give such boosts
3. Test out the effect of stats
   1. Ensure spells properly scale with the main attributes
   2. Ensure weapon damage properly scales with the main attributes
   3. Ensure spell damage reduction and piercing are applied on the right elements

## UI

1. Ensure damage numbers jump to a good height (visible) for units of different sizes
2. Ensure dying damage number shows

# Minigame

## Pawn Transition

1. Ensure we can still open up the break menu
2. Ensure no other menus or control can be accessed besides that of the minigame

# Interactable Level

## Basic Features

1. Ensure we can use keys on doors
   1. Make sure only the key with the proper id works, and if it is the wrong key the error message pops out.
2. Ensure we cannot open locked doors and pass through them
3. Ensure interactable pickups are correctly loaded
4. Ensure wardrobes are correctly loaded as well
5. Ensure that interactable that have their locations in an unreachable spot can be navigated via backup.
6. Ensure actor turns towards interactable before using it
7. Test out each of the interactable decorators
   1. Named interactable
   2. Conditional interactable
   3. Trigger interactable
8. Test out the interactable cursor
9. Ensure we cannot interact with multiple interactable during blocking interactions

## Inventory Testing

1. Test inventory drag and drop
   1. Test dragging from slot to slot
   2. Test dragging from item to inventory
   3. Test throwing away items by dropping them on the in game menu (anywhere outside of the inventory)
   4. Test dragging item to another page (hover over the arrow buttons to swap pages while holding onto an item)
   5. Test dragging item to another hero unit (hover over hero buttons to swap inventories)
2. Test inventory edge cases
   1. Test overflow inventory
   2. Tests overfill pickup (pickup shouldn’t disappear from ground if we don’t have enough room to grab everything)
   3. Test adding new items after we drop items after an overflow
      1. Add items via new pickup
      2. Add items via storage
      3. Add items via sho pNPC
3. Test the above cases with a multi hero party (use different heroes to hold inventory items)

# NPC Room

## NPC UI Test

1. When talking to an NPC, depending on the type (normal, intimate, shopkeeper), we should get different menus available
2. Ensure we can navigate the dialog wheel completely to initiate a conversation
   1. Ensure that after the conversation the wheel shows back up
   2. Ensure that we can right click to get back to the talk menu and it brings us to the correct talk menu depending on what kind of NPC we are talking to.
3. Ensure our actionbar is hidden when we start talking to an NPC, and returned after we click leave
4. Test same inventory testing features on NPC shop
   1. Ensure confirmation boxes come up on rare items
5. **TBD:** Once gifting is implemented, test out that menu as well

## Test Dialog

1. Test out dialog with various choices
2. Test out dialog that triggers things
3. Test out dialog conditionals
4. Test our custom dialog lines triggers
5. Test empty conversation starter tests
6. Test out preset dialog and preset dialog triggers
7. Test out conversation changing triggers
8. Test out default response mechanic
9. Test our conversation starter mechanic
10. Test out conversation topic on the dialog wheel
11. Test out bWantsToConverse flag
12. **TBD** – Test out dialog name text replacement (Hero1 replaced by actual hero name in first party slot and so forth. Hero replaced by interacting hero).

## NPC AI

1. Test NPC Patrolling
   1. Ensure patrol path loops
   2. NPC stops patrolling when we start to talk
   3. NPC should resume patrolling after we stop talking

## Dialog Pipeline

1. Test the dialog tool
2. Test the script to import dialog into the dialog table
3. **TBD -** split up dialog into separate tables to be loaded in separately

## Shop Mechanics

1. Ensure correct messages pop up when we do not have the requirements to buy something
2. Ensure we have the correct trade items and squeezies to trade for an item
3. Test buying item with filled backpack

# Mechanics Room

## Triggers

1. Test every single trigger with good and bad parameters

## Effects

1. Ensure fade and occlusion culling effects work and are togglable
2. Ensure the barricade kicks players from walking in a certain area

# Storyboard Room

## Events

1. Storyboard menu should fly in and out when transitioning
2. Trigger a storyboard event and ensure that the triggers associated with it proc
3. Test out the next chapter and section transitions

# Movie Room

## Cutscenes

1. Ensure cutscenes can be triggered

# Item Usage Room

## Item Check

1. **TBD:** Check functionality of items here using item spawner
2. Most tests on inventory can be done in interactable room