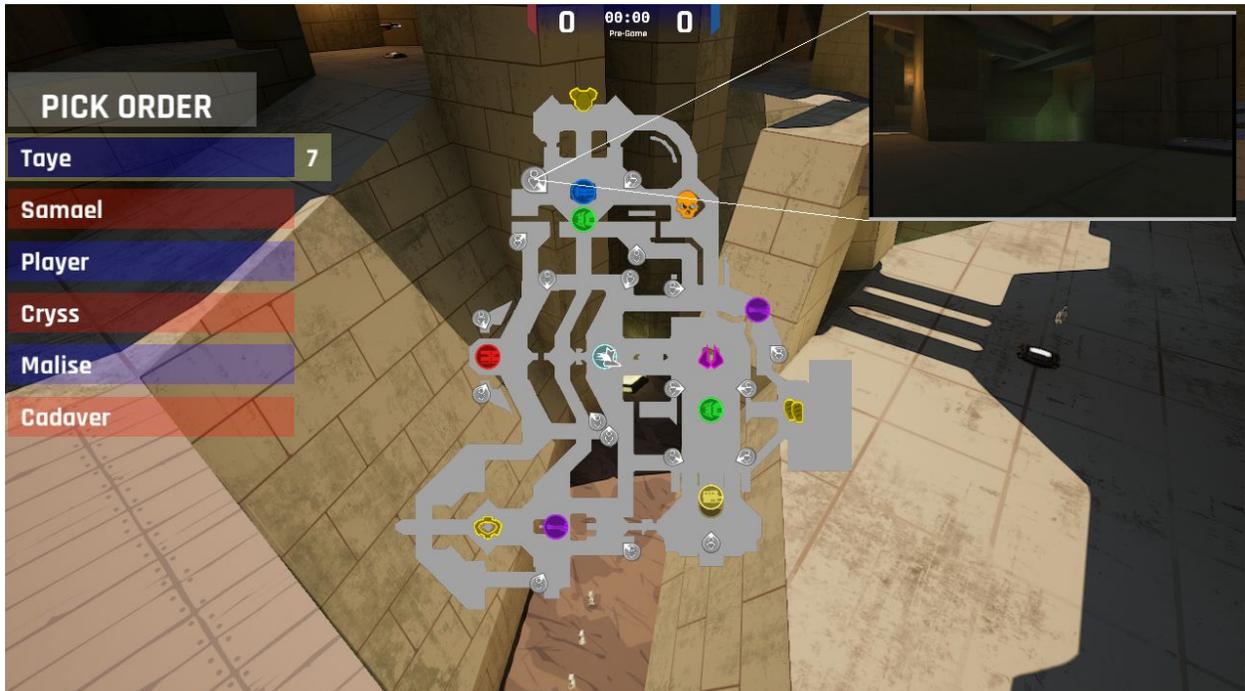


What Is Showdown?

Showdown is a round based single life elimination team game mode that focuses on coordination and resource acquisition and scarcity.

Basic Game Flow



1. Coin toss decides which team picks first
2. Players from both teams now alternate in picking their spawn points from the mini map within the given time
3. Once everyone has picked their spawns the match begins
4. Players collect weapons and powerups and try to eliminate the other team within the given round time
5. The first team to eliminate all enemy players wins the round
 - a. If the end of round time hits with players still being alive on both teams the team with most players alive wins the round
 - b. If there are equal amounts of players alive on both teams the team with the larger stack (combined HP & Armor) wins the round
6. Players return to the spawn selection screen
7. The pick order cycles once for both teams and the losing team gets first pick for the next round
8. Repeat until one team fulfills the number of rounds necessary to win the match

Additional Information

- Our target format is 3 vs 3
- All resources during a round are finite
 - Weapons, Health, Ammo, Powerups, and Armor are available to be picked up immediately but do not respawn during the round
- Players start each round with 200 HP
 - Regular HP Pickups will heal players back up to 200
 - Vials and Keg Of Health will overheal Players up to 299 HP
- 60s before the end of a round the Overcharge powerup spawns into the map
 - The starting location of the powerup is visible both on the minimap during the pick phase as well as in the world through geometry
 - Once picked up the owning player has 30s to live but is now overcharged
 - HP and ammo are replenished
 - Enemy locations are periodically made visible via pings through the wall
 - Damage output and movement speed are amplified
- When a player is killed they drop their entire inventory along with a generic ammo box

Playing Showdown

To get the best idea of what Showdown is all about (and to form your own opinion about what might work well or doesn't) the best thing to do is to actually go and play a couple rounds online or vs bots.

Stock Map Examples

These are the current stock maps that we and/or our community have found to play well in TSD

- DM-Chill
- DM-Deck
- DM-Focus
- DM-Sand
- DM-Temple

Showdown Level Design

The basic premise of the Game Mode is to be able to utilize our existing levels to maximize usage due to our limited resources while at the same time breathing new life into them with unique fresh gameplay.

As a starting point, the type of levels that have crystallized to be more likely to be fun to play and best suited for the mode are the ones that were initially built with classic UT Team Deathmatch & Free For All in mind.

That being said there is a lot of room and desire for experimentation. Part of the great potential of this contest is to allow people to come up with unique scenarios and setups that could help shape the future of this mode.

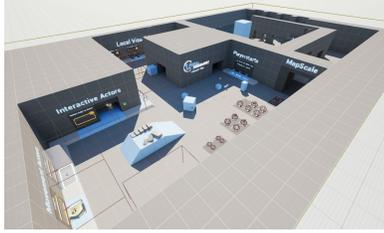
Design Ingredients

Some of the things that we have found to contribute to making a good TSD map so far.

- Medium size
 - We want the map to be able to support 6 players while still allowing them room to spread out just a little and disengage without getting lost too easily
- Iconic powerup locations
 - Visually easy to remember and call out
 - Interesting to fight over with regards to risk vs reward
 - I.e in a highly visible or disadvantageous position
 - These have worked well so far when thought of like controllable areas in a TDM match, i.e. limited ways in with obvious choke points that could be held by 2 or more people
- At least two ways to get to every key location in the map
 - Unless specifically designed for other reasons it is always desirable to do this to encourage and allow players to engage in cut off game play and try and anticipate their opponents movement in a rewarding fashion
 - These don't have to be in close proximity
- Interesting and fair Playerstart placement
 - These have a lot of power over how the map plays out and will determine how interesting the plays can be that people will come up with
 - Totally fine to follow standard good practice
 - I.e. always face spawns in a useful direction
 - Never spawn players too far away from a weapon, ideally facing one or in a position that naturally funnels a player towards one quickly
 - Don't spawn players next to power ups and/or dominant weapons specific to the map

Technical Ingredients

These are some of the things necessary to know about for putting together a TSD capable level.



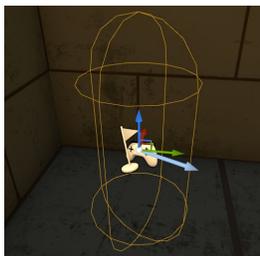
Browsing around **Example_Map** (which opens on editor load) is a good resource for a basic level setup and some other guidelines and examples of items we use to build our levels. I.e. Weapon and Powerup Bases, lighting & Post Process actors etc



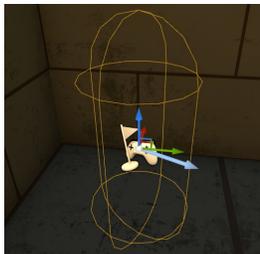
The MiniMap in TSD utilizes the Nav Mesh polygons as visual data, therefore it's crucial you have a **NavMeshBoundsVolume(s)** around the playable areas of your level for the game mode to function at all.



The Overcharge Powerup utilizes **UTGenericObjectivePoint** actors as a spawn location (located in the Class Viewer). We typically place 3 of these in iconic locations around the map and the mode will randomly pick between them at the beginning of a round. Placing more or less than that is fair game as desired.



UTPlayerStart actors are used as the spawn points in UT. **IgnoreInShowdown** can be set to true in its details to provide a richer density of spawns for other game modes without overcrowding the selection in TSD.



UTTeamPlayerStart actors are can be used for experimental maps where you want to define a fixed selection for each team to pick from that is limited by the designer. I.e. a front line scenario like one of our test shells named DM-Kapow. TeamNum 0=Red 1=Blue.

Level Ship Checklist

This is the general checklist we follow when shipping a map.

General

LightMassImportance Volume(s)	Needs to encompass all playable areas of the map snugly since it generates samples for indirect lighting that are stored and therefore use memory. Can use more than one but volume shapes will always revert to rectangles under the hood)
World Settings	Level Summary - Everything needs to be filled out; Title, Player Counts, Author, Description, Screenshot. GameMode - revert any overrides back to default for non tutorial content if applicable
Map Check	Everything looks clean. Errors that can't be resolved in content need to be delegated and/or investigated with code.

Gameplay

Blocking Volume Pass	Map sealed, InvisibleWall fine for most geometry. Use BlockAll when applicable for gameplay reasons (i.e. for predictable flak shard ricochet of a surface that is almost smooth). Spectator cameras need to be kept inside of the map as well. Players can't get on top of undesirable locations.
Spectator Cameras	Hotkeys support 10 max (0-9) placed in iconic and/or strategic locations across the map
KillZ Volume(s)	If applicable to the map frame far out borders of the level with these to prevent players from hitting walls in nowhere land and/or getting views of unmeshed or unperformant areas.

Mesh Collision Setting Pass	IgnoreOnlyPawn for solid geometry, NoCollision for fog sheets etc; Disable weapon related collision selectively where applicable (i.e. don't want projectiles hitting empty spaces between railing handles etc)
Generic Objective Markers	I.e. Showdown uses these to randomly place the Overcharge Powerup. A total of 3 spread out in interesting/iconic locations similar to other major powerup rules work well (i.e. think of it as if placing 3 Domination points).
Navmesh Bounds Volume(s)	Needed for bot pathing and also used for minimap generation. Should snugly encompass navigable playspace. We do not want nav polies to generate outside of playable space; in that case adjust volume for optimal snug.
UTNavBlockingVolume(s)	Used for allowing navmesh to generate in tight spots that otherwise would be too narrow. I.e. a thin beam that is intended to be walkable.

Visual

Holes	Sanity check map is visually sealed properly; i.e. nooks and crannies tend to develop little holes after visual passes where higher frequency and/or organic meshing occurs. Any areas where the player can fall to his/her death need to have geo supporting the player falling without seeing OOE (this needs enough space available so that you could reasonably translocate back up before dying or use jumpboots). Redeemers flying around, etc.
Post Process	Except for gameplay specific effects all settings should be left default unless mentioned below.

	WhiteBalance: Temp +/-1000; Tint +/- 0.1
	SceneColor: VignetteIntensity 0 - 0.4
	Bloom: DirtMask however applicable
	ScreenSpaceReflections: MaxRoughness 0.6 - 0.8
	AmbientOcclusion: Intensity 0 - 1.0, Radius can be tweaked for look but 99% of the time 200 works, RadiusInWorldSpace true/false based on desired look
	AutoExposure: Min/Max Brightness 1.0, Exposure Bias 0 - 2.0
	GlobalIllumination: IndirectLightingIntensity 1.0 - 1.5

Audio

Music	Hooked up in World Settings > Level Settings. Ensure that music track is set to loop. Ensure an even distribution of audio tracks across all levels of the game.
Ambient Audio Pass	Ensure that there are no areas in the world where things fall completely silent, and limit overlap of ambient radii to avoid sound dropping. At a minimum, ensure that all obvious sources (torches, moving machinery), and gameplay relevant items (traps, etc.) have audible audio. Make sure that all moving objects (doors, lifts, etc.) have assigned audio. All ambient audio files should be in the Effects sound group and the Ambient sound class.

Reverb Volumes	Sanity check all reverb to ensure that it is not distracting to gameplay. Set transition time to low value (1s or less typically) and make sure that priority values are set if there are nested volumes.
-----------------------	---

Performance

Decals	Make sure Fade Distance is set properly (decal fades out at a reasonable distance, spotfix where necessary for pops). Make sure bounds are snug with the environment (i.e. maintain a flat profile for puddles/stains on floors/walls etc to reduce overdraw. Watch the bounds to avoid decal stretching along edges or nearby objects.
Mesh Collision	Distant objects (including the sky dome) that are outside of reasonably perceptive range for impact FX etc for weapons and aren't necessary for gamepaly clarity should have their collision set to NoCollision.
Detail Settings	Set High/Medium/low settings on geometry and particle systems. I.e. StaticMeshes need to be set to not cast a shadow (typically pebbles, clutter, etc). Anything that is gameplay relevant (i.e. particle system purposefully occluding an area) should always be set to Low.
Shader Complexity	Viewmode > ShaderComplexity. Bright green whenever possible. Reduce overdraw as much as possible.
Cull Distance Volume(s)	Usually one for the entire map will do the trick. Smaller granularity when applicable. Outpost23 has good reference settings though they will always need to be spotchecked in every map from all major angles.

Light Maps	Utilize Lightmap Density Viewmode to achieve roughly even density across the map. Lowball everything first and dial back up where needed. Special focus on directly lit/shadowed areas. Indirectly lit areas can often get away with very low res.
Detail Mode	Actors in the world can be set to low/medium/high detail. Whatever they are set to acts as the lowest common denominator, i.e. leaving something at the default setting of "low" means it will always show up. Thing that are hidden in this way can not cast shadows or be relevant to gameplay.
StaticMesh Count	Merge clusters of meshes sharing the same material(s) whenever feasible to reduce draw calls.