

# Quantum Mechanic: Next-Generation Features Roadmap

## Executive Summary

This roadmap identifies architectural bottlenecks and designs "gravity well" systems that pull players into deep engagement loops. Each feature is designed for **modular integration** with the existing Quantum Mechanic framework.

---



## Phase 1: Procedural Model Generation System (Foundation)

### Bottleneck Identified

The current system uses primitive capsules. We need a **parametric mesh generator** that creates visually distinct models on-demand without external asset dependencies.

### ProceduralModelFactory.cs

#### Architecture:

- Argument-driven generation: `GenerateModel(ModelRequest request)`
- Mesh types: Humanoid, Weapon, Armor, Creature, Environmental
- Style parameters: Cyberpunk, Fantasy, Organic, Mechanical
- UV unwrapping for runtime texture application
- LOD generation (3 levels: High/Medium/Low)

#### Model Request Structure:

```
csharp
```

```

public class ModelRequest
{
    public ModelType Type; // Humanoid, Weapon, Armor, etc.
    public ModelStyle Style; // Cyberpunk, Fantasy, Organic
    public float ScaleFactor;
    public Dictionary<string, float> Parameters; // Height, bulk, sharpness, etc.
    public string[] Augmentations; // For cybernetic overlays
}

```

## Technical Approach:

- **Humanoid Generation:** Parametric skeleton with blend shapes
  - Body type sliders (endomorph/ectomorph/mesomorph)
  - Height/proportions
  - Muscle definition
  - Gender presentation (continuous slider, not binary)
- **Weapon Generation:** Modular component assembly
  - Blade types: Straight, curved, serrated, energy-field
  - Handle geometry: Grip patterns, guard styles
  - Material hints: Metal, wood, crystal, plasma
  - Physics properties: Weight, balance point, swing arc
- **Augmentation Overlays:** Additive mesh layers
  - Cybernetic limbs with glowing circuitry
  - Biometric implants
  - Psionic resonators (floating geometric shapes)

**Gravity Well:** Players unlock "design tokens" through gameplay to customize model generation parameters.

---

## 6 Phase 2: Deus Ex-Style Augmentation System

### Bottleneck Identified

Character progression needs depth beyond level/XP. Augmentations create **permanent build diversity** and strategic choice tension.

### AugmentationManager.cs

#### Core Mechanics:

#### Augmentation Slots:

- Neural (head): Vision modes, hack speed, memory capacity
- Torso: Energy shields, oxygen reserves, toxin filters
- Arms: Strength multipliers, weapon stabilization, tool integration
- Legs: Jump height, sprint duration, silent movement
- Dermal: Armor rating, camo skin, sensory feedback

#### Energy System:

- Each augmentation consumes "Bioelectric Energy" (rechargeable resource)
- Trade-offs: More augments = faster energy drain
- Energy cells found in dungeons or crafted

#### Augmentation Tree Examples:

## Icarus Landing System (Legs)

- Tier 1: Reduced fall damage
- Tier 2: No fall damage + shockwave on landing
- Tier 3: Kinetic energy conversion (falling charges energy)

## Typhoon Explosive System (Torso)

- Tier 1: 360° blade discharge (close range AoE)
- Tier 2: Increased radius + stun effect
- Tier 3: Secondary shard projectiles

## Smart Vision (Neural)

- Tier 1: Highlight interactive objects
- Tier 2: See enemies through walls
- Tier 3: Trajectory prediction for ranged attacks

## Network Integration:

- Augmentations affect packet data (TransformData includes active augments)
- Server validates energy consumption to prevent cheating
- Visual effects sync across clients (glowing limbs, energy discharges)

**Gravity Well:** Augmentation choices are **permanent** unless you find rare "Recalibration Kits" in high-level dungeons. This creates meaningful build diversity and trade communities.

---

## Phase 3: D&D Character Creation System

### Bottleneck Identified

Players need **narrative investment** at session start. D&D-style creation provides this through choice paralysis (positive).

### CharacterCreationSystem.cs

#### Race Selection (8 Races):

- Human: +2 to any stat, bonus skill point
- Elf: +2 Dexterity, darkvision, magic affinity
- Dwarf: +2 Constitution, poison resistance, mining bonus
- Orc: +2 Strength, intimidation bonus, berserker rage
- Tiefling: +2 Charisma, fire resistance, demonic pact abilities
- Dragonborn: +2 Strength, breath weapon, scale armor
- Gnome: +2 Intelligence, small size (stealth bonus), invention skill
- Halfling: +2 Luck, reroll failed saves, nimble

#### Class System (6 Base Classes):

- **Warrior:** Melee specialist, heavy armor, rage abilities
- **Rogue:** Stealth, backstab, lockpicking, trap detection
- **Mage:** Elemental magic, ritual casting, mana system
- **Cleric:** Healing, buffs, divine intervention, turn undead
- **Ranger:** Ranged combat, animal companion, survival
- **Psion:** Psionic powers, telekinesis, mind blast, astral projection

## Stat Point Allocation (27-Point Buy):

- Strength, Dexterity, Constitution, Intelligence, Wisdom, Charisma
- Stats range 8-15 (before racial mods)
- Point costs scale exponentially (14→15 costs 2 points)

## Background System:

- Soldier: +2 Athletics, starting weapon proficiency
- Scholar: +2 Arcana, extra cantrip
- Criminal: +2 Stealth, lockpick toolkit
- Noble: +2 Persuasion, starting gold bonus
- Each background grants unique dialogue options (stored in PlayerData)

## Procedural Model Integration:

csharp

```
ModelRequest request = new ModelRequest
{
    Type = ModelType.Humanoid,
    Style = GetRaceStyle(selectedRace),
    Parameters = new Dictionary<string, float>
    {
        {"height", GetRacialHeight(selectedRace)},
        {"bulk", GetStatValue(Constitution) / 10f},
        {"musculature", GetStatValue(Strength) / 10f}
    }
};

GameObject characterModel = ProceduralModelFactory.GenerateModel(request);
```

**Gravity Well:** Multiclassing unlocks at level 5. Each class choice opens new augmentation compatibility and weapon proficiencies.

---

## Phase 4: WoW-Style Auction House

### Bottleneck Identified

Economy needs **player-driven price discovery** and asynchronous trading. Current EconomyManager only supports NPC vendors.

### AuctionHouseSystem.cs

#### Core Features:

##### Listing Creation:

- Item + starting bid + buyout price + duration (1h/12h/24h/48h)
- Auction fee: 5% of starting bid (lost even if no sale)
- Deposit refunded on successful sale

##### Bidding System:

- Real-time bid updates via network packets (PacketType.Auction)
- Proxy bidding: Set max bid, system auto-bids for you
- Bid increment: 5% of current price
- Outbid notifications

##### Search & Filter:

- Category tree: Weapons > Melee > Swords > Energy Swords

- Level range filter
- Rarity filter (common/uncommon/rare/epic/legendary)
- Price sort (ascending/descending)
- Recently posted / Ending soon tabs

## **Mail System (Required for Auction House):**

- Inbox stores won items + refunded deposits
- 30-day expiration with auto-return to sender
- Attachments: Gold + items
- System mail for auction notifications

## **Server Architecture:**

csharp

```
public class AuctionListing
{
    public uint ListingId;
    public uint SellerId;
    public string SellerName;
    public string ItemId;
    public int Quantity;
    public int StartingBid;
    public int BuyoutPrice;
    public int CurrentBid;
    public uint CurrentBidderId;
    public long ExpirationTimestamp;
    public AuctionStatus Status; // Active, Sold, Expired
}
```

## **Network Packets:**

- **AuctionListRequest**: Client requests active listings
- **AuctionBidRequest**: Client places bid
- **AuctionBuyoutRequest**: Client buyouts immediately
- **AuctionListingUpdate**: Server broadcasts new bids

**Gravity Well:** "Market Barons" achievement for 1000+ successful sales. Unlocks premium AH stall with custom banner (procedurally generated).

---

## **✨ Phase 5: Magic System Architecture**

### **Bottleneck Identified**

Combat needs **strategic depth** beyond melee/ranged. Magic introduces resource management, combo systems, and environmental interaction.

### **MagicSystem.cs**

#### **Mana System:**

- Base mana pool ( $100 + \text{Intelligence} * 10$ )
- Regeneration: 5/second out of combat, 1/second in combat
- Mana potions (instant restore) vs. mana crystals (slow regen boost)

#### **Schools of Magic (6 Schools):**

#### **Evocation (Damage):**

- Fireball: AoE damage, ignites terrain

- Lightning Bolt: Chain damage (jumps to 3 targets)
- Ice Shard: Single target + slow effect
- Arcane Missiles: Homing projectiles

### **Conjuration (Summoning):**

- Summon Familiar: AI-controlled combat pet
- Create Food/Water: Generate consumables
- Gate: Teleportation portal (requires 2 casters for long distance)

### **Abjuration (Defense):**

- Mana Shield: Converts mana to damage absorption
- Dispel Magic: Remove buffs/debuffs
- Counterspell: Interrupt enemy casts (timing challenge)

### **Transmutation (Utility):**

- Polymorph: Turn enemy into sheep (1 minute, breaks on damage)
- Slow/Haste: Time manipulation
- Stone to Flesh: Dungeon puzzle solving

### **Necromancy (Dark Magic):**

- Raise Dead: Temporary skeleton minions
- Drain Life: Damage + self-heal
- Fear: Cause enemies to flee

### **Illusion (Control):**

- Invisibility: Stealth mode (breaks on action)
- Mirror Image: Decoy clones
- Charm: Temporary mind control

## **Spellcasting Mechanics:**

- Casting time (channeled spells interruptible)
- Cooldowns (prevent spam)
- Reagent costs for powerful spells
- Spell combo system: Cast Ice Shard → Lightning Bolt = Frozen Shatter (bonus damage)

## **Spell Progression:**

- Spell ranks: Fireball I → Fireball X (increased damage/reduced cost)
- Spell unlocks via leveling OR found as scroll loot in dungeons
- Spellbook UI stores 100+ spells, hotbar has 12 slots

## **Network Synchronization:**

csharp

```
public class SpellCastData
{
    public string spellId;
    public uint casterId;
    public uint targetId; // 0 for ground-targeted
    public float targetPosX, targetPosY, targetPosZ;
    public float castStartTime;
    public float castDuration;
}
```

**Gravity Well:** "Archmage" prestige class unlocks at level 50 with 100+ spells learned. Grants ability to create custom spell combinations.

---

## 🎯 Phase 6: Ranged Combat Physics System

### Bottleneck Identified

Current system has no projectile physics. Ranged combat needs **skill expression** through leading targets, bullet drop, and ricochet mechanics.

### ProjectilePhysicsSystem.cs

#### Weapon Categories:

#### Bows/Crossbows:

- Parabolic arc trajectory (gravity-affected)
- Draw time affects power/distance
- Wind simulation (optional difficulty modifier)
- Arrow types: Broadhead (damage), Bodkin (armor pierce), Fire (DoT)

## **Firearms:**

- Hitscan vs. projectile toggle (balance vs. realism)
- Bullet drop at 50+ meters
- Ricochet angles on metal surfaces
- Weapon spread (increases with rapid fire)
- Reload mechanics (clip size, reload speed stats)

## **Energy Weapons:**

- Laser: Instant hitscan, heat buildup (overheat = forced cooldown)
- Plasma: Slow-moving projectiles, AoE splash damage
- Rail Gun: Penetrates multiple enemies, long charge time

## **Thrown Weapons:**

- Kunai: Fast, low damage, affected by gravity
- Grenades: Timed/impact fuses, bounce physics
- Boomerangs: Return trajectory (catch for ammo refund)

## **Physics Calculations:**

csharp

```

public class Projectile : MonoBehaviour
{
    public Vector3 velocity;
    public float mass;
    public float drag;
    public bool affectedByGravity;
    public int penetrationPower;
    public DamageType damageType;

    void FixedUpdate()
    {
        // Apply gravity
        if (affectedByGravity)
            velocity += Physics.gravity * Time.fixedDeltaTime;

        // Apply drag
        velocity *= (1 - drag * Time.fixedDeltaTime);

        // Raycast for collision
        RaycastHit hit;
        if (Physics.Raycast(transform.position, velocity.normalized, out hit, velocity.magnitude * Time.f
        {
            OnProjectileHit(hit);
        }

        transform.position += velocity * Time.fixedDeltaTime;
    }
}

```

## Hit Detection:

- Headshots: 2x damage multiplier

- Limb damage: Debuffs (arm hit = reduced melee damage, leg = slow)
- Armor penetration: Reduces damage based on armor rating vs. penetration value

## Network Optimization:

- Client-side prediction for projectile spawn
- Server authority for hit detection (anti-cheat)
- Interpolation for smooth visual trajectory on high latency

**Gravity Well:** "Sharpshooter" skill tree unlocks trick shots (ricochet targeting, explosive arrows, piercing shots hit 3 enemies).

---

## ⚔️ Phase 7: Extensive Weapon System

### Bottleneck Identified

Weapon diversity creates **build identity** and loot excitement. Need 100+ unique weapons with distinct stats/abilities.

### WeaponDatabase.cs

### Weapon Progression Tree:

#### Tier 1 - Primitive (Levels 1-10):

- Stick: 5 damage, fast attack speed
- Stone Axe: 12 damage, slow, chance to stun
- Wooden Bow: 8 damage, 20m range

#### Tier 2 - Medieval (Levels 10-25):

- Iron Sword: 25 damage, balanced
- Longsword: 35 damage, slow, cleave (hits 2 enemies)
- Rapier: 20 damage, fast, critical chance +15%
- Warhammer: 40 damage, very slow, armor shred
- Crossbow: 30 damage, 40m range, reload delay

### **Tier 3 - Exotic (Levels 25-40):**

- Katana: 45 damage, fast, bleed DoT
- Flail: 38 damage, ignores 50% block chance
- Chakram: 35 damage, thrown, returns, hits on throw + return
- Compound Bow: 42 damage, 60m range, charged shots

### **Tier 4 - Magical (Levels 40-60):**

- Flaming Sword: 55 + 15 fire damage, ignites targets
- Frostblade: 50 damage, 30% slow on hit
- Lightning Staff: 60 lightning damage, chain to 2 targets
- Poison Dagger: 30 damage + 50 poison DoT over 10s

### **Tier 5 - Technological (Levels 60-80):**

- Pulse Rifle: 70 damage, 3-round burst, 100m range
- Plasma Cutter: 65 damage, melts armor, 5m range
- Gauss Rifle: 120 damage, penetrates, 2s charge time
- Railgun: 200 damage, 200m range, 5s reload

## Tier 6 - Psionic (Levels 80-100):

- Mindbreaker: 80 psi damage, confuses target (attacks allies)
- Void Edge: 90 damage, teleports user behind target on hit
- Reality Anchor: 75 damage, prevents target from teleporting
- Thought Ripper: 100 damage, steals mana on hit

## Weapon Stats Schema:

csharp

```
public class WeaponData
{
    public string weaponId;
    public string displayName;
    public WeaponType type; // Melee, Ranged, Magic, Psionic
    public DamageType damageType; // Physical, Fire, Ice, Lightning, Psi
    public int minDamage;
    public int maxDamage;
    public float attackSpeed; // Attacks per second
    public float range;
    public int criticalChance; // Percentage
    public float criticalMultiplier;
    public WeaponAbility[] abilities; // On-hit effects
    public int levelRequirement;
    public string[] requiredAugmentations;
    public ModelRequest modelRequest; // For procedural generation
}
```

## Weapon Abilities Examples:

- Lifesteal: 15% damage healed

- Chain Lightning: 20% chance to arc to nearby enemy
- Explosive Rounds: 10% chance for AoE explosion
- Phase Shift: 5% chance to teleport 5m forward on hit
- Soul Capture: Kills grant +1% damage (stacks to 50%)

## **Crafting System Integration:**

- Weapon schematics found in dungeons
- Combine materials: Iron Ore + Fire Essence = Flaming Sword base
- Enchanting: Add gem sockets for stat boosts
- Weapon upgrading: +1 to +10 (each tier increases stats 10%)

**Gravity Well:** Legendary weapons with unique models/effects. "Excalibur" (requires 1000 honorable kills), "Anubis' Staff" (50+ necromancy spells cast), "Quantum Disruptor" (unlock all psionic augmentations).

---

## **Phase 8: Psionic Power System**

### **Bottleneck Identified**

Psionics offer **non-violent** and **creative problem-solving** options distinct from magic.

### **PsionicSystem.cs**

#### **Psi Points System:**

- Regenerates slowly (1 per 10 seconds)
- Maximum pool:  $50 + \text{Wisdom} * 2$
- Meditation ability: Regenerate 10 Psi over 5 seconds (channeled, interruptible)

## **Psionic Disciplines:**

### **Telekinesis:**

- Lift Object: Move physics objects, use as shield or throw
- Force Push: Knockback enemies, opens doors
- Crush: Deal damage by compressing target
- Orbital Strike: Levitate boulder, slam down for AoE

### **Telepathy:**

- Mind Read: Reveal enemy stats/weaknesses
- Suggestion: Force NPC to perform action (trade, flee, attack ally)
- Memory Wipe: Reset aggro, enemies forget you
- Hive Mind: Link with party, share vision/buffs

### **Clairvoyance:**

- Precognition: See 3 seconds into future (dodge attacks)
- Remote Viewing: Scry distant locations
- Aura Sight: Detect invisible enemies, see through illusions
- Temporal Echo: Leave a "save point", teleport back in 30s

### **Psychokinesis:**

- Pyrokinesis: Ignite objects/enemies with mind
- Cryokinesis: Freeze water, create ice bridges
- Electrokinesis: Overload electronics, recharge energy weapons

- Molecular Agitation: Phase through walls (high Psi cost)

## Astral Projection:

- Soul Walk: Leave body, explore as ghost (invisible, can't interact)
- Possess: Take control of weak-minded NPC (10 Psi/second drain)
- Dreamscape: Enter shared mental space for party communication
- Death Trance: Fake death, enemies ignore you

## Psionic Combat:

- Mind Blast: Cone AoE stun (3s duration)
- Ego Whip: Single-target high damage + confusion
- Id Insinuation: Plant fear, target flees
- Psychic Crush: Execute low-health enemies instantly

## Network Synchronization:

- Psionic effects create visual particle systems (brain waves, auras)
- Telekinesis syncs object physics state across clients
- Server validates Psi point consumption

**Gravity Well:** "Transcendent" prestige class at 100+ total Psi powers used. Unlocks "Psionic Storm" ultimate (costs 50 Psi, massive AoE damage + enemy disarm).

---



## Phase 9: Dungeon Generation System

### Bottleneck Identified

Handcrafted levels don't scale. Procedural dungeons provide **infinite replayability** and emergent challenges.

## DungeonGenerator.cs

### Generation Algorithm:

- BSP (Binary Space Partitioning) for room layout
- Delaunay triangulation for room connections
- Minimum spanning tree for main path
- Extra edges for loops/shortcuts

### Dungeon Parameters:

csharp

```
public class DungeonRequest
{
    public int floorCount; // Multi-level dungeons
    public DungeonTheme theme; // Crypt, Forest, Tech Lab, Void
    public int difficulty; // Affects enemy density, trap complexity
    public int sizeMultiplier; // Rooms count
    public string[] guaranteedLoot; // Boss drops
    public bool hasBoss;
}
```

### Room Types:

- Entrance: Safe zone, no enemies
- Combat: Enemy spawns, loot chests
- Puzzle: Lever/pressure plate mechanics, no combat

- Treasure: High-value loot, trapped
- Mini-Boss: Elite enemy, checkpoint
- Boss: Final encounter, locked until all keys found
- Secret: Hidden passage, requires specific augmentation/spell

## **Enemy Spawning:**

- Difficulty scaling: floor 1 = level 10 enemies, floor 10 = level 100
- Enemy composition: 70% trash mobs, 20% elites, 10% casters
- Patrol routes: A\* pathfinding between waypoints
- Aggro linking: Pulling one enemy alerts nearby pack

## **Environmental Hazards:**

- Spike traps: Periodic activation, telegraphed
- Fire jets: Area denial
- Poison gas: DoT damage, requires antidote or Constitution save
- Crumbling floor: Fall to lower level
- Laser grids: Requires acrobatics check or Invisibility

## **Loot Distribution:**

- Trash mobs: 20% drop rate (consumables, materials)
- Elites: 50% drop rate (uncommon weapons/armor)
- Mini-bosses: 100% drop rate (rare items, augmentation parts)
- Boss: Guaranteed legendary + schematic + cosmetic

## Procedural Model Integration:

csharp

```
// Generate dungeon-specific enemy models
ModelRequest orcRequest = new ModelRequest
{
    Type = ModelType.Creature,
    Style = ModelStyle.Fantasy,
    Parameters = new Dictionary<string, float>
    {
        {"bulk", 1.5f}, // Muscular
        {"height", 2.2f}, // Tall
        {"tusks", 1.0f} // Orc-specific
    }
};
```

## Network Architecture:

- Server generates dungeon seed, sends to clients
- Clients generate identical dungeon locally (deterministic)
- Server tracks doors, chests, enemy states
- Packet: **DungeonSyncData** (opened doors, looted chests)

**Gravity Well:** "Dungeon Master" title for clearing 100 procedural dungeons. Unlocks ability to craft custom dungeon keys (control theme, difficulty, guaranteed loot).

---

## Phase 10: Integration & Polish

### System Interconnections:

#### CharacterCreation → Augmentations:

- Race affects augmentation compatibility (Elf gets magic-focused augs, Dwarf gets mining/crafting boosts)

#### Augmentations → Weapons:

- Arm augmentation "Weapon Mount" allows dual-wielding 2H weapons
- Neural augmentation "Targeting Computer" adds laser sight to ranged weapons

#### Magic → Psionics:

- Hybrid builds possible but difficult (separate resource pools)
- "Arcane Psi" prestige class merges both (unlocks Spellblade stance)

#### AuctionHouse → Crafting:

- Materials sold on AH drive weapon prices
- Legendary schematics tradeable (high value items)

#### Dungeons → Everything:

- Best source of augmentation parts, spell scrolls, weapon schematics, crafting materials

#### ProceduralModels → All Systems:

- Every new race generates unique model

- Every weapon generates unique mesh
  - Every armor piece generates unique overlay
  - Augmentations add visual cybernetics to base model
- 

## Technical Architecture: Modular Integration

All new systems follow the Quantum Mechanic philosophy:

### Event-Driven:

csharp

```
public class AugmentationManager : MonoBehaviour
{
    public event Action<string> OnAugmentationInstalled;
    public event Action<float> OnBioelectricEnergyChanged;
}

// EconomyManager subscribes
AugmentationManager.Instance.OnAugmentationInstalled += (augId) =>
{
    EconomyManager.Instance.SpendCurrency(GetAugCost(augId));
};
```

### Network Packets:

csharp

```
public enum PacketType : byte
{
    // Existing...
    Augmentation = 10,
    SpellCast = 11,
    AuctionBid = 12,
    ProjectileSpawn = 13,
    PsionicEffect = 14,
    DungeonGeneration = 15
}
```

## Save System Integration:

csharp

```
[Serializable]
public class PlayerData
{
    // Existing fields...
    public string[] installedAugmentations;
    public string[] learnedSpells;
    public string characterRace;
    public string characterClass;
    public int[] attributeScores; // STR, DEX, CON, INT, WIS, CHA
    public string[] activeWeapons; // Primary, Secondary
    public string[] psionicPowers;
}
```

## 🚀 Implementation Priority

### Quarter 1: Foundation Systems

1. ProceduralModelFactory (enables all visual content)
2. CharacterCreationSystem (player investment)
3. WeaponDatabase (core combat variety)

**Quarter 2: Combat Depth** 4. MagicSystem (strategic combat layer) 5. RangedCombatPhysics (skill expression) 6. PsionicSystem (alternative playstyle)

**Quarter 3: Progression & Economy** 7. AugmentationManager (long-term progression) 8. AuctionHouseSystem (player economy) 9. DungeonGenerator (content pipeline)

**Quarter 4: Polish & Balance** 10. Cross-system integration testing 11. Network optimization for complex interactions 12. Procedural model refinement

---

## 🎨 Artistic Vision: The "Quantum Aesthetic"

**Core Principle:** Cyberpunk meets high fantasy meets cosmic horror.

### Visual Themes:

- **Cybernetic Augmentations:** Glowing circuit patterns (neon blue/cyan)
- **Magic Effects:** Arcane geometric fractals (purple/gold)
- **Psionic Manifestations:** Distorted reality ripples (pink/white)
- **Energy Weapons:** Plasma containment fields (green/orange)
- **Procedural Models:** Faceted low-poly style with vertex color gradients

### UI Philosophy:

- "No HUD" mode for immersion
- Diegetic UI (augmentation overlays appear on character model)

- Minimalist stat displays (bars only appear when values change)
- 

## The Gravity Well Effect

Each system is designed to create "**one more turn**" engagement:

- **Augmentations:** "Just need one more energy cell to afford Icarus"
- **Auction House:** "Let me check if that legendary dropped in price"
- **Dungeons:** "This seed might have the Excalibur schematic"
- **Magic/Psionics:** "One more spell scroll to complete the combo"
- **Character Builds:** "What if I respec to Elf Mage/Psion hybrid?"

The interconnected systems create **emergent gameplay loops** where progression in one area opens opportunities in others.

---

## Bootstrapper Extension

The `ProjectBootstrapper` will be extended with:

```
[MenuItem("Project/Initialize Advanced Features")]
public static void InitializeAdvancedFeatures()
{
    CreateAugmentationPrefabs();
    GenerateWeaponDatabase();
    SetupMagicEffects();
    CreateDungeonTemplates();
    // ... etc
}
```

**One-Click Setup Remains:** Generate all 100+ weapons, 50+ spells, 20+ augmentations, all procedurally with visual variations.

---

**This roadmap transforms Quantum Mechanic from a functional demo into a deep, replayable Mini-MORPG with AAA-level system complexity using only pure Unity + C#.**