TinyWorld Economy Guide & Technical Overview

Status

Draft specification for the TinyWorld token, gold, island, asset, and marketplace economy.

This document has two audiences:

1. **Players and community members** who are new to TinyWorld or new to crypto.
2. **Crypto-native users, developers, and partners** who want to understand how the on-chain and in-game economy works.

This guide explains the intended relationship between:

- `$TINYWORLD`, the public Solana token.
- `GOLD`, the non-withdrawable in-game spending allowance.
- `ISLANDS`, scarce ownable world assets.
- `NFTs / on-chain assets`, player-owned game items.
- `Marketplace trading`, both inside and outside the game.
- `Technical integration`, assuming a TypeScript / React / Solana application.

Part 1 - Player Overview

1. What is TinyWorld?

TinyWorld is a game world made of small floating islands, resources, characters, items, and player-driven economies.

Players can explore, build, mine, craft, trade, and own parts of the world.

TinyWorld uses two different types of value:

Type	What it is	Can it be traded outside the game?
`$TINYWORLD`	Public Solana token	Yes
`GOLD`	In-game spending power	No
`ISLAND NFTs`	Ownable land/islands	Yes
`Item NFTs`	Optional on-chain game assets	Yes, if enabled
Internal resources	Wood, ore, crystal, energy, etc.	No, unless converted into an on-chain item

The core idea is simple:

`$TINYWORLD` is the external token.
`GOLD` is gameplay spending power.
`ISLANDS` and selected assets can be player-owned and tradable.

---

2. What is `$TINYWORLD`?

`$TINYWORLD` is the public Solana token connected to the TinyWorld ecosystem.

Players may be able to:

- Hold `$TINYWORLD` in their own Solana wallet.
- Send `$TINYWORLD` wallet-to-wallet.
- Trade `$TINYWORLD` on supported Solana exchanges or swap platforms.
- Use `$TINYWORLD` for TinyWorld-related purchases.
- Use `$TINYWORLD` in the official TinyWorld marketplace.
- Use `$TINYWORLD` to unlock in-game benefits, depending on game rules.

The important point:

> Players hold `$TINYWORLD` in their own wallets. TinyWorld does not need to hold player funds for them.

---

3. What is GOLD?

`GOLD` is TinyWorld's in-game spending allowance.

GOLD is used for gameplay actions such as:

- Building.
- Crafting.
- Upgrading.
- Speeding up actions.
- Unlocking cosmetics.
- Buying internal game services.
- Paying internal gameplay fees.
- Participating in game systems.

GOLD is **not** designed to be a cash balance.

GOLD:

- Cannot be withdrawn.
- Cannot be redeemed from TinyWorld for money.
- Cannot be exchanged directly for SOL, USDC, or `$TINYWORLD`.
- Has no fixed real-world value.
- Is not intended to be a bank balance, deposit, or stored-value account.

The clean principle is:

`$TINYWORLD` has market value.
`GOLD` has game utility.

---

4. How does holding `$TINYWORLD` affect GOLD?

TinyWorld may calculate a player's GOLD allowance based on their `$TINYWORLD` wallet status.

Example:

```
A player holds 10,000 $TINYWORLD.
The game gives them 1,000 GOLD spending allowance for the current cycle.
```

If the player later sends or sells some `$TINYWORLD`, their future GOLD allowance may decrease.

Example:

```
Alice holds 10,000 $TINYWORLD.
Alice has 1,000 GOLD allowance.

Alice sends 5,000 $TINYWORLD to Bob.

Alice now holds 5,000 $TINYWORLD.
Alice's future GOLD allowance may reduce.

Bob now holds more $TINYWORLD.
Bob's future GOLD allowance may increase.
```

This does not mean GOLD itself has a cash value.

It means the player's wallet status can unlock gameplay power.

---

5. Does GOLD go up and down in value?

The preferred wording is:

> GOLD does not have a market price.
> A player's GOLD allowance may change based on their wallet status and gameplay status.

Things that may affect GOLD allowance:

- Amount of `$TINYWORLD` held.
- Whether `$TINYWORLD` is locked for gameplay access.
- Island ownership.
- Season rank.
- Quest progress.

- Guild membership.
- Special NFTs or items.
- Gameplay multipliers.
- GOLD already spent during the current cycle.

Things that should **not** directly affect GOLD allowance:

- `$TINYWORLD` fiat price.
- Market cap.
- DEX liquidity.
- USD value.
- SOL price.

The safer design is:

```
10,000 $TINYWORLD = a defined gameplay tier
```

Not:

```
£100 worth of $TINYWORLD = a cash-equivalent amount of GOLD
```

This keeps TinyWorld's in-game economy stable and avoids treating GOLD like money.

---

6. Suggested GOLD model

TinyWorld can use a tiered allowance system.

Example:

`$TINYWORLD` Held	Tier	Weekly GOLD Allowance
1,000	Bronze	100 GOLD
10,000	Silver	500 GOLD
50,000	Gold	1,500 GOLD
100,000	Mythic	2,500 GOLD

Alternative model:

```
GOLD allowance = square root of $TINYWORLD held x multiplier
```

This prevents whales from completely dominating the economy while still rewarding larger holders.

---

7. What happens when GOLD is spent?

GOLD should work like an allowance or gameplay energy system.

Example:

```
Alice has 1,000 GOLD allowance this week.
Alice spends 300 GOLD on upgrades.
Alice has 700 GOLD remaining.
```

If Alice sells some `$TINYWORLD` during the cycle, the game may reduce her remaining available allowance.

Example:

```
Alice originally had 1,000 GOLD allowance.
Alice spent 300 GOLD.
Alice sells half her $TINYWORLD.
Her new allowance is 500 GOLD.
She has 200 GOLD remaining for the cycle.
```

Already-spent GOLD should not usually be clawed back.

Items already bought should not vanish just because a wallet balance changed.

The better player experience is:

> Wallet changes affect future or remaining allowance, not already-completed actions.

---

8. Can players send `$TINYWORLD` directly to each other?

Yes.

Because `$TINYWORLD` is a normal Solana token, players can send it wallet-to-wallet.

Example:

```
Alice sends 10,000 $TINYWORLD to Bob.
Alice's wallet balance goes down.
Bob's wallet balance goes up.
TinyWorld reads the updated wallet balances.
Alice's future GOLD allowance may go down.
Bob's future GOLD allowance may go up.
```

This can happen:

- Outside the game.
- Through a wallet.
- Through a Solana swap or marketplace.
- Through a TinyWorld interface that asks the wallet to sign a transaction.

TinyWorld does not need to hold the tokens.

---

9. Does TinyWorld earn fees when players send `$TINYWORLD`?

Not automatically.

There are different types of fees:

Solana network fee

A normal Solana transaction fee goes to the network, not TinyWorld.

Marketplace fee

If players trade through the official TinyWorld marketplace, TinyWorld may charge a marketplace fee.

Example:

```
Alice sells an island to Bob for 50,000 $TINYWORLD.
TinyWorld marketplace fee = 5%.
Alice receives 47,500 $TINYWORLD.
TinyWorld treasury receives 2,500 $TINYWORLD.
Bob receives the island NFT.
```

Token transfer fee

If `$TINYWORLD` is created using a Solana token standard that supports transfer fees, the token itself can charge a fee on transfers.

Example:

```
Alice sends 1,000 $TINYWORLD.
Bob receives 990 $TINYWORLD.
10 $TINYWORLD is withheld as a token transfer fee.
```

This is possible, but it should be used carefully.

Transfer taxes can annoy users, reduce exchange support, and make the token feel less open.

Recommended default:

> Let wallet-to-wallet transfers remain simple.
> Earn through game actions, primary sales, marketplace fees, minting fees, upgrades, and optional premium systems.

Part 2 - Islands and Land Ownership

10. What are islands?

Islands are scarce ownable world assets in TinyWorld.

Example:

```
There are 20 original floating islands.
Each island can be owned by a player wallet.
Each island has a unique identity.
Each island may have gameplay rights, settings, and economic rules.
```

An island may include:

- World position.
- Name.
- Visual style.
- Resource types.
- Mining zones.
- Build permissions.
- Tax settings.
- Access rules.
- Upgrade slots.
- Decoration slots.
- Marketplace permissions.

---

11. How are islands represented?

Each island should be represented as an on-chain asset, most likely an NFT or equivalent Solana asset.

The NFT proves ownership.

The game reads the blockchain to determine who owns each island.

Example:

```
Wallet A owns Island #7 NFT.
TinyWorld gives Wallet A island management controls for Island #7.
Wallet A sells Island #7 NFT to Wallet B.
TinyWorld sees Wallet B is now the owner.
Wallet B gets the island controls.
Wallet A loses those controls.
```

This means ownership is not just stored in TinyWorld's private database.

The public chain acts as the ownership ledger.

---

12. Initial island sales

At launch, TinyWorld may sell the first islands directly.

Recommended flow:

```
1. Player connects their Solana wallet.
2. Player chooses an island.
3. Player clicks Buy.
4. Wallet opens a transaction approval.
5. Player pays in SOL, USDC, or $TINYWORLD.
6. Payment goes to the TinyWorld treasury wallet.
7. Island NFT transfers or mints to the player wallet.
8. TinyWorld detects ownership.
9. Player receives island controls in-game.
```

The key point:

> The purchase should happen through a proper on-chain sale/mint contract or trusted marketplace flow, not through manual "send money and wait" transfers.

Avoid this:

```
Send money to this wallet and we will manually send your island later.
```

Prefer this:

```
Click Buy Island.
Approve transaction.
Payment and island transfer happen atomically.
```

Atomic means the payment and asset transfer happen together, or the transaction fails.

---

13. Initial sale formats

TinyWorld can sell islands in several ways.

Fixed-price sale

Each island has a fixed price.

Example:

```
Island #1 = 100,000 $TINYWORLD
Island #2 = 100,000 $TINYWORLD
Island #3 = 100,000 $TINYWORLD
```

Pros:

- Simple.
- Easy to explain.
- Good for first launch.

Cons:

- Bots or whales may buy quickly.
- Some islands may be more valuable than others.

Auction

Each island is auctioned.

Example:

```
Island #1 auction opens.
Players bid in $TINYWORLD or SOL.
Highest bidder wins.
Payment settles.
Island transfers.
```

Pros:

- Market discovers price.
- Better for scarce/high-value islands.

Cons:

- More complex.
- Can feel less accessible.

Whitelist sale

Only selected wallets can buy during an early phase.

Example allowlist groups:

- Early players.
- Token holders.
- Community contributors.
- Testers.
- Creators.
- Game builders.

Pros:

- Rewards real community.
- Reduces bot risk.

Cons:

- Requires allowlist management.
- Can create fairness concerns.

Hybrid model

Example:

```
5 islands reserved for community.
5 islands sold at fixed price.
5 islands auctioned.
5 islands kept by treasury for future events.
```

This is often the best structure because it avoids putting the entire world into one sale format.

---

14. Player-to-player island sales

After the initial sale, players can sell islands to each other.

There are two routes.

Official TinyWorld marketplace

```
Alice lists Island #7 for 50,000 $TINYWORLD.
Bob buys it through TinyWorld.
Marketplace contract transfers payment and island.
TinyWorld takes a marketplace fee.
Bob becomes the new island owner.
```

This is the preferred user experience.

TinyWorld can provide:

- Clear listings.
- Verified island data.
- Game stats.
- Tax settings.
- Resource history.
- Marketplace fees.
- Safer settlement.
- Better support.

Direct wallet-to-wallet sale

Players may also transact outside the official marketplace.

Example:

```
Bob sends Alice 50,000 $TINYWORLD.
Alice sends Bob Island #7 NFT.
TinyWorld sees Bob now owns the NFT.
Bob gets island controls.
```

TinyWorld may not receive a fee from this type of trade unless the asset standard, marketplace, or token mechanics enforce one.

This is part of open ownership.

If players own assets in their own wallets, they can move them.

---

15. What happens when an island is sold?

When an island NFT changes owner:

```
Old owner loses island controls.
New owner gains island controls.
Island settings may persist.
Pending taxes/rewards need a defined settlement rule.
```

Recommended settlement rule:

```
Ownership changes apply from the next game tick or season checkpoint.
Unclaimed internal resources earned before sale belong to the previous owner.
Future resources after sale belong to the new owner.
```

This prevents confusion.

Part 3 - Island Taxes and Resource Economies

16. What are island taxes?

Island owners may be able to set gameplay taxes on activity that happens on their island.

Example:

```
A miner gathers 100 ORE on Island #7.
Island tax = 10%.
Miner receives 90 ORE.
Island owner receives 10 ORE.
```

The safest version is:

> Taxes are paid in internal game resources, not directly in cash-value tokens.

Examples:

- ORE.
- WOOD.
- CRYSTAL.

- ENERGY.
- GOLD.
- Build points.
- Crafting materials.

These resources can be used inside the game.

---

17. Why not pay island tax directly in `$TINYWORLD`?

It is possible, but it increases risk.

If islands generate `$TINYWORLD` income automatically, they start to look like yield-generating assets.

That may create:

- Regulatory risk.
- Tax complexity.
- Securities-style concerns.
- "Passive income" expectations.
- Speculative land flipping.
- Player backlash if returns fall.

Safer wording:

```
Island owners receive in-game resources from gameplay activity.
```

Riskier wording:

```
Buy islands to earn passive token income.
```

TinyWorld should avoid marketing islands as investments.

---

18. Recommended island tax model

The recommended model:

```
Island owners set resource tax rates.
Players mine or gather resources.
Tax is paid in internal resources.
Island owners use resources to upgrade, craft, decorate, or operate their islands.
Rare crafted assets may optionally become NFTs.
Those NFTs can be traded player-to-player.
```

This creates an indirect value path:

```
Own island
↓
Collect in-game resources
```

```
↓
Upgrade/craft/build
↓
Create valuable game assets
↓
Optionally sell assets to other players
```

This is better than:

```
Own island
↓
Automatically receive tradable token yield
```

---

19. Tax limits

TinyWorld should cap tax rates to avoid abuse.

Example:

```
Minimum tax: 0%
Maximum tax: 20%
Default tax: 5%
Tax change cooldown: 24 hours or one game cycle
```

Other safety rules:

- Players must see tax rates before mining.
- Tax changes should not apply mid-action.
- New tax rates apply from the next cycle.
- Abusive islands can be ignored by players.
- Game may apply maximum caps per resource type.

Part 4 - NFTs and the On-Chain Ledger

20. What is on-chain?

The blockchain should store or verify important ownership events.

On-chain:

- `$TINYWORLD` token balances.
- Island ownership.
- NFT item ownership.
- Marketplace sales.
- Primary sale purchases.
- Optional royalty/fee records.

- Optional locking/escrow states.
- Optional crafting/minting events.

Off-chain/in-game:

- GOLD allowance.
- Internal resources.
- XP.
- Quest progress.
- Building placement.
- Local island state.
- Combat data.
- Temporary rewards.
- Cooldowns.
- Session data.
- Player preferences.

The rule:

> Put ownership and settlement on-chain.
> Keep high-frequency gameplay off-chain.

---

21. Why not put everything on-chain?

Games need fast interactions.

Mining, movement, crafting, combat, and building can happen frequently.

Putting every action on-chain would be:

- Slower.
- More expensive.
- More annoying for players.
- Worse for UX.
- Harder to scale.

So TinyWorld should use a hybrid model:

```
Blockchain = ownership and settlement
Game backend = gameplay state and simulation
React client = interface and wallet signing
```

---

22. NFTs as ownership keys

An NFT can act as a key.

Example:

```
If wallet owns Island #12 NFT:
  show Island Admin Panel
else:
  hide Island Admin Panel
```

The NFT does not need to contain every detail of the island.

It only needs to prove ownership.

The game database can store extended state:

```
{
  "islandId": "island-12",
  "ownerWallet": "9x...",
  "name": "Crystal Hollow",
  "taxRate": 0.08,
  "resourceProfile": ["ore", "crystal"],
  "level": 3,
  "buildSlots": 42
}
```

The chain says who owns the island.

The backend says what is happening on the island.

---

23. Item NFTs

Not every item should be an NFT.

Use NFTs for:

- Rare items.
- Land.
- Founder assets.
- Special cosmetics.
- Unique machines.
- High-value crafted assets.
- Transferable player creations.

Do not use NFTs for:

- Every piece of wood.
- Every ore unit.
- Every basic sword.
- Every temporary boost.
- Every low-value consumable.

Recommended item categories:

Item Type	On-chain?
Basic resources	No
Common consumables	No
Normal crafted tools	Usually no
Rare cosmetics	Maybe
Unique land/islands	Yes
Founder assets	Yes
Player-created rare assets	Maybe
High-value tradeable machines	Maybe

---

24. Marketplace sales

Marketplace sales should be atomic.

Example sale:

```
Alice lists Island #7 for 50,000 $TINYWORLD.
Bob clicks Buy.
Bob signs transaction.
Transaction transfers:
  50,000 $TINYWORLD from Bob
  marketplace fee to TinyWorld treasury
  remainder to Alice
  Island #7 NFT to Bob
```

After the transaction confirms:

```
TinyWorld indexer detects ownership change.
Backend updates owner mapping.
Bob gets island controls.
Alice loses island controls.
```

Part 5 - Economy Safety Rules

25. Do not promise redemption

TinyWorld should avoid saying:

```
GOLD can be cashed out.
GOLD is backed by $TINYWORLD.
GOLD has a fixed dollar value.
TinyWorld guarantees liquidity.
TinyWorld will buy assets back.
Island owners earn passive income.
```

Preferred wording:

```
GOLD is non-withdrawable gameplay spending power.
$TINYWORLD is a public token with market risk.
Island and asset prices are determined by players and external markets.
TinyWorld does not guarantee resale value.
```

---

26. Avoid bank-like mechanics

Avoid:

```
Player deposits token.
TinyWorld credits redeemable balance.
Player later withdraws from TinyWorld.
TinyWorld manages pooled reserves.
TinyWorld guarantees conversion.
```

Prefer:

```
Player holds token in wallet.
Game reads wallet status.
Player gets gameplay allowance.
Player owns NFTs directly.
Players trade assets peer-to-peer.
TinyWorld only settles official marketplace actions.
```

---

27. Risk disclosure

TinyWorld should clearly disclose:

- `$TINYWORLD` can go up or down.
- Liquidity may be limited.
- Market prices are not guaranteed.
- GOLD cannot be withdrawn.
- NFTs may lose value.
- Player-to-player trades outside TinyWorld are at player risk.
- Wallet security is the player's responsibility.
- TinyWorld does not provide financial advice.
- Game rules may evolve.
- On-chain transactions are generally irreversible.

Part 6 - Technical Architecture

28. Recommended stack

Frontend:

- TypeScript.
- React.
- Next.js or Vite.
- Solana wallet adapter.
- TanStack Query or SWR for data fetching.
- Zustand or Redux for local state.
- Tailwind/shadcn/ui for UI.

Backend:

- Node.js / TypeScript.
- PostgreSQL.
- Redis for caching/game session state.
- Worker/indexer service.
- RPC provider for Solana reads.
- WebSocket or server-sent events for live updates.
- Optional queue system for chain events.

Blockchain:

- Solana.
- SPL token or Token-2022 for `$TINYWORLD`.
- NFT standard for islands/items.
- Marketplace/sale contract or trusted protocol.
- Treasury wallet.
- Optional multisig for treasury/admin control.

---

29. Core services

Recommended services:

```
React App
  - wallet connect
  - game UI
  - marketplace UI
  - island dashboard
  - inventory UI

API Server
  - player profile
  - game state
  - GOLD calculation
```

```
  - island state
  - tax settings
  - marketplace metadata

Chain Indexer
  - reads token balances
  - reads NFT ownership
  - listens for transfers
  - updates backend projections

Game Engine / Simulation
  - mining
  - crafting
  - resource generation
  - island tax calculation
  - season ticks

Database
  - player profiles
  - wallet links
  - island state
  - resources
  - GOLD spend ledger
  - marketplace cache
  - audit logs
```

---

30. Source of truth

Use different sources of truth for different things.

Data	Source of Truth
`$TINYWORLD` balance	Solana
Island ownership	Solana
NFT item ownership	Solana
GOLD allowance	Backend calculation
GOLD spent	Backend ledger
Internal resources	Backend ledger
Quest progress	Backend
Island tax rate	Backend, optionally anchored on-chain
Marketplace settlement	Solana
Marketplace display/cache	Backend

---

31. Wallet connection

The React app should let users connect a Solana wallet.

Basic responsibilities:

- Detect wallet.
- Connect wallet.
- Show wallet address.
- Request signatures for transactions.
- Read token balances.
- Read NFT ownership.
- Submit transactions.
- Show confirmation state.

Pseudo-flow:

```
connectWallet()
loadPlayerProfile(walletAddress)
loadTokenBalance(walletAddress, TINYWORLD_MINT)
loadOwnedIslands(walletAddress)
calculateGoldAllowance(walletAddress)
renderGameState()
```

---

32. Player authentication

Use wallet signature authentication.

Flow:

```
1. Player connects wallet.
2. Backend generates nonce.
3. Player signs message with wallet.
4. Backend verifies signature.
5. Backend creates session/JWT.
6. Player is authenticated as that wallet.
```

Example signed message:

```
Sign in to TinyWorld

Wallet: <wallet>
Nonce: <random nonce>
Issued At: <timestamp>
```

Do not authenticate users only by letting them type a wallet address.

They must prove wallet ownership by signing a message.

---

33. GOLD calculation

GOLD should be calculated from wallet and game state.

Example inputs:

```ts
type GoldInputs = {
  tinyworldHeld: bigint;
  lockedTinyworld: bigint;
  islandCount: number;
  seasonRank: number;
  nftBonuses: string[];
  spentThisCycle: number;
};
```

Example result:

```ts
type GoldAllowance = {
  cycleId: string;
  baseAllowance: number;
  bonuses: {
    source: string;
    amount: number;
  }[];
  totalAllowance: number;
  spent: number;
  available: number;
};
```

Example logic:

```ts
function calculateGoldAllowance(input: GoldInputs): GoldAllowance {
  const base = getTierAllowance(input.tinyworldHeld);

  const islandBonus = input.islandCount > 0 ? Math.floor(base * 0.1) : 0;
  const rankBonus = getRankBonus(input.seasonRank);

  const total = base + islandBonus + rankBonus;
  const available = Math.max(0, total - input.spentThisCycle);

  return {
    cycleId: getCurrentCycleId(),
    baseAllowance: base,
    bonuses: [
      { source: "island_owner", amount: islandBonus },
      { source: "season_rank", amount: rankBonus }
    ],
    totalAllowance: total,
    spent: input.spentThisCycle,
```

```
    available
  };
}
```

---

34. GOLD ledger

Even though GOLD is not on-chain, it should still have a proper internal ledger.

Do not store only a mutable number.

Use append-only events.

Example:

```
type GoldLedgerEvent =
  | {
      type: "ALLOWANCE_RECALCULATED";
      wallet: string;
      cycleId: string;
      totalAllowance: number;
      reason: string;
      createdAt: string;
    }
  | {
      type: "GOLD_SPENT";
      wallet: string;
      cycleId: string;
      amount: number;
      action: string;
      referenceId: string;
      createdAt: string;
    }
  | {
      type: "GOLD_REFUNDED";
      wallet: string;
      cycleId: string;
      amount: number;
      reason: string;
      referenceId: string;
      createdAt: string;
    };
```

This makes disputes easier.

You can explain exactly why a player had a certain amount of available GOLD.

---

35. Island data model

Example:

```ts
type Island = {
  id: string;
  nftMint: string;
  ownerWallet: string;
  name: string;
  coordinates: {
    x: number;
    y: number;
    z: number;
  };
  taxRate: number;
  resourceTypes: ResourceType[];
  level: number;
  buildSlots: number;
  status: "active" | "locked" | "seasonal" | "retired";
  createdAt: string;
  updatedAt: string;
};
```

---

36. Island ownership sync

The chain indexer should keep island ownership updated.

Flow:

```
1. Indexer watches known island NFT mints.
2. Transfer event occurs.
3. Indexer identifies new owner wallet.
4. Backend updates island.ownerWallet.
5. Backend emits game event.
6. UI updates island controls.
```

Important:

- Use chain ownership as final authority.
- Cache ownership for performance.
- Revalidate before sensitive actions.
- Handle transfers outside your marketplace.
- Handle failed or delayed indexer updates gracefully.

---

37. Island admin permissions

Before allowing island management actions, verify ownership.

Example:

```
async function canManageIsland(wallet: string, islandId: string) {
  const island = await db.island.findUnique({ where: { id: islandId } });

  if (!island) return false;

  const currentOwner = await chain.getNftOwner(island.nftMint);

  return normalizeWallet(currentOwner) === normalizeWallet(wallet);
}
```

Use cached owner data for normal UI.

Use fresh chain verification for important actions.

Important actions include:

- Changing tax rate.
- Selling through official marketplace.
- Renaming island.
- Withdrawing internal tax resources.
- Upgrading island.
- Changing access rules.

---

38. Tax calculation

Example mining event:

```
type MiningEvent = {
  minerWallet: string;
  islandId: string;
  resource: "ORE" | "WOOD" | "CRYSTAL";
  grossAmount: number;
};
```

Tax logic:

```
function applyIslandTax(event: MiningEvent, island: Island) {
  const taxRate = clamp(island.taxRate, 0, 0.2);
  const taxAmount = Math.floor(event.grossAmount * taxRate);
  const minerAmount = event.grossAmount - taxAmount;

  return {
    miner: {
```

```
      wallet: event.minerWallet,
      resource: event.resource,
      amount: minerAmount
    },
    islandOwner: {
      wallet: island.ownerWallet,
      resource: event.resource,
      amount: taxAmount
    }
  };
}
```

Use an internal resource ledger.

Example:

```
type ResourceLedgerEvent = {
  id: string;
  wallet: string;
  resource: ResourceType;
  amount: number;
  direction: "credit" | "debit";
  reason: string;
  referenceId: string;
  createdAt: string;
};
```

---

39. Primary island sale technical flow

Recommended flow:

```
1. Admin configures island sale.
2. Frontend displays available islands.
3. Player connects wallet.
4. Player selects island.
5. Backend creates sale intent.
6. Frontend builds transaction.
7. Player signs transaction.
8. Transaction transfers payment to treasury.
9. Transaction mints/transfers island NFT to buyer.
10. Backend verifies transaction.
11. Island ownership is updated.
12. Player receives island controls.
```

Sale intent example:

```
type IslandSaleIntent = {
```

```
  id: string;
  islandId: string;
  buyerWallet: string;
  priceAmount: string;
  priceMint: "SOL" | "USDC" | "TINYWORLD";
  treasuryWallet: string;
  expiresAt: string;
  status: "created" | "submitted" | "confirmed" | "expired" | "failed";
};
```

Do not mark an island sold until the transaction is confirmed.

---

40. Marketplace technical flow

Listing:

```
1. Seller owns island NFT.
2. Seller creates listing.
3. Marketplace verifies ownership.
4. Listing appears in marketplace.
```

Purchase:

```
1. Buyer clicks Buy.
2. Transaction is built.
3. Buyer signs transaction.
4. Payment transfers from buyer.
5. Marketplace fee transfers to treasury.
6. Net proceeds transfer to seller.
7. NFT transfers to buyer.
8. Indexer updates ownership.
9. Game updates controls.
```

Listing type:

```
type MarketplaceListing = {
  id: string;
  assetType: "island" | "item";
  assetMint: string;
  sellerWallet: string;
  priceAmount: string;
  priceMint: "SOL" | "USDC" | "TINYWORLD";
  feeBps: number;
  status: "active" | "sold" | "cancelled" | "expired";
  createdAt: string;
  updatedAt: string;
};
```

---

41. Official marketplace vs external marketplace

TinyWorld should support both.

Official marketplace:

- Better UX.
- Verified assets.
- Game metadata.
- TinyWorld fee support.
- Safer settlement.
- Better support.

External marketplace:

- Open ecosystem.
- Players can trade freely.
- TinyWorld may not receive fees.
- TinyWorld still reads ownership after transfer.

Rule:

> The game follows ownership, wherever the transfer happened.

---

42. Token transfer fee decision

TinyWorld must decide whether `$TINYWORLD` should include a transfer fee.

Option A - No transfer fee

Pros:

- Cleaner token UX.
- Better composability.
- Easier exchange support.
- Less hostile to users.
- Better for wallet-to-wallet use.

Cons:

- TinyWorld does not earn from direct transfers.

Option B - Transfer fee

Pros:

- TinyWorld can earn from token movement.
- Wallet-to-wallet transfers can fund treasury.

Cons:

- More complex.
- Some tools may not support it well.
- Users may dislike it.
- May hurt liquidity and market maker interest.
- Needs clear disclosure.

Recommended default:

```
No transfer fee at launch.
Earn from game actions, primary sales, marketplace fees, minting, upgrades, cosmetics, and c
```

---

43. Treasury design

TinyWorld should have a clearly defined treasury wallet.

Treasury may receive:

- Primary island sale proceeds.
- Marketplace fees.
- Minting fees.
- Upgrade fees.
- Premium action fees.
- Cosmetic sales.
- Optional crafting fees.

Treasury should ideally be controlled by a multisig, not a single hot wallet.

Treasury policy should define:

- Who controls funds.
- What funds are used for.
- Whether any token buybacks/burns exist.
- Whether fees are retained, burned, or recycled into the game.
- How treasury transactions are disclosed.

Avoid promising treasury actions unless they are fixed and legally reviewed.

---

44. Recommended database tables

Core tables:

```
players
wallet_links
gold_ledger_events
resource_ledger_events
islands
```

```
island_ownership_snapshots
island_tax_settings
nft_assets
marketplace_listings
marketplace_sales
chain_transactions
sale_intents
game_cycles
audit_logs
```

Example `players`:

```
CREATE TABLE players (
  id UUID PRIMARY KEY,
  primary_wallet TEXT UNIQUE NOT NULL,
  username TEXT,
  created_at TIMESTAMP NOT NULL DEFAULT now(),
  updated_at TIMESTAMP NOT NULL DEFAULT now()
);
```

Example `islands`:

```
CREATE TABLE islands (
  id TEXT PRIMARY KEY,
  nft_mint TEXT UNIQUE NOT NULL,
  owner_wallet TEXT,
  name TEXT NOT NULL,
  tax_rate NUMERIC NOT NULL DEFAULT 0.05,
  level INTEGER NOT NULL DEFAULT 1,
  status TEXT NOT NULL DEFAULT 'active',
  created_at TIMESTAMP NOT NULL DEFAULT now(),
  updated_at TIMESTAMP NOT NULL DEFAULT now()
);
```

Example `gold_ledger_events`:

```
CREATE TABLE gold_ledger_events (
  id UUID PRIMARY KEY,
  wallet TEXT NOT NULL,
  cycle_id TEXT NOT NULL,
  event_type TEXT NOT NULL,
  amount INTEGER NOT NULL,
  reason TEXT NOT NULL,
  reference_id TEXT,
  created_at TIMESTAMP NOT NULL DEFAULT now()
);
```

45. API endpoints

Suggested API routes:

```
POST /api/auth/nonce
POST /api/auth/verify

GET /api/me
GET /api/me/wallet
GET /api/me/gold
GET /api/me/resources
GET /api/me/islands
GET /api/me/assets

GET /api/islands
GET /api/islands/:id
POST /api/islands/:id/tax-rate
POST /api/islands/:id/rename

GET /api/marketplace/listings
POST /api/marketplace/listings
POST /api/marketplace/listings/:id/cancel
POST /api/marketplace/listings/:id/buy-intent

GET /api/sales/islands
POST /api/sales/islands/:id/buy-intent
POST /api/transactions/confirm

POST /api/game/mine
POST /api/game/craft
POST /api/game/spend-gold
```

---

46. Frontend screens

Recommended screens:

Wallet connect

- Connect wallet.
- Sign in.
- Show wallet address.
- Show `$TINYWORLD` balance.
- Show GOLD allowance.

Player dashboard

- `$TINYWORLD` held.
- Current GOLD allowance.
- GOLD spent this cycle.
- Available GOLD.
- Owned islands.
- Owned assets.
- Recent activity.

Island map

- Show all islands.
- Ownership state.
- For-sale status.
- Tax rates.
- Resources.
- Activity.

Island detail

- Island owner.
- Resource profile.
- Current tax.
- Upgrade level.
- Buy/listing status.
- Marketplace actions.

Island admin

- Rename island.
- Set tax rate.
- Manage permissions.
- View tax income.
- Upgrade island.
- Configure build zones.

Marketplace

- Buy islands.
- Sell islands.
- Buy/sell assets.
- Filter by resource type, price, owner, rarity.
- Show official/verified status.

---

47. Indexer requirements

The indexer should track:

- `$TINYWORLD` balances for active players.
- Island NFT ownership.
- Item NFT ownership.
- Marketplace sales.
- Primary sale transactions.
- Transfer events.
- Failed/unknown transactions.
- Wallet-to-wallet island transfers.

Indexer strategy:

```
For active players:
  refresh balances on login, marketplace action, and cycle tick.

For islands:
  watch known NFT mints and update owner mapping.

For marketplace:
  verify every submitted transaction before updating backend state.

For external transfers:
  periodically re-sync ownership and detect changes.
```

---

48. Confirmation policy

Do not unlock critical ownership features on unconfirmed transactions.

Suggested states:

```
created
submitted
processed
confirmed
finalized
failed
expired
```

For UX, you can show optimistic pending state.

For actual ownership rights, wait for confirmed/finalized state depending on risk tolerance.

---

49. Security requirements

Minimum requirements:

- Wallet signature auth.
- Nonce replay protection.
- Server-side transaction verification.
- Chain revalidation for sensitive actions.
- Treasury multisig.
- Admin key separation.
- Rate limits.
- Audit logs.
- Marketplace listing validation.
- Tax rate caps.
- No client-trusted balances.
- No client-trusted ownership.
- No manual sale fulfilment.
- Clear error states.

---

50. Abuse and exploit cases

TinyWorld must plan for:

- Players selling tokens after receiving GOLD.
- Players trying to double-spend GOLD.
- Players transferring island NFTs during pending actions.
- Players listing assets they no longer own.
- Marketplace bypass.
- Wash trading.
- Bot purchases during island sale.
- Fake NFTs pretending to be official assets.
- Phishing links.
- Failed transactions.
- RPC/indexer delays.
- Tax rate abuse.
- Game economy inflation.

Important rule:

> The backend must never trust the frontend for balances, ownership, or settlement.

Part 7 - Publishing-Friendly Summary

TinyWorld uses a hybrid game economy.

`$TINYWORLD` is the public Solana token that players can hold, send, and trade in their own wallets. Holding `$TINYWORLD` can unlock in-game benefits such as GOLD allowance, access tiers, island rights, or premium actions.

`GOLD` is not a cash balance. It is non-withdrawable gameplay spending power. Players can use GOLD inside TinyWorld, but they cannot redeem it from TinyWorld for money, SOL, USDC, or `$TINYWORLD`.

Islands are scarce ownable game assets. Each island can be represented by an on-chain NFT or similar Solana asset. When a wallet owns an island asset, the game gives that wallet control over the island. If the asset is sold or transferred, the game follows the new owner.

Players may trade islands and selected assets through the official TinyWorld marketplace or directly wallet-to-wallet. The official marketplace can provide better safety, verified data, and marketplace fees. External trades may happen without TinyWorld taking a fee, but the game can still recognize the new owner because ownership is on-chain.

Island owners may be able to set gameplay taxes on resources mined or gathered on their islands. The recommended model is for those taxes to be paid in internal game resources, not directly as token yield. This keeps the economy game-first rather than turning islands into passive income products.

The core design principle is:

```
$TINYWORLD = market token
GOLD = game utility
ISLANDS = ownable world assets
NFTs = optional player-owned items
Marketplace = player-to-player settlement
Backend = gameplay state
Blockchain = ownership ledger
```

TinyWorld does not need to hold player funds for this system to work. Players hold their own tokens and assets in their wallets. The game reads wallet status and ownership, then unlocks gameplay features accordingly.

Part 8 - Open Decisions

The following decisions should be made before production launch:

Token design

- SPL Token or Token-2022?
- Transfer fee or no transfer fee?
- Fixed supply or mint authority?

- Burn mechanics?
- Treasury allocation?
- Liquidity strategy?

GOLD design

- Wallet-held or locked `$TINYWORLD`?
- Linear, tiered, or square-root allowance?
- Daily, weekly, or seasonal cycle?
- Can unspent GOLD roll over?
- Can GOLD be gifted inside the game?
- Can GOLD be used in player-to-player trades, or only system actions?

Island sale design

- Fixed price, auction, whitelist, or hybrid?
- Payment in SOL, USDC, `$TINYWORLD`, or multiple?
- How many islands sold initially?
- How many reserved for treasury/events?
- Marketplace fee percentage?
- Royalty policy?

Island tax design

- Maximum tax rate?
- Tax paid in which resources?
- Tax change cooldown?
- Can islands block access?
- Do taxes apply instantly or next cycle?
- What happens to unclaimed taxes when an island is sold?

NFT design

- Which assets become NFTs?
- Which remain internal?
- Metadata standard?
- Update authority policy?
- Royalty policy?
- Compressed or standard NFTs?
- Marketplace compatibility?

Legal/compliance

- Public risk disclosures.
- Token marketing rules.
- No redemption language.

- No guaranteed returns.
- No passive income claims.
- Consumer protection review.
- Tax/accounting review.
- Jurisdiction-specific legal review.

Part 9 - Recommended MVP

The recommended first version:

```
1. Launch $TINYWORLD as a normal tradable Solana token.
2. Let players connect wallets.
3. Calculate GOLD allowance from $TINYWORLD quantity held.
4. Make GOLD non-withdrawable.
5. Sell a small number of island NFTs through an official sale flow.
6. Let the game read island ownership from chain.
7. Allow island owners to set internal resource taxes.
8. Keep taxes paid in internal resources.
9. Add official marketplace for islands/assets.
10. Charge marketplace fees only when users use the official marketplace.
```

Avoid in MVP:

```
GOLD redemption
Guaranteed token value
Automatic token yield from islands
Cash-equivalent balances
Company-held player deposits
Manual wallet sales
Complex transfer taxes
Over-promising treasury mechanics
```

The MVP should prove the fun loop first:

```
Hold token
Get gameplay power
Own island
Mine resources
Upgrade island
Craft assets
Trade assets
Build world
```

Then expand the economy only where the game actually needs it.

Part 10 - One-Sentence Explanation

TinyWorld lets players hold a real Solana token, use that wallet status to unlock non-withdrawable in-game GOLD, own scarce island assets on-chain, and trade selected assets peer-to-peer without TinyWorld needing to custody player funds or promise cash redemption.