# **UML Design Documentation**

## **NASA Farm Navigators**

Version: 2.0

Date: September 2025

## 1. Class Diagram - Core System Architecture

GameEngine
- instance: GameEngine
- gameState: GameState
- dataManager: NASADataIntegrator
- resolutionManager: ResolutionManager
- depthAnalyzer: SoilDepthAnalyzer
- contextAdapter: FarmContextAdapter
+ getInstance(): GameEngine
+ initialize(config: GameConfig): void
+ update(deltaTime: number): void
+ switchResolution(resolution: number): void
+ adaptToContext(context: FarmContext): void
manages ▼
NASADataIntegrator
- earthdataAuth: EarthdataAuth
- appEEARSClient: AppEEARSClient
- cropCASMAClient: CropCASMAClient
- worldviewClient: WorldviewClient
- dataCache: DataCache
- resolutionMap: Map <datatype, resolution[]=""></datatype,>
+ fetchSMAPData(depth: string): SMAPData
+ fetchNDVI(resolution: number): NDVIData
+ fetchGPMPrecipitation(): GPMData
+ getDataAtResolution(type: string, res: number)
+ validateDataAccuracy(data: any): boolean

```
+ explainDataLimitations(data: any): string[]
            uses
       ResolutionManager
- availableResolutions: Resolution[]
- currentResolution: number
- resolutionImpacts: Map<number, Impact>
+ switchResolution(meters: number): void
+ compareResolutions(r1: number, r2: number)
+ demonstratePixelSize(resolution: number)
| + getDetectionCapability(res: number): string[] |
+ educateResolutionTrade offs(): Tutorial
                                    SoilDepthAnalyzer
- surfaceData: SoilData (0-5cm)
| - rootZoneData: SoilData (0-100cm)
| - deepSoilData: SoilData (100cm+)
- cropRootDepths: Map<Crop, number>
+ analyzeMoistureByDepth(depth: number): number
+ recommendIrrigationDepth(crop: Crop): string
+ visualizeDepthProfile(): DepthVisualization
+ teachDepthImportance(): EducationalContent
       FarmContextAdapter
- context: FarmingContext
- scale: FarmScale
- resources: ResourceAvailability
- constraints: Constraint[]
| + adaptToSmallholder(): GameplayModifiers
| + adaptToIndustrial(): GameplayModifiers
+ adjustDataGranularity(scale: FarmScale): void
| + customizeObjectives(context: FarmingContext) |
```

```
Farm
- farmld: UUID
I - location: GeoCoordinate
- size: number (hectares)
- farmType: FarmType (SMALLHOLDER | INDUSTRIAL)
- fields: Field[]
- livestock: LivestockUnit[]
| - waterRights: WaterAllocation
- sustainabilityMetrics: SustainabilityScore
+ createField(size: number, resolution: number)
| + applyPrecisionAgriculture(data: NASAData): void |
+ calculateWaterSavings(): number
+ optimizeResourceAllocation(): AllocationPlan
| + demonstrateROI(): FinancialMetrics
             contains
           Field
- fieldId: UUID
| - geometry: Polygon
- soilType: SoilClassification
- currentCrop: Crop
- moistureProfile: MoistureData[]
| - ndviHistory: NDVITimeSeries
- appliedResolution: number
+ plantCrop(crop: Crop, date: Date): boolean
| + irrigateWithPrecision(zones: Zone[]): void
| + applyVariableRateFertilizer(prescription: Map) |
| + detectStressWithNDVI(threshold: number): Alert |
+ compareResolutionImpact(r1: number, r2: number)
```

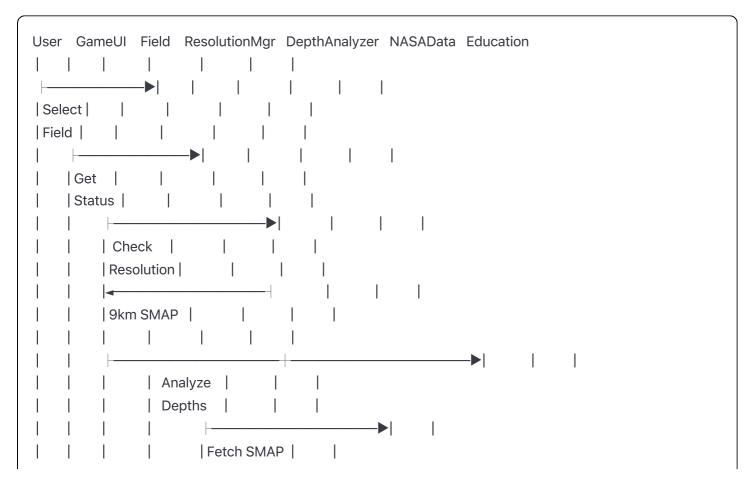
#### 2. Component Diagram - System Architecture

1	NASA Farm Navigators System	1		
	<u> </u>			
İ	·			

Presentation Layer		
Game UI    Dashboard    Tutorial        (Phaser.js)    (React)    System       		
l l		
Game Logic Layer	•	
Farm Manager     Data Processor     Education		
Resolution    Depth    Context        Educator    Analyzer    Adapter		
Data Integration Layer	V	
AppEEARS     Crop-CASMA     Worldview		

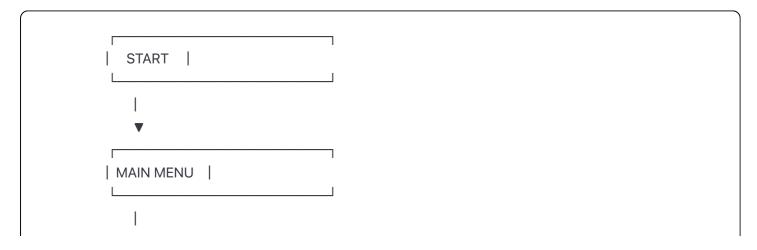
```
| | | Connector | | Connector | | Cache | | |
     Infrastructure Layer | |
| | | Services | | Worker | | Cache | | |
```

#### 3. Sequence Diagram - Precision Irrigation Decision Flow



```
L3 & L4
             | Multi-depth |
             Data
     | Surface:25% |
        RootZone:45%
 | Display |
   Alert
Show
                 |"Low |
                  |Explain |
|moisture|
| at 9km |
                  Resolution
grid" |
             | Tutorial |
|Deep | -
     Apply -
     | Variable | Calculate |
   Rate Root Needs
             Zone Map
      Updated |
  Status |
| "30% |
| water |
saved"
```

### 4. State Diagram - Game Progression States



```
| NEW PLAYER | | RETURNING USER |
| TUTORIAL | |
1. Resolution
2. Depth
3. Context
       \blacksquare
    FARM SETUP
    - Location
    - Scale
    - Context
       PLAYING 
  | | - Monitor | |
  | | - Decide | |
   | | - Act | |
  | | - Learn | |
       \blacksquare
   | DATA ANALYSIS | | |
   | | - Resolution | |
  | | - Depth | |
  | | - Accuracy | |
  | ACTION | |
```

```
| | - Irrigation | |
| | - Fertilizer | |
| | - Harvest | |
     ▼ |
 | FEEDBACK ||
   - 11
| | - Results | |
| | - Education | |
| | - Score | |
SEASON CHANGE |----
 YEAR END
 - Analytics
 - Achievements
 - Insights
```

#### 5. Activity Diagram - Data Resolution Education Flow

```
|Show Pixel Grid |
  on Farm Map
  ◆ Can detect feature? ◆
            No
  Yes
|Show Feature | |Explain Why Not |
| Detected | | "Feature smaller |
               than pixel"
  |Compare Multiple |
  Resolutions
  |Interactive Quiz |
  |"Which resolution |
  for this task?"
  ◆ Correct Answer? ◆
  Yes
            No
|Award XP | | Explain |
|Progress | |Concept |
  | Save Learning
  | Progress |
```



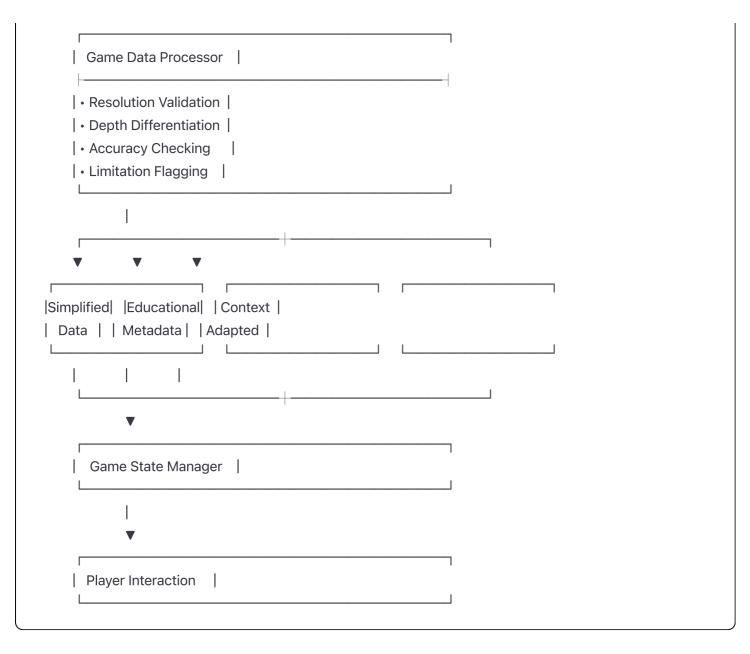
#### 6. Use Case Diagram - System Actors and Functions

```
NASA Farm Navigators System
Manage Farm with NASA Data
| Student | | Farmer | | Educator | |
                  I I
| |Learn Data | |Apply Precision| | | |
| Resolution | Agriculture | |
                  | Understand | Compare Farm | |
| |Soil Depths | |Contexts | | |
                 |Practice | |Track | | |
| | Irrigation | | | Sustainability | | |
    NASA Data Sources
    - AppEEARS
    - Crop-CASMA
```

#### 7. Deployment Diagram - Infrastructure Architecture

```
AWS Application Load |
   Balancer (ALB)
             | AWS ECS | AWS ECS | AWS ECS | | | | | | | |
| Container | Container | Container |
|||Node.js || ||Node.js || ||
AWS Services Layer
| | PostgreSQL | Time Series | | | |
 |rosigresql | |Ime Series | | | |
| | | ElastiCache | Lambda | |
| | | Redis | | Functions | |
      1 1
| | | S3 | | SQS | | |
```

#### 8. Data Flow Diagram - NASA Data Processing Pipeline



## 9. Package Diagram - System Modules

nasa-farm-navigators	
	1
@nfn/core	]
• GameEngine	
• StateManager	
	1
	ı I
@nfn/nasa-integration     	ļ <b>[</b>
• EarthdataAuth	

@nfn/resolution-engine	
ResolutionManager     PixelCalculator     DetectionAnalyzer     ComparisonVisualizer	
@nfn/depth-analysis	
SoilDepthProfiler     RootZoneCalculator     IrrigationOptimizer     DepthEducator	
@nfn/context-adapter	
SmallholderAdapter     IndustrialAdapter     ResourceConstraints     ScaleAdjuster	
I	
@nfn/education	
TutorialSystem     ProgressTracker     AssessmentEngine     CertificationManager	
@nfn/offline	

• ConflictResolver	
	T I