

Documentation Complète - Utilitaire Cartes Hexagonales Wargames

Contexte et Objectif

Projet : Utilitaire pour créer et analyser des grilles hexagonales sur cartes de wargames

Langage : Free Pascal 3.2.2 (Object Pascal)

Librairies : raylib + raygui (guvacode)

Statut : Application fonctionnelle avec détection automatique des terrains

Architecture Technique

Structure des fichiers principaux

```
hexagongridflatop.lpr    // Programme principal + interface
├── initvariable.pas      // Variables globales, types, sauvegarde/chargement
├── hexagonlogic.pas      // Génération grille + calcul voisins
├── detectionlogic.pas    // Analyse couleurs + classification terrain
├── boutonclik.pas       // Utilitaires boutons
└── traceastar.pas       // Algorithme A* (pathfinding)
```

Types de données principaux

pascal

```
THexCell = record
  Number: integer;           // Numéro unique (1..TotalNbreHex)
  center: TVector2;          // Position centre (x, y)
  Vertices: array[0..5] of TPoint; // 6 sommets hexagone
  Color: TColor;              // Couleur affichage (damier)
  ColorPt: TColor;            // Couleur échantillonnée carte
  Selected: boolean;          // État sélection
  Neighbors: array[1..6] of integer; // Numéros 6 voisins
  Colonne, ligne: integer;    // Position grille
  Poshexagone: TEmplacement;  // Type emplacement (coin/bord/centre)
  TypeTerrain: Integer;       // Type détecté (0=aucun, 1,2,3...=types)
  IsReference: Integer;       // Numéro référence (0=normal, 1,2,3...=ordre)
  // + Champs A*, parité, etc.
end;

TColorSignature = record
  DominantColors: array[0..2] of TColor; // 3 couleurs principales
  ColorCounts: array[0..2] of Integer; // Fréquence chaque couleur
  TotalPixels: Integer;                // Total pixels analysés
  IsValid: Boolean;                    // Signature valide
end;
```

Variables globales importantes

pascal

```
HexGrid: array of THexCell; // Grille dynamique hexagones
columns, rows: integer;      // Dimensions grille (16x20 par défaut)
TotalNbreHex: integer;       // Nombre total hexagones
HexOrientation: THexOrientation; // hoFlatTop ou hoPointyTop
AppMode: TAppMode;           // amNormal, amDetection, amSuppression
DetectionActive: Boolean;     // Mode sélection références actif
NombreReferences: Integer;    // Nombre références sélectionnées
ReferenceSignatures: array of TColorSignature; // Signatures références
ShowResetDialog: Boolean;     // Popup confirmation reset
ValeurSpinnerCorrection: Integer; // Type sélectionné pour correction (1,2,3...)
```

Fonctionnalités Principales

1. Gestion des Cartes

- **Import** : PNG, BMP, JPG, JPEG depuis ./ressources/
- **Sauvegarde** : Système complet ./save/NomCarte/ (image + paramètres + CSV)

- **Chargement** : Restauration complète projet (image + grille + détections)
- **Export CSV** : hexgridplat.csv avec toutes les données hexagones

2. Configuration Grille Hexagonale

- **Dimensions** : 2-1000 colonnes/lignes (dynamique)
- **Orientations** : Flat Top (sommet plat) / Pointy Top (sommet pointu)
- **CoinIn** : Décalage colonnes/lignes impaires (true/false)
- **Ajustement précis** : Position pixel par pixel, zoom, glisser-déposer
- **Calcul automatique** : 6 voisins par hexagone selon orientation et position

3. Modes d'Application

pascal

amNormal // Sélection + informations hexagones

amDetection // Analyse automatique + correction manuelle terrains

amSuppression // Non implémenté

4. Détection Automatique des Terrains

Pipeline complet :

1. **Sélection références** : Clic sur hexagones représentatifs de chaque terrain
2. **Analyse couleurs** : Extraction signature couleur (3 dominantes + fréquences)
3. **Classification** : Comparaison toutes signatures avec références
4. **Correction manuelle** : Spinner + clic pour corriger erreurs

Algorithme analyse couleur :

- Zone analyse : Cercle 80% rayon hexagone
- Échantillonnage : Tous pixels dans cercle
- Regroupement : Couleurs similaires (seuil Manhattan ≤ 40)
- Signature : 3 couleurs dominantes triées par fréquence

Interface Utilisateur

Panneau de Contrôle (droite)

[Sauve les coord Hex] // Export CSV
[Charger une carte] // Liste cartes sauvegardées
[Importer carte] // Sélection image ressources
[Sauver carte] // Sauvegarde projet complet
[Générer grille] // Applique nouveaux paramètres

☐ Pointy Top // Orientation hexagones
☐ Numbers // Affichage numéros
☐ Afficher grille // Masquer/afficher hexagones
☐ CoinIn // Décalage impaires

Colonnes: [16] Lignes: [20] // Dimensions grille
Mode: ☐Normal ☒Détection ☐Suppression // Toggle modes

=== EN MODE DÉTECTION ===

Status: [Prêt/En cours/Terminé]

[Commencer sélection] / [Terminer sélection]

SI références > 0:

Modification du terrain: [Spinner 1-NombreReferences]

Type: X // Valeur sélectionnée

Zone d'Information (bas)

- **Hexagone sélectionné** : Numéro, position L/C, centre (x,y)
- **Voisins** : Liste 6 hexagones adjacents [N1] [N2] [N3] [N4] [N5] [N6]
- **Couleurs** : Affichage + couleur carte RGB
- **Détection** (mode détection) : " RÉFÉRENCE #X | Type terrain: Y" ou "Type terrain: Z (classifié)"

Raccourcis Clavier

- **Flèches** : Déplacement grille pixel par pixel
- **+/-** : Zoom hexagones
- **Ctrl+S** : Sauvegarde paramètres ajustement
- **Ctrl+L** : Chargement paramètres ajustement
- **Ctrl+R** : Reset paramètres par défaut

Workflows Types

Nouveau Projet

1. **Import carte** → Sélection image
2. **Configuration grille** → Dimensions, orientation, position

3. **Mode Détection** → Sélection références terrain
4. **Classification auto** → Algorithme analyse tous hexagones
5. **Corrections manuelles** → Spinner + clics pour ajustements
6. **Sauvegarde** → Projet complet

Reprise Projet

1. **Charger carte** → Sélection projet existant
2. **Corrections** → Mode détection + spinner si nécessaire
3. **Sauvegarde** → Mise à jour

Reset Détection

- **Nouvelle détection** avec références existantes → **Popup confirmation**
 - **"Oui"** : Reset complet (références + types + corrections) + nouvelle sélection
 - **"Non"** : Annulation, garde existant
-

Formats de Sauvegarde

Structure ./save/NomCarte/

NomCarte.png/.jpg/.bmp // Image originale carte
ajustements.txt // Paramètres grille (dimensions, orientation, positions, etc.)
hexgridplat.csv // Données complètes hexagones

Format ajustements.txt

```
HexDiameter=70.67
HexScale=0.97
HexOrientation=0 // 0=FlatTop, 1=PointyTop
CoinIn=False
DeltaRelativeX=43.00
DeltaRelativeY=75.00
Columns=16
Rows=20
```

Format hexgridplat.csv (34 colonnes)

csv

Number,CenterX,CenterY,ColorR,ColorG,ColorB,ColorPtR,ColorPtG,ColorPtB,
BSelected,Colonne,Ligne,Emplacement,PairImpairLigne,
Vertex1X,Vertex1Y,Vertex2X,Vertex2Y,Vertex3X,Vertex3Y,
Vertex4X,Vertex4Y,Vertex5X,Vertex5Y,Vertex6X,Vertex6Y,
Neighbor1,Neighbor2,Neighbor3,Neighbor4,Neighbor5,Neighbor6,
TypeTerrain,IsReference

Colonnes clés détection :

- **TypeTerrain** : 0=non déterminé, 1,2,3...=types détectés/corrigés
- **IsReference** : 0=hexagone normal, 1,2,3...=numéro référence

API Fonctions Principales

Génération Grille

pascal

```
procedure GenerateHexagons;           // Génère tous hexagones avec positions
procedure CalculateNeighbors;         // Calcule voisins selon orientation
procedure RecalculerDimensionsHex;    // HexRadius, HexWidth, HexHeight
procedure GenererNouvelleGrille;      // Redimensionne + régénère + voisins
```

Détection Terrain

pascal

```
procedure StartReferenceSelection;     // Active mode sélection
procedure HandleDetectionClick(hexNumber); // Ajoute référence
procedure StopReferenceSelection;      // Lance analyse + classification
procedure ResetDetectionComplete;     // Reset complet système
function AnalyzeHexagonColors(hexNumber): TColorSignature; // Analyse 1 hexagone
procedure ClassifyAllHexagons;        // Classifie toute la grille
```

Sauvegarde/Chargement

pascal

```
procedure SauvegarderCarteUniverselle; // Sauvegarde complète projet
procedure LoadCarteComplete(carteName); // Chargement complet projet
function LoadDetectionDataFromCSV: Boolean; // Restaure TypeTerrain + IsReference
procedure SaveHexGridToCSV;           // Export CSV avec détections
```

Points Techniques Importants

Gestion Mémoire

- **HexGrid dynamique** : SetLength() à chaque changement dimensions
- **ReferenceSignatures** : Taille TotalNbHex + 1 (index 1..n)
- **Pas fuites** : raylib gère automatiquement textures/images

Performance

- **Analyse couleur** peut être lente sur grandes grilles (>50x50)
- **Classification** : Progression visible avec StatusDetection
- **Raylib non thread-safe** : tout sur thread principal

Algorithmes Clés

- **Voisins hexagones** : 2 algorithmes selon orientation (Flat/Pointy Top)
 - **Parité colonnes/lignes** : Comportement différent pair/impair + CoinIn
 - **Distance couleurs** : Manhattan ($|R1-R2| + |G1-G2| + |B1-B2|$) \leq seuil
 - **Classification** : Mode strict (ordre) vs non-strict (présence)
-

Extensions Possibles

Export Formats Jeu

- **Vassal** : XML + notation hexagonale + règles mouvement
- **Table Top Simulator** : JSON positions 3D + snap points
- **Roll20** : Grille overlay + macros

Améliorations Détection

- **Clustering avancé** : K-means au lieu seuil fixe
- **IA classification** : Réseaux neurones pour terrains complexes
- **Multi-résolution** : Grilles hiérarchiques

Interface

- **Preview 3D** : Rendu perspective grille
 - **Éditeur règles** : Coûts mouvement par terrain
 - **Batch processing** : Traitement multiple cartes
-

État Actuel vs TODO

✓ Fonctionnel

- Génération grille hexagonale (orientations, dimensions dynamiques)
- Calcul voisins automatique (gestion bords/coins)
- Import/export cartes avec paramètres
- Détection automatique terrain (analyse couleur + classification)
- Correction manuelle (spinner + clics)
- Sauvegarde/chargement projets complets
- Interface complète avec modes
- Reset détection avec confirmation

▣ Non Implémenté

- Mode suppression
- Export formats jeu (Vassal, TTS, etc.)
- Notation hexagonale standard (A1, B2, etc.)
- Règles mouvement par terrain
- Optimisations performance grandes grilles

Cette documentation couvre l'état complet du projet pour permettre une nouvelle session de développement efficace.