

# Snake 2D Complete Template Documentation

---

## **AudioManager.cs**

Singleton that manages all sound effects and background music.

- `Awake()` → creates `AudioSource` components for every sound in the list
- `Start()` → automatically starts background music if name is set
- `Play(string name)` → finds sound by name and plays it (one-shot or looped)
- `Stop(string name)` → stops the sound with given name

## **DatabaseManager.cs**

Singleton wrapper around `PlayerPrefs` for saving/loading game progress and settings.

- `SaveLevelProgress(int level)` → saves highest unlocked level
- `SaveDifficultyLevel(string difficulty)` → saves chosen difficulty name
- `LoadLevelProgress()` → returns saved level (defaults to 1)
- `LoadDifficultyLevel()` → returns saved difficulty (defaults to "Medium")
- `ResetProgress()` → deletes all `PlayerPrefs`

## **GridManager.cs**

Central grid logic — bounds, wrapping, tile placement, occupation checks for snake/food/obstacles.

- `UpdateSnakeTiles()` → clears old snake tiles → places new ones using cached tiles per body part
- `UpdateFoodTiles()` → refreshes food tile visuals
- `UpdateObstacleTiles()` → refreshes obstacle tile visuals
- `IsCellOccupied(Vector3Int)` → true if any tile exists at position
- `isCellFood(Vector3Int, out Food)` → checks if food is present and returns it
- `WrapPosition(Vector3Int)` → loops coordinates when bounds are disabled
- `CheckBounds(Vector3Int)` → true if position is outside allowed area (when bounds enabled)

## **Loader.cs**

Static scene management helper with enum-based loading.

- `LoadScene(Scenes target)` → loads scene by enum value and updates current scene tracking

- ReloadScene() → reloads currently active scene

## **GameManager.cs**

Overall game state machine (playing / paused / won / lost), score, difficulty application, win/lose events.

- SetState(State) → changes game state and fires OnStateChanged event
- IsGamePlaying() → true when state == Playing
- SaveLevelProgress (via DatabaseManager when winning level)
- Difficulty loaded & applied to SnakeManager move speed in Start()
- Score increased via Food event → checks win condition if score-based

## **FixedAspectRatio.cs**

Camera component that forces 16:9 aspect ratio with letterboxing/pillarboxing.

- Start() → calculates and sets camera viewport rect to maintain 16:9

## **FoodManager.cs**

Spawns and tracks food instances, enforces maximum food count, respawns with delay.

- SpawnFood() → instantiates food at random free cell
- GetFoodList() → returns active food objects
- Automatically removes eaten food via event and spawns new one when under limit

## **ObstacleManager.cs**

Spawns predefined obstacles from list of cells (editable via custom editor).

- CreateObstacles() (in Start) → instantiates Obstacle prefabs at saved cells
- GetObstacleList() → active obstacle instances
- GetObstacleCells() → list of grid positions (used by editor)

## **Obstacle.cs**

Simple data container for one obstacle tile.

- SetCell / GetCell → grid position
- GetSprite() → sprite to display on tilemap

## **SnakeBodyPart.cs**

One segment of the snake (head, body, tail) — handles position, direction, sprite selection.

- Move() → advances one cell in current direction + wrapping
- SetPos(Vector3Int) → teleports to cell (with wrapping)
- SetDirection(Vector3Int) → changes movement direction + decides if curve sprite needed
- GetSprite() → returns current sprite (straight / curve / head/tail variant)

## **GameInputManager.cs**

Unified input reading (keyboard + mobile swipe).

- MoveDirection property → current intended direction (updated every frame)
- Prevents 180° reversal
- Swipe detection with minimum pixel threshold

## **BackgroundScaler.cs**

Scales background sprite to exactly cover the playable grid area + padding.

- Update() → recalculates scale and position based on current grid bounds

## **Food.cs**

Single food item logic — position, score value, eat behavior.

- FoodEaten() → triggers events, plays sound, destroys self
- SetCell / GetCell → grid position & world position update
- GetSprite() → visual for tilemap

## **SnakeManager.cs**

Core snake logic — movement, growth, collision detection, step timing.

- Update() → counts down move timer → moves whole body → checks collisions
- Food\_foodEaten → inserts new body part before tail
- GetSnakeBodyPartList() → all segments (for grid rendering)
- SetMoveSpeed(int) → changes movement frequency (difficulty)

## **UI-related scripts (ButtonUI, ButtonHandlerUI, ScoreManagerUI, GameResultUI, LevelButtonUI, LevelsMenuButtonHandlerUI, MainMenuButtonHandlerUI, SettingsMenuButtonHandlerUI)**

- **ButtonUI** → plays button sound on pointer down
- **ButtonHandlerUI** → connects pause/resume/restart/home buttons to GameManager & scene loading
- **ScoreManagerUI** → updates TMP text when score changes
- **GameResultUI** → shows win/lose overlay when state changes

- **LevelButtonUI** → carries level index, used for unlock logic
- **LevelsMenuButtonHandlerUI** → enables/disables level buttons based on saved progress
- **MainMenuButtonHandlerUI** → handles navigation to modes/settings/quit
- **SettingsMenuButtonHandlerUI** → saves chosen difficulty and returns to main menu