

Delivery game

Τεχνική Τεκμηρίωση Παιχνιδιού
Προσομοίωσης Διανομής

Τίτλος Εργασίας: Ανάπτυξη προηγμένου παιχνιδιού προσομοίωσης διανομής (Advanced Delivery Simulator) σε γλώσσα C με χρήση της βιβλιοθήκης Raylib.

Στοιχεία Συγγραφέων:

Μιχαήλ Μιχαηλίδης - 11929 Λουκάς Λίτσο - 11922

Ακαδημαϊκά Στοιχεία:

Μάθημα: Δομημένος προγραμματισμός - 004 (1ο εξάμηνο)

Ακαδημαϊκό Έτος: 2025 - 2026

Τμήμα Ηλεκτρολόγων μηχανικών και μηχανικών υπολογιστών Πολυτεχνικής σχολής ΑΠΘ

ΠΕΡΙΕΧΟΜΕΝΑ

Άδεια Χρήσης (License)	3
Περιγραφή Λογισμικού & Δυνατότητες	4
Περιβάλλον Ανάπτυξης & Εργαλεία.	5
Οδηγίες Μεταγλώττισης (Compilation)	5
Οδηγίες Εγκατάστασης & Εκτέλεσης	7
Οδηγίες Χειρισμού (Manual)	7
Παράρτημα: Τεκμηρίωση Κώδικα & Συναρτήσεων	8
Αναφορές	27

1. ΑΔΕΙΑ ΧΡΗΣΗΣ (LICENSE)

Το παρόν λογισμικό διατίθεται υπό την άδεια **zlib/libpng License**.

Η επιλογή αυτής της άδειας έγινε για να υπάρχει συμβατότητα με τη βιβλιοθήκη **Raylib** που χρησιμοποιήθηκε για την ανάπτυξη, καθώς και για την ευελιξία που προσφέρει στη χρήση πόρων (εικόνες/ήχος) που απαιτούν αναφορά δημιουργού (attribution).

Όροι της άδειας zlib:

Copyright (c) 2025-2026 Lucas Liço, Michail Michailidis

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

2. ΠΕΡΙΓΡΑΦΗ ΛΟΓΙΣΜΙΚΟΥ

Το λογισμικό αποτελεί μία προηγμένη προσομοίωση επαγγέλματος διανομέα σε ανοιχτό κόσμο (open world). Σε αντίθεση με απλές υλοποιήσεις, η συγκεκριμένη εκδοχή ενσωματώνει ρεαλιστικά στοιχεία διαχείρισης πόρων και οικονομίας.

Βασικοί Μηχανισμοί:

- **Σύστημα Διανομών:** Ο παίκτης δέχεται παραγγελίες μέσω εφαρμογής στο κινητό του (Delivery App), πλοηγείται στο κατάστημα και παραδίδει στον πελάτη.
- **Κίνηση & AI Traffic:** Στους δρόμους κινούνται οχήματα με τεχνητή νοημοσύνη (AI) που ακολουθούν τον Κώδικα Οδικής Κυκλοφορίας. Ο παίκτης πρέπει να αποφεύγει τις συγκρούσεις.
- **Smartphone Interface:** Η αλληλεπίδραση με το παιχνίδι γίνεται μέσω ενός εικονικού smartphone που περιλαμβάνει εφαρμογές (Apps) για Χάρτες, Παραγγελίες, κ.α.

Προηγμένα Χαρακτηριστικά (Advanced Features):

1. **Καύσιμα & Ανεφοδιασμός:** Τα οχήματα καταναλώνουν καύσιμα. Ο παίκτης πρέπει να παρακολουθεί τη στάθμη και να επισκέπτεται πρατήρια για ανεφοδιασμό. Αν ξεμείνει, υπάρχει μηχανισμός έκτακτης ανάγκης (Emergency Rescue).
2. **Μηχανικές Βλάβες & Συντήρηση (Mechanic):** Οι συγκρούσεις προκαλούν ζημιές (Health). Ο παίκτης πρέπει να επισκεφτεί το συνεργείο για επισκευές. Αν η υγεία μηδενιστεί, ο παίκτης μεταφέρεται στο συνεργείο με χρηματική ποινή.
3. **Αγορά Οχημάτων (Dealership):** Υπάρχει δυνατότητα επίσκεψης σε αντιπροσωπεία για την αγορά νέων, καλύτερων οχημάτων.
4. **Οικονομία:** Τα χρήματα από τις παραδόσεις χρησιμοποιούνται για καύσιμα, επισκευές και αναβαθμίσεις.
5. **Tutorial & Progression:** Υπάρχει εισαγωγικό επίπεδο εκμάθησης (Tutorial) και σύστημα αποθήκευσης προόδου (Save System) που φορτώνει αυτόματα κατά την εκκίνηση.

3. ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ

Το έργο αναπτύχθηκε σε περιβάλλον Windows 11, χρησιμοποιώντας τη γλώσσα C και τη βιβλιοθήκη γραφικών Raylib.

- **Λειτουργικό Σύστημα:** Windows 11
- **Γλώσσα Προγραμματισμού:** C (ISO/IEC Standard)
- **IDE / Editor:** Visual Studio Code (VS Code)
- **Version Control:** Git
- **Βιβλιοθήκες:**
 - **Standard C Library:** (stdio.h, stdlib.h, string.h, math.h, time.h)
 - **Raylib:** Έκδοση 4.x ή 5.x (για γραφικά, ήχο, είσοδο χρήστη, διαχείριση παραθύρου).

4. ΟΔΗΓΙΕΣ ΜΕΤΑΓΛΩΤΤΙΣΗΣ (COMPILATION)

Για τη μεταγλώττιση του κώδικα χρησιμοποιείται ο compiler gcc (συνήθως μέσω MinGW-w64). Το πακέτο που παραδίδεται είναι πλήρες και περιέχει όλα τα απαραίτητα εργαλεία και βιβλιοθήκες.

4.1. Περιεχόμενα Πακέτου

Ο φάκελος της εργασίας είναι οργανωμένος ως εξής:

- **src/:** Περιέχει τον πηγαίο κώδικα (.c, .h).
- **resources/:** Περιέχει όλα τα assets (εικόνες, ήχους, μοντέλα).
- **build/:** Περιέχει το σύστημα παραγωγής (Premake) και τις εξωτερικές βιβλιοθήκες.
 - **build/external/:** Περιέχει τον πηγαίο κώδικα της **Raylib**, ώστε να μπορεί να χτιστεί από την αρχή χωρίς εξωτερικές εξαρτήσεις.
- **Makefile:** Αρχείο αυτοματοποίησης της μεταγλώττισης.

4.2. Μεταγλώττιση μέσω Visual Studio Code (Προτεινόμενος Τρόπος)

Το έργο είναι παραμετροποιημένο για VS Code.

1. Ανοίξτε τον φάκελο του έργου ("Open Folder") στο VS Code.
2. Πατήστε **F5**.
3. Το σύστημα θα εκτελέσει αυτόματα το premake5 για τη ρύθμιση του έργου και το mingw32-make για τη μεταγλώττιση της Raylib και του παιχνιδιού.

4.3. Μεταγλώττιση μέσω Γραμμής Εντολών (Command Line)

Αν δεν χρησιμοποιείτε VS Code, μπορείτε να χτίσετε το έργο με το παρεχόμενο Makefile.

1. Ανοίξτε το τερματικό (Terminal/Command Prompt) στον κεντρικό φάκελο.
2. Εκτελέστε την εντολή:
`mingw32-make`

(Σημείωση: Αν βρίσκεστε σε Linux/macOS, χρησιμοποιήστε απλώς `make`).

3. Θα δείτε τα μηνύματα `==== Building raylib ====` και `==== Building DeliveryGame ====`.
4. Το εκτελέσιμο `DeliveryGame.exe` (ή παρόμοιο) θα δημιουργηθεί στον φάκελο `bin/Debug`.

4.4. Χειροκίνητη Μεταγλώττιση (Fallback)

Σε περίπτωση που τα εργαλεία αυτοματισμού αποτύχουν, μπορείτε να μεταγλωττίσετε τα αρχεία χειροκίνητα με τον `gcc`, αρκεί να έχετε ρυθμίσει τα `paths` της Raylib:

```
gcc src/*.c -o DeliveryGame.exe -O2 -Wall -Wno-missing-braces -I src -I  
build/external/raylib/src -L build/external/raylib/src -lraylib -lopengl32 -lgdi32 -lwinmm
```

4.5. Αντιμετώπιση Προβλημάτων (Troubleshooting)

- **Σφάλμα "command not found: mingw32-make":** Το σύστημά σας δεν έχει τον compiler στο PATH. Εάν χρησιμοποιείτε το πακέτο `w64devkit` της Raylib, βεβαιωθείτε ότι εκτελείτε την εντολή μέσα από το περιβάλλον του.
- **Σφάλμα "No such file or directory" (assets):** Το παιχνίδι πρέπει να εκτελείται από τον φάκελο που περιέχει το `resources/`. Αν το τρέξετε από το `bin/Debug/`, βεβαιωθείτε ότι έχετε αντιγράψει τον φάκελο `resources` εκεί ή εκτελέστε το από το `root` του project.
- **Αργή μεταγλώττιση:** Την πρώτη φορά η διαδικασία θα χτίσει τη βιβλιοθήκη Raylib. Αυτό μπορεί να πάρει λίγο χρόνο. Οι επόμενες φορές θα είναι πολύ πιο γρήγορες.

5. ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ & ΕΚΤΕΛΕΣΗΣ

Εγκατάσταση

Το πρόγραμμα είναι "portable" (φορητό) και δεν απαιτεί εγκατάσταση στο σύστημα.

1. Αποσυμπιέστε το αρχείο .zip της εργασίας.
2. Βεβαιωθείτε ότι η δομή φακέλων (src, resources, build) διατηρείται ακέραια.

Εκτέλεση

1. Μεταβείτε στον φάκελο που περιέχει το εκτελέσιμο (συνήθως στο root ή στο bin/Debug μετά τη μεταγλώττιση).
2. Κάντε διπλό κλικ στο DeliveryGame.exe.

6. ΟΔΗΓΙΕΣ ΧΕΙΡΙΣΜΟΥ (USER MANUAL)

Γενικός Χειρισμός

- **Ποντίκι:** Πλοήγηση στα μενού, χρήση του κινητού (SmartPhone), επιλογή επιλογών στο χάρτη.
- **Πληκτρολόγιο:** Έλεγχος οχήματος και συντομεύσεις.

Οδήγηση

- **W / Πάνω Βέλος:** Γκάζι / Κίνηση Μπροστά.
- **S / Κάτω Βέλος:** Φρένο / Όπισθεν.
- **A / Αριστερό Βέλος:** Στροφή Αριστερά.
- **D / Δεξί Βέλος:** Στροφή Δεξιά.
- **Space:** Χειρόφρενο.
- **E (πατημένο):** Αλληλεπίδραση με κτίρια (Βενζινάδικο, Συνεργείο, Παραλαβή/Παράδοση).
- **R:** (Όταν τελειώσει η βενζίνη) Κλήση οδικής βοήθειας.

Smartphone & Εφαρμογές

- **Tab / M:** Άνοιγμα/Κλείσιμο κινητού (Phone).
- **1-6:** Γρήγορη επιλογή εφαρμογών (όταν το κινητό είναι ανοιχτό).
- **Delivery App:** Αποδοχή νέων παραγγελιών.
- **Maps App:** Προβολή χάρτη και GPS. Κάντε **κλικ** στο χάρτη για ορισμό προορισμού.

7. ΠΑΡΑΡΤΗΜΑ: ΤΕΚΜΗΡΙΩΣΗ ΣΥΝΑΡΤΗΣΕΩΝ

Ακολουθεί αναλυτική περιγραφή των συναρτήσεων ανά μονάδα μεταγλώττισης (source file).

7.1. Main Module (main.c)

Κεντρικό αρχείο του προγράμματος. Περιέχει την `main()` και τον κεντρικό βρόχο (game loop), καθώς και τη διαχείριση αρχικοποίησης όλων των υποσυστημάτων.

FixPath

- **Δήλωση:** `void FixPath(void)`
- **Περιγραφή:** Ελέγχει αναδρομικά τους γονικούς φακέλους (έως 5 επίπεδα) για να εντοπίσει τον φάκελο `resources`. Εξασφαλίζει ότι το πρόγραμμα μπορεί να βρει τα αρχεία πόρων ανεξάρτητα από το πού εκτελείται το εκτελέσιμο.
- **Παράμετροι:** Καμία.
- **Επιστρεφόμενη Τιμή:** Καμία (`void`).

main

- **Δήλωση:** `int main(void)`
- **Περιγραφή:** Η κύρια συνάρτηση εισόδου του προγράμματος.
 - Αρχικοποιεί το παράθυρο, τον ήχο και τις ρυθμίσεις οθόνης.
 - Εκτελεί το αρχικό μενού (`RunStartMenu_PreLoad`).
 - Φορτώνει τον χάρτη (βάσει επιλογής ή `config file`) και αρχικοποιεί όλα τα υποσυστήματα (Παίκτης, Κίνηση, Κάμερα, Τηλέφωνο, Tutorial, Dealership).
 - Εκτελεί τον κεντρικό βρόχο (Game Loop) που διαχειρίζεται τα Updates (λογική) και Draws (γραφικά).
 - Διαχειρίζεται καταστάσεις όπως θάνατος παίκτη, έλλειψη καυσίμων και αλληλεπιδράσεις (βενζινάδικα, συνεργεία).
 - Κατά τον τερματισμό, αποθηκεύει την πρόοδο και αποδεσμεύει τη μνήμη.
- **Παράμετροι:** Καμία.
- **Επιστρεφόμενη Τιμή:** `int` (0 για επιτυχή τερματισμό).

7.2. Camera Module (camera.c)

Διαχείριση της κάμερας που ακολουθεί τον παίκτη. Περιλαμβάνει λογική για ομαλή κίνηση (smoothing) και αποφυγή εμποδίων.

InitCamera

- **Δήλωση:** `void InitCamera(void)`

- **Περιγραφή:** Αρχικοποιεί την 3D κάμερα (τύπου Camera3D της Raylib) με προοπτική προβολή (Perspective) και ορίζει την αρχική της θέση και γωνία θέασης (FOV 45.0).
- **Παράμετροι:** Καμία.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

Update_Camera

- **Δήλωση:** void Update_Camera(Vector3 player_position, GameMap *map, float player_angle, float dt)
- **Περιγραφή:** Ενημερώνει τη θέση και τον στόχο της κάμερας για να ακολουθεί τον παίκτη.
 - Εφαρμόζει γραμμική παρεμβολή (Lerp) για ομαλή κίνηση του στόχου.
 - Υπολογίζει την ιδανική θέση πίσω από τον παίκτη βάσει της γωνίας του.
 - Ελέγχει για συγκρούσεις με κτίρια (Collision Check) και πλησιάζει την κάμερα αν υπάρχει εμπόδιο ανάμεσα σε αυτήν και τον παίκτη (zoom in).
 - Εφαρμόζει περιορισμούς ύψους και απόστασης για να μη χάνεται η οπτική επαφή.
- **Παράμετροι:**
 - player_position: Η τρέχουσα θέση του παίκτη.
 - map: Δείκτης στον χάρτη για έλεγχο συγκρούσεων.
 - player_angle: Η γωνία στροφής του οχήματος.
 - dt: Delta time για ανεξαρτησία από τα FPS.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

7.3. Traffic System (traffic.c)

Διαχειρίζεται τη λογική κίνησης, αναπαραγωγής (spawning) και απεικόνισης των οχημάτων AI. Υλοποιεί σύστημα κυκλοφορίας που σέβεται τους δρόμους του χάρτη, τις κατευθύνσεις και τα άλλα οχήματα.

InitTraffic

- **Δήλωση:** void InitTraffic(TrafficManager *traffic)
- **Περιγραφή:** Αρχικοποιεί το σύστημα κυκλοφορίας, ορίζοντας όλα τα οχήματα στον πίνακα traffic->vehicles ως ανενεργά (active = false).
- **Παράμετροι:** traffic: Δείκτης στη δομή διαχείρισης κυκλοφορίας.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

UpdateTraffic

- **Δήλωση:** void UpdateTraffic(TrafficManager *traffic, Vector3 player_position, GameMap *map, float dt)

- **Περιγραφή:** Η κύρια συνάρτηση ενημέρωσης της κυκλοφορίας.
 - **Spawning:** Δημιουργεί νέα οχήματα σε τυχαία σημεία του δρόμου εντός συγκεκριμένης ακτίνας από τον παίκτη (SPAWN_RADIUS).
 - **Despawning:** Αφαιρεί οχήματα που απομακρύνονται πολύ από τον παίκτη (DESPAWN_RADIUS).
 - **AI Logic:** Για κάθε ενεργό όχημα, υπολογίζει την ταχύτητα βάσει του ορίου του δρόμου, ελέγχει για εμπόδια (άλλα αυτοκίνητα ή τον παίκτη) και φρενάρει αν χρειαστεί.
 - **Navigation:** Διαχειρίζεται την κίνηση κατά μήκος των ακμών (edges) του γράφου και επιλέγει τυχαία νέα πορεία στις διασταυρώσεις μέσω της FindNextEdge.
 - **Lane Logic:** Υπολογίζει τη θέση του οχήματος ώστε να κινείται στη δεξιά λωρίδα του δρόμου.
- **Παράμετροι:**
 - traffic: Δείκτης στη δομή κυκλοφορίας.
 - player_position: Θέση παίκτη (για spawn/despawn/collisions).
 - map: Ο χάρτης του παιχνιδιού.
 - dt: Delta Time.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

DrawTraffic

- **Δήλωση:** void DrawTraffic(TrafficManager *traffic)
- **Περιγραφή:** Σχεδιάζει τα 3D μοντέλα των οχημάτων.
 - Χρησιμοποιεί διαδικαστική σχεδίαση (Procedural Drawing) με κύβους (DrawCube) για να δημιουργήσει 3 τύπους οχημάτων: Sedan, Van, Truck.
 - Περιστρέφει τα μοντέλα ανάλογα με την κλίση του δρόμου.
 - Σχεδιάζει φώτα πορείας και φρένων (που ανάβουν όταν το όχημα επιβραδύνει).
- **Παράμετροι:** traffic: Δείκτης στη δομή κυκλοφορίας.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

TrafficCollision

- **Δήλωση:** Vector3 TrafficCollision(TrafficManager *traffic, float playerX, float playerZ, float playerRadius)
- **Περιγραφή:** Ελέγχει αν ο παίκτης συγκρούεται με κάποιο από τα ενεργά οχήματα της κυκλοφορίας.
- **Παράμετροι:**
 - playerX, playerZ: Συντεταγμένες παίκτη.
 - playerRadius: Ακτίνα σύγκρουσης παίκτη.

- **Επιστρεφόμενη Τιμή:** Vector3.
 - Αν υπάρχει σύγκρουση: Επιστρέφει διάνυσμα με κατεύθυνση απώθησης (X, Y) και την ταχύτητα πρόσκρουσης (Z).
 - Αν δεν υπάρχει: Επιστρέφει {0, 0, -1}.

7.4. Car Monitor (car_monitor.c)

Διαχειρίζεται την εφαρμογή "Car Monitor" στο κινητό του παίκτη. Επιτρέπει την προβολή ζωντανών διαγνωστικών στοιχείων και την ενεργοποίηση/απενεργοποίηση των μετρητών στο HUD (Head-Up Display).

DrawCarMonitorApp

- **Δήλωση:** void DrawCarMonitorApp(Player *player, Vector2 localMouse, bool click)
- **Περιγραφή:** Σχεδιάζει το UI της εφαρμογής.
 - **Pin Stats:** Παρέχει κουμπιά (Toggle Buttons) για την εμφάνιση ταχύμετρου, δείκτη καυσίμου, θερμομέτρου και μετρητή G-Force στην κύρια οθόνη.
 - **Live Diagnostics:** Εμφανίζει τεχνικά χαρακτηριστικά του οχήματος σε πραγματικό χρόνο (Τελική ταχύτητα, Επιτάχυνση 0-100, Αυτονομία καυσίμου).
 - **Unlockables:** Διαχειρίζεται το κλείδωμα επιλογών (Thermometer, G-Force) αν ο παίκτης δεν έχει αγοράσει τις αντίστοιχες αναβαθμίσεις.
- **Παράμετροι:**
 - player: Δείκτης στον παίκτη (για ανάγνωση/εγγραφή ρυθμίσεων).
 - localMouse: Θέση ποντικίου στην οθόνη του κινητού.
 - click: Αν πατήθηκε το ποντίκι.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

7.5. Dealership (dealership.c)

Υλοποιεί την αντιπροσωπεία αυτοκινήτων, επιτρέποντας στον παίκτη να δει, να αγοράσει και να οδηγήσει νέα οχήματα. Περιλαμβάνει βάση δεδομένων με στατιστικά για 7 τύπους οχημάτων και τις αναβαθμίσεις τους (Base vs Sport/Luxury).

InitCarDatabase

- **Δήλωση:** void InitCarDatabase(void)
- **Περιγραφή:** Αρχικοποιεί τον πίνακα carDatabase με τα στατιστικά για όλα τα οχήματα (Van, Sedan, SUV, Hatchback, Race Car, Truck, Family Van). Ορίζει ταχύτητα, επιτάχυνση, τιμή, κατανάλωση καυσίμου και φυσικές ιδιότητες (τριβή, βάρος).

- **Παράμετροι:** Καμία.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

EnterDealership

- **Δήλωση:** void EnterDealership(Player *player)
- **Περιγραφή:** Ενεργοποιεί την κατάσταση αντιπροσωπείας. Φορτώνει τα 3D μοντέλα των οχημάτων στη μνήμη για προβολή στον εκθεσιακό χώρο (Showroom) και επαναφέρει την κάμερα και την επιλογή στο πρώτο όχημα.
- **Παράμετροι:** player: Δείκτης στον παίκτη (για έλεγχο ιδιοκτησίας).
- **Επιστρεφόμενη Τιμή:** Καμία (void).

UpdateDealership

- **Δήλωση:** void UpdateDealership(Player *player)
- **Περιγραφή:** Διαχειρίζεται την πλοήγηση στο μενού της αντιπροσωπείας.
 - **Arrow Keys:** Αλλαγή επιλεγμένου οχήματος (Αριστερά/Δεξιά) και έκδοσης (Πάνω/Κάτω για Base/Upgrade).
 - **Enter:** Αγορά οχήματος (αν υπάρχουν χρήματα) ή επιλογή για οδήγηση (αν έχει αγοραστεί). Καλεί την BuyCar ή SelectCar.
 - **Backspace:** Έξοδος από την αντιπροσωπεία.
- **Παράμετροι:** player: Δείκτης στον παίκτη.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

DrawDealership

- **Δήλωση:** void DrawDealership(Player *player)
- **Περιγραφή:** Σχεδιάζει τον 3D χώρο της αντιπροσωπείας και το 2D UI.
 - **3D Showroom:** Σχεδιάζει τοίχους, δάπεδο, έπιπλα, οθόνες και το επιλεγμένο όχημα που περιστρέφεται σε πλατφόρμα.
 - **2D Stats Panel:** Εμφανίζει πίνακα με τα στατιστικά του οχήματος (Ταχύτητα, Επιτάχυνση, Φρένα, Οικονομία, Κόστος) και συγκρίνει την Base έκδοση με την Upgrade.
- **Παράμετροι:** player: Δείκτης στον παίκτη.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

BuyCar

- **Δήλωση:** void BuyCar(Player *player, CarStats stats, Model sourceModel, int index, bool isUpgrade)
- **Περιγραφή:** Εκτελεί τη συναλλαγή αγοράς. Αφαιρεί το ποσό από τον παίκτη, ενημερώνει τη σημαία ιδιοκτησίας (ownedCars ή ownedUpgrades) και καλεί την SelectCar για να τοποθετήσει τον παίκτη στο νέο όχημα.
- **Παράμετροι:**

- player: Ο παίκτης.
- stats: Τα στατιστικά του νέου οχήματος.
- index: Ο δείκτης του οχήματος στη λίστα.
- isUpgrade: Αν πρόκειται για την αναβαθμισμένη έκδοση.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

7.6. Delivery App (delivery_app.c)

Διαχειρίζεται τη λογική των αποστολών και την οικονομία του παιχνιδιού. Υλοποιεί προηγμένους μηχανισμούς αξιολόγησης της οδήγησης (G-Force monitoring) για ευαίσθητα φορτία.

InitDeliveryApp

- **Δήλωση:** void InitDeliveryApp(PhoneState *phone, GameMap *map)
- **Περιγραφή:** Αρχικοποιεί τις υποδοχές εργασιών (Task Slots) ως παραδοθείσες, ώστε να δημιουργηθούν νέες εργασίες αυτόματα στην επόμενη ανανέωση.
- **Παράμετροι:**
 - phone: Δείκτης στο σύστημα κινητού.
 - map: Δείκτης στον χάρτη (για εύρεση καταστημάτων).
- **Επιστρεφόμενη Τιμή:** Καμία (void).

UpdateDeliveryApp

- **Δήλωση:** void UpdateDeliveryApp(PhoneState *phone, Player *player, GameMap *map)
- **Περιγραφή:** Η καρδιά του συστήματος αξιολόγησης.
 - **Job Generation:** Δημιουργεί νέες εργασίες με τυχαία χαρακτηριστικά (Fragile, Heavy, Time Limit) ανάλογα με τον τύπο του καταστήματος (π.χ. Καφές = Χρονικό όριο, Ποτά = Εύθραυστο).
 - **Physics Monitoring:** Υπολογίζει τη δύναμη G (rawG) που ασκείται στο όχημα βάσει της αλλαγής ταχύτητας. Αν υπερβεί το όριο αντοχής του φορτίου, μειώνει την αμοιβή.
 - **Tips:** Υπολογίζει το φιλοδώρημα κατά την παράδοση, επιβραβεύοντας τη γρήγορη και ασφαλή οδήγηση.
- **Παράμετροι:** phone, player, map.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

UpdateDeliveryInteraction

- **Δήλωση:** void UpdateDeliveryInteraction(PhoneState *phone, Player *player, GameMap *map, float dt)
- **Περιγραφή:** Διαχειρίζεται τη φυσική αλληλεπίδραση του παίκτη με το σημείο

παραλαβής/παράδοσης. Ελέγχει αν ο παίκτης κρατάει πατημένο το πλήκτρο E για τον απαιτούμενο χρόνο (4 δευτερόλεπτα) ώστε να ολοκληρωθεί η διαδικασία.

- **Παράμετροι:** dt: Delta Time για το χρονόμετρο.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

DrawDeliveryApp

- **Δήλωση:** void DrawDeliveryApp(PhoneState *phone, Player *player, GameMap *map, Vector2 mouse, bool click)
- **Περιγραφή:** Σχεδιάζει το UI της εφαρμογής Delivery.
 - **Home Screen:** Λίστα διαθέσιμων εργασιών.
 - **Details Screen:** Λεπτομέρειες επιλεγμένης εργασίας, κινδύνους (Spill Risk) και κουμπί αποδοχής.
 - **Profile Screen:** Στατιστικά οδηγού (κέρδη, παραδόσεις).
- **Παράμετροι:** phone, player, map, mouse, click.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

7.7. Map System (map.c)

Το μεγαλύτερο και πιο σύνθετο υποσύστημα του παιχνιδιού. Διαχειρίζεται τη φόρτωση, την απεικόνιση και την πλοήγηση στον κόσμο. Ενσωματώνει τεχνολογίες Streaming και Baking για βελτιστοποίηση. Ο χάρτης βασίζεται σε δεδομένα OpenStreetMap που έχουν υποστεί επεξεργασία.

LoadGameMap

- **Δήλωση:** GameMap LoadGameMap(const char *fileName)
- **Περιγραφή:** Φορτώνει τα δεδομένα του χάρτη από αρχείο κειμένου.
 - Διαβάζει κόμβους (Nodes), δρόμους (Edges), κτίρια (Buildings) και περιοχές (Areas).
 - Καλεί συναρτήσεις αρχικοποίησης για τα Manifests, το γράφο πλοήγησης και το πλέγμα συγκρούσεων.
 - Φορτώνει τα 3D Assets της πόλης (LoadCityAssets).
 - Εξαναγκάζει τη φόρτωση των τομέων γύρω από το σημείο εκκίνησης (0,0) για να μην εμφανιστεί ο παίκτης στο κενό.
- **Παράμετροι:** fileName: Η διαδρομή του αρχείου χάρτη (.map).
- **Επιστρεφόμενη Τιμή:** GameMap: Η δομή που περιέχει όλα τα δεδομένα του κόσμου.

DrawGameMap

- **Δήλωση:** void DrawGameMap(GameMap *map, Camera camera)

- **Περιγραφή:** Η κύρια συνάρτηση απεικόνισης του 3D κόσμου.
 - Ενημερώνει το σύστημα Streaming (UpdateMapStreaming) για να φορτώσει/ξεφορτώσει περιοχές γύρω από την κάμερα.
 - Υπολογίζει ποιοι τομείς (Sectors) είναι ορατοί στην κάμερα (Frustum Culling & Distance Check).
 - Σχεδιάζει τα στατικά μοντέλα των ενεργών τομέων (δρόμοι, κτίρια, δέντρα).
 - Σχεδιάζει ειδικά σημεία ενδιαφέροντος (βενζινάδικα, συνεργεία) και τυχόν ενεργά γεγονότα (ατυχήματα, έργα).
- **Παράμετροι:**
 - map: Δείκτης στα δεδομένα χάρτη.
 - camera: Η κάμερα του παιχνιδιού.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

UpdateMapStreaming

- **Δήλωση:** void UpdateMapStreaming(GameMap *map, Vector3 playerPos)
- **Περιγραφή:** Διαχειρίζεται τη δυναμική φόρτωση του χάρτη (Sector Streaming).
 - Χωρίζει τον κόσμο σε κελιά (Grid Cells) των 100x100 μέτρων.
 - Ελέγχει την απόσταση του παίκτη από κάθε τομέα.
 - Ξεφορτώνει τομείς που είναι πολύ μακριά (UNLOAD_DIST) για εξοικονόμηση μνήμης.
 - Εντοπίζει νέους τομείς που πρέπει να φορτωθούν και εκκινεί τη διαδικασία φόρτωσης σταδιακά (ProcessSectorLoadStep) για να μην παγώνει το παιχνίδι.
- **Παράμετροι:**
 - map: Δείκτης στα δεδομένα χάρτη.
 - playerPos: Η θέση του παίκτη.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

BakeBuildingGeometry

- **Δήλωση:** void BakeBuildingGeometry(Building *b)
- **Περιγραφή:** Παράγει διαδικαστικά (procedurally) το 3D μοντέλο ενός κτιρίου.
 - Χρησιμοποιεί το αποτύπωμα (footprint) από το OSM.
 - Επιλέγει τυχαίο στυλ (ουρανοξύστης, μοντέρνο, κλασικό) και ύψος.
 - Δημιουργεί τοίχους, παράθυρα, πόρτες και οροφές, τοποθετώντας τα στο SectorBuilder για να γίνουν ένα ενιαίο Mesh.
- **Παράμετροι:** b: Δείκτης στο κτίριο που θα επεξεργαστεί.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

FindPath

- **Δήλωση:** `int FindPath(GameMap *map, Vector2 startPos, Vector2 endPos, Vector2 *outPath, int maxPathLen)`
- **Περιγραφή:** Υλοποιεί τον αλγόριθμο A* (A-Star) για την εύρεση της συντομότερης διαδρομής στο οδικό δίκτυο. Χρησιμοποιείται από το GPS του παίκτη.
- **Παράμετροι:**
 - `startPos, endPos`: Σημεία αρχής και τέλους.
 - `outPath`: Πίνακας όπου θα αποθηκευτούν τα σημεία της διαδρομής.
 - `maxPathLen`: Μέγιστο πλήθος σημείων.
- **Επιστρεφόμενη Τιμή:** `int`: Το πλήθος των σημείων της διαδρομής (0 αν δεν βρέθηκε).

CheckMapCollision

- **Δήλωση:** `bool CheckMapCollision(GameMap *map, float x, float z, float radius, bool isCamera)`
- **Περιγραφή:** Ελέγχει για συγκρούσεις με το περιβάλλον χρησιμοποιώντας ένα βελτιστοποιημένο πλέγμα (Spatial Grid). Αντί να ελέγχει όλα τα κτίρια, ελέγχει μόνο αυτά που βρίσκονται στο ίδιο κελί με τον παίκτη ($O(1)$ αντί για $O(N)$).
- **Παράμετροι:**
 - `x, z`: Συντεταγμένες ελέγχου.
 - `radius`: Ακτίνα σύγκρουσης.
 - `isCamera`: Αν είναι `true`, αγνοεί τα γεγονότα (Events) για να μην κολλάει η κάμερα.
- **Επιστρεφόμενη Τιμή:** `bool` (`True` αν υπάρχει σύγκρουση).

7.8. Maps App (maps_app.c)

Η εφαρμογή πλοήγησης στο κινητό του παίκτη. Παρέχει 2D προβολή του χάρτη, σύστημα αναζήτησης τοποθεσιών, φιλτράρισμα σημείων ενδιαφέροντος (POIs) και διαχείριση GPS.

InitMapsApp

- **Δήλωση:** `void InitMapsApp(void)`
- **Περιγραφή:** Αρχικοποιεί την κατάσταση της εφαρμογής χαρτών. Φορτώνει τα εικονίδια των POIs, ορίζει το αρχικό Zoom της κάμερας και μηδενίζει τις μεταβλητές κατάστασης (αναζήτηση, φίλτρα).
- **Παράμετροι:** Καμία.
- **Επιστρεφόμενη Τιμή:** Καμία (`void`).

UpdateMapsApp

- **Δήλωση:** void UpdateMapsApp(GameMap *map, Vector2 currentPlayerPos, float playerAngle, Vector2 localMouse, bool isClicking)
- **Περιγραφή:** Διαχειρίζεται την αλληλεπίδραση χρήστη με το χάρτη.
 - **Navigation:** Επιτρέπει τη μετακίνηση (pan) και το zoom του χάρτη με το ποντίκι.
 - **Tracking:** Υλοποιεί λειτουργία αυτόματης παρακολούθησης του παίκτη με περιστροφή χάρτη (Heading Up) ή σταθερό βορρά (North Up).
 - **Search:** Διαχειρίζεται την πληκτρολόγηση στην μπάρα αναζήτησης και το φιλτράρισμα αποτελεσμάτων.
 - **Click:** Αν ο χρήστης κάνει κλικ στο χάρτη, θέτει νέο προορισμό GPS, "κλειδώνοντας" το σημείο στον πλησιέστερο δρόμο (SnapToRoad).
- **Παράμετροι:**
 - map: Δεδομένα χάρτη.
 - currentPlayerPos, playerAngle: Στοιχεία παίκτη για το rendering.
 - localMouse, isClicking: Είσοδος ποντικιού εντός της οθόνης του κινητού.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

DrawMapsApp

- **Δήλωση:** void DrawMapsApp(GameMap *map)
- **Περιγραφή:** Σχεδιάζει το γραφικό περιβάλλον της εφαρμογής.
 - Καλεί την DrawMap2DView (από το map.c) για την απεικόνιση των δρόμων/κτιρίων.
 - Σχεδιάζει τη διαδρομή GPS (κόκκινη γραμμή) και την απόσταση μέχρι τον προορισμό.
 - Σχεδιάζει τα εικονίδια (Player, POIs, Events) με περιστροφή ώστε να μένουν όρθια ανεξάρτητα από την περιστροφή του χάρτη.
 - Σχεδιάζει το UI (Μπάρα αναζήτησης, Φίλτρα, Κουμπιά ελέγχου, Πυξίδα).
- **Παράμετροι:** map: Δεδομένα χάρτη.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

SetMapDestination

- **Δήλωση:** void SetMapDestination(GameMap *map, Vector2 dest)
- **Περιγραφή:** Ορίζει έναν νέο προορισμό για το GPS. Καλεί την FindPath για να υπολογίσει τη βέλτιστη διαδρομή και ενεργοποιεί την καθοδήγηση.
- **Παράμετροι:**
 - map: Ο χάρτης.
 - dest: Οι συντεταγμένες του προορισμού.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

SearchLocations

- **Δήλωση:** int SearchLocations(GameMap *map, const char* query, MapLocation* results)
- **Περιγραφή:** Αναζητά τοποθεσίες στο χάρτη βάσει ονόματος (case-insensitive substring search).
- **Παράμετροι:**
 - query: Το κείμενο αναζήτησης.
 - results: Πίνακας για την αποθήκευση των αποτελεσμάτων.
- **Επιστρεφόμενη Τιμή:** int: Πλήθος αποτελεσμάτων που βρέθηκαν.

7.9. Mechanic (mechanic.c)

Υλοποιεί το σύστημα συνεργείου αυτοκινήτων (mechanic shop). Παρέχει γραφικό περιβάλλον για την επισκευή του οχήματος και την αγορά μηχανικών αναβαθμίσεων.

DrawMechanicWindow

- **Δήλωση:** bool DrawMechanicWindow(Player *player, PhoneState *phone, bool isActive, int screenW, int screenH)
- **Περιγραφή:** Σχεδιάζει το παράθυρο του συνεργείου (UI).
 - **Επισκευές:** Υπολογίζει το κόστος επισκευής βάσει της ζημιάς (100 - health) και επιτρέπει την πλήρη αποκατάσταση.
 - **Αναβαθμίσεις Απόδοσης:** Προσφέρει βελτιώσεις για τον κινητήρα (Επιτάχυνση) και τα φρένα.
 - **Αναβαθμίσεις Utility:** Επιτρέπει την επέκταση του ρεζερβουάρ καυσίμου και την προσθήκη θερμομόνωσης (Insulation) για τα ευπαθή φορτία.
 - **Αγορά Apps:** Πουλάει αισθητήρες για το κινητό (Θερμόμετρο, G-Force Meter).
- **Παράμετροι:**
 - player: Δείκτης στον παίκτη (για stats και χρήματα).
 - isActive: Αν το παράθυρο είναι ανοιχτό.
 - screenW, screenH: Διαστάσεις οθόνης για κλιμάκωση UI.
- **Επιστρεφόμενη Τιμή:** bool: True αν το παράθυρο πρέπει να παραμείνει ανοιχτό, False αν ο χρήστης πάτησε "Leave".

MechButton

- **Δήλωση:** static bool MechButton(Rectangle rect, const char* text, Color color, Vector2 mouse, bool disabled)
- **Περιγραφή:** Helper συνάρτηση για τη δημιουργία κουμπιών στο UI του συνεργείου. Διαχειρίζεται την κατάσταση hover, το κλικ και την

απενεργοποιημένη κατάσταση (γκρι χρώμα) αν ο παίκτης δεν έχει αρκετά χρήματα.

- **Παράμετροι:** rect, text, color, mouse, disabled.
- **Επιστρεφόμενη Τιμή:** bool (True αν πατήθηκε).

7.10. Phone System (phone.c)

Διαχειρίζεται το εικονικό smartphone του παίκτη, το οποίο λειτουργεί ως το κεντρικό UI για την πλοήγηση, την αποδοχή εργασιών, τις ρυθμίσεις και τη μουσική. Υλοποιεί ένα σύστημα "Apps" όπου κάθε εφαρμογή έχει τη δική της λογική σχεδίασης και ενημέρωσης.

InitPhone

- **Δήλωση:** void InitPhone(PhoneState *phone, GameMap *map)
- **Περιγραφή:** Αρχικοποιεί την κατάσταση του κινητού.
 - Φορτώνει τα εικονίδια των εφαρμογών (Jobs, Maps, Bank, Music, Settings, CarMon).
 - Αρχικοποιεί τις επιμέρους εφαρμογές (Maps, Delivery).
 - Φορτώνει τη βιβλιοθήκη μουσικής από το φάκελο πόρων.
 - Δημιουργεί το RenderTexture που χρησιμοποιείται ως οθόνη.
- **Παράμετροι:**
 - phone: Δείκτης στη δομή κατάστασης κινητού.
 - map: Δείκτης στον χάρτη (για αρχικοποίηση άλλων apps).
- **Επιστρεφόμενη Τιμή:** Καμία (void).

UpdatePhone

- **Δήλωση:** void UpdatePhone(PhoneState *phone, Player *player, GameMap *map)
- **Περιγραφή:** Διαχειρίζεται τη λογική του κινητού σε κάθε frame.
 - Ελέγχει το πλήκτρο TAB για άνοιγμα/κλείσιμο.
 - Ενημερώνει το animation ολίσθησης (slide animation).
 - Διαχειρίζεται τα shortcuts πληκτρολογίου (1-6) για γρήγορη αλλαγή εφαρμογών.
 - Υπολογίζει τη θέση του ποντικιού μέσα στην οθόνη του κινητού (Local Mouse Position) ανεξάρτητα από την κλιμάκωση της οθόνης.
 - Καλεί την Update της ενεργής εφαρμογής.
- **Παράμετροι:**
 - phone: Κατάσταση κινητού.
 - player: Δεδομένα παίκτη.
 - map: Δεδομένα χάρτη.

- **Επιστρεφόμενη Τιμή:** Καμία (void).

DrawPhone

- **Δήλωση:** void DrawPhone(PhoneState *phone, Player *player, GameMap *map, Vector2 localMouse, bool click)
- **Περιγραφή:** Σχεδιάζει το κινητό και το περιεχόμενό του.
 - Χρησιμοποιεί BeginTextureMode για να σχεδιάσει την τρέχουσα εφαρμογή (Home, Maps, κτλ.) στην εικονική οθόνη.
 - Σχεδιάζει την μπάρα κατάστασης (Status Bar) με ώρα και μπαταρία.
 - Σχεδιάζει τις ειδοποιήσεις (Notifications) πάνω από τις εφαρμογές.
 - Τέλος, σχεδιάζει το πλαίσιο του κινητού (Bezel) και την οθόνη στην κάτω δεξιά γωνία της κύριας οθόνης.
- **Παράμετροι:** localMouse: Συντεταγμένες ποντικιού σχετικές με την οθόνη του κινητού.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

ShowPhoneNotification

- **Δήλωση:** void ShowPhoneNotification(const char *text, Color color)
- **Περιγραφή:** Εμφανίζει μια προσωρινή ειδοποίηση (popup) στο πάνω μέρος της οθόνης του κινητού. Χρησιμοποιείται για μηνύματα όπως "Game Saved", "New Job", κτλ.
- **Παράμετροι:**
 - text: Το κείμενο της ειδοποίησης.
 - color: Το χρώμα του πλαισίου.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

DrawAppMusic

- **Δήλωση:** void DrawAppMusic(PhoneState *phone, Vector2 mouse, bool click)
- **Περιγραφή:** Υλοποιεί το UI της εφαρμογής μουσικής.
 - Σαρώνει το φάκελο resources/Music για αρχεία MP3/WAV.
 - Παρέχει κουμπιά Play, Pause, Next, Previous.
 - Εμφανίζει μπάρα προόδου και πληροφορίες καλλιτέχνη/τραγουδιού.
- **Παράμετροι:** phone, mouse, click: Κατάσταση και είσοδος.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

7.11. Player (player.c)

Δομή και συμπεριφορά του παίκτη. Διαχειρίζεται τη φυσική του οχήματος, την είσοδο χρήστη και τα στατιστικά (υγεία, καύσιμα, χρήματα).

InitPlayer

- **Δήλωση:** Player InitPlayer(Vector3 startPos)
- **Περιγραφή:** Δημιουργεί και αρχικοποιεί τη δομή Player. Ορίζει τις φυσικές παραμέτρους του οχήματος (ταχύτητα, επιτάχυνση, φρένα, τριβή), την αρχική ποσότητα καυσίμου, τα χρήματα και φορτώνει το προεπιλεγμένο μοντέλο οχήματος.
- **Παράμετροι:** startPos: Η θέση εκκίνησης του παίκτη στον κόσμο.
- **Επιστρεφόμενη Τιμή:** Player struct (αρχικοποιημένη).

LoadPlayerContent

- **Δήλωση:** void LoadPlayerContent(Player *player)
- **Περιγραφή:** Φορτώνει το 3D μοντέλο (.obj) του οχήματος από τον φάκελο resources.
- **Παράμετροι:** player: Δείκτης στη δομή του παίκτη.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

UpdatePlayer

- **Δήλωση:** void UpdatePlayer(Player *player, GameMap *map, TrafficManager *traffic, float dt)
- **Περιγραφή:** Η κύρια συνάρτηση ενημέρωσης του παίκτη σε κάθε frame.
 - Διαβάζει την είσοδο (WASD) για κίνηση και στροφή.
 - Υπολογίζει τη φυσική κίνησης (επιτάχυνση, φρενάρισμα, τριβή/coasting).
 - Υπολογίζει τη νέα θέση και καλεί την ResolveMovement για έλεγχο συγκρούσεων.
 - Μειώνει τα καύσιμα βάσει της απόστασης που διανύθηκε.
- **Παράμετροι:**
 - player: Δείκτης στη δομή παίκτη.
 - map, traffic: Δείκτες για ελέγχους περιβάλλοντος.
 - dt: Χρονικό βήμα (Delta Time).
- **Επιστρεφόμενη Τιμή:** Καμία (void).

ResolveMovement

- **Δήλωση:** void ResolveMovement(Player* player, GameMap* map, TrafficManager* traffic, float moveAmount, int axis)
- **Περιγραφή:** Ελέγχει αν η σχεδιαζόμενη κίνηση θα προκαλέσει σύγκρουση.
 - Αν χτυπήσει κτίριο (Map Collision), ακινητοποιεί το όχημα και προκαλεί ζημιά αν η ταχύτητα είναι υψηλή.
 - Αν χτυπήσει άλλο όχημα (Traffic Collision), προκαλεί "αναπήδηση" και ζημιά στην υγεία του παίκτη.

- **Παράμετροι:**
 - moveAmount: Η απόσταση που πρέπει να μετακινηθεί.
 - axis: Ο άξονας κίνησης (1 για X, 0 για Z).
- **Επιστρεφόμενη Τιμή:** Καμία (void).

DrawHealthBar

- **Δήλωση:** void DrawHealthBar(Player *player)
- **Περιγραφή:** Σχεδιάζει τη μπάρα υγείας (Health Bar) στο UI, αλλάζοντας χρώμα και μέγεθος ανάλογα με την κατάσταση του παίκτη.
- **Παράμετροι:** player: Δείκτης για ανάγνωση της τιμής health.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

AddMoney

- **Δήλωση:** void AddMoney(Player *player, const char* desc, float amount)
- **Περιγραφή:** Προσθέτει (ή αφαιρεί) χρήματα από το πορτοφόλι του παίκτη και καταγράφει τη συναλλαγή στο ιστορικό.
- **Παράμετροι:**
 - desc: Περιγραφή συναλλαγής (π.χ. "Delivery Pay").
 - amount: Ποσό (θετικό ή αρνητικό).
- **Επιστρεφόμενη Τιμή:** Καμία (void).

7.12. Save System (save.c)

Διαχειρίζεται την αποθήκευση και φόρτωση της προόδου του παιχνιδιού σε δυαδικό αρχείο. Εφαρμόζει βασική κρυπτογράφηση (XOR obfuscation) για προστασία των δεδομένων.

SaveGame

- **Δήλωση:** bool SaveGame(Player *player, PhoneState *phone)
- **Περιγραφή:** Αποθηκεύει την τρέχουσα κατάσταση στο δίσκο.
 - Συσκευάζει όλα τα δεδομένα (θέση, χρήματα, στατιστικά οχήματος, ιστορικό, ρυθμίσεις) σε μία δομή GameSaveData.
 - Εφαρμόζει κρυπτογράφηση XOR.
 - Γράφει τα δεδομένα στο αρχείο αποθήκευσης.
- **Παράμετροι:** player, phone.
- **Επιστρεφόμενη Τιμή:** bool (True αν πέτυχε).

LoadGame

- **Δήλωση:** bool LoadGame(Player *player, PhoneState *phone)
- **Περιγραφή:** Φορτώνει την πρόοδο από το δίσκο.

- Διαβάζει και αποκρυπτογραφεί το αρχείο.
- Ελέγχει την έκδοση του save (SAVE_VERSION) για συμβατότητα.
- Επαναφέρει τα δεδομένα του παίκτη και **φορτώνει το σωστό 3D μοντέλο οχήματος** που είχε αγοράσει ο χρήστης.
- **Παράμετροι:** player, phone.
- **Επιστρεφόμενη Τιμή:** bool (True αν βρέθηκε και φορτώθηκε σωστά).

ResetSaveGame

- **Δήλωση:** void ResetSaveGame(Player *player, PhoneState *phone)
- **Περιγραφή:** Διαγράφει το αρχείο αποθήκευσης και επαναφέρει όλες τις μεταβλητές του παιχνιδιού στις αρχικές τους τιμές (New Game state). Χρησιμοποιείται από το μενού ρυθμίσεων.
- **Παράμετροι:** player, phone.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

7.13. Screen Visuals (screen_visuals.c)

Διαχειρίζεται την απεικόνιση όλων των στοιχείων HUD (Head-Up Display) και των οπτικών εφέ που δεν ανήκουν στο κινητό τηλέφωνο. Περιλαμβάνει τα όργανα του οχήματος, τις ειδοποιήσεις φορτίου και το UI αλληλεπίδρασης με πρατήρια.

UpdateVisuals

- **Δήλωση:** void UpdateVisuals(float dt)
- **Περιγραφή:** Ενημερώνει χρονόμετρα για οπτικά εφέ και διαχειρίζεται τη δυναμική αυξομείωση της τιμής του καυσίμου (Market Flux) ανά τακτά χρονικά διαστήματα.
- **Παράμετροι:** dt: Delta Time.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

DrawVisualsWithPinned

- **Δήλωση:** void DrawVisualsWithPinned(Player *player, PhoneState *phone)
- **Περιγραφή:** Σχεδιάζει τα όργανα που έχει επιλέξει ο παίκτης να "καρφιτσώσει" στην οθόνη μέσω του Car Monitor App.
 - **Speedometer / Fuel Gauge:** Σχεδιάζει τα αναλογικά όργανα.
 - **G-Force Meter:** Αν είναι ενεργό, εμφανίζει τον μετρητή δυνάμεων.
 - **Thermometer:** Αν είναι ενεργό, εμφανίζει τη θερμοκρασία του φορτίου.
 - **WASD Overlay:** Εμφανίζει τα πλήκτρα κίνησης (χρήσιμο για αρχάριους).
- **Παράμετροι:** player, phone.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

DrawSpeedometer

- **Δήλωση:** void DrawSpeedometer(float currentSpeed, float maxSpeed, int screenWidth)
- **Περιγραφή:** Σχεδιάζει το κυκλικό ταχύμετρο με βελόνα και ψηφιακή ένδειξη χιλιομέτρων.
- **Παράμετροι:** currentSpeed, maxSpeed, screenWidth.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

DrawCargoHUD

- **Δήλωση:** void DrawCargoHUD(PhoneState *phone, Player *player)
- **Περιγραφή:** Εμφανίζει συνοπτικά την κατάσταση της τρέχουσας αποστολής στο πάνω δεξιά μέρος της οθόνης. Περιλαμβάνει:
 - Χρονόμετρο (με αντίστροφη μέτρηση αν υπάρχει όριο).
 - Μπάρα ακεραιότητας (Cargo Integrity) για εύθραυστα φορτία.
 - Μπάρα θερμοκρασίας για ευπαθή προϊόντα.
- **Παράμετροι:** phone, player.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

DrawRefuelWindow

- **Δήλωση:** bool DrawRefuelWindow(Player *player, bool isActive, int screenW, int screenH)
- **Περιγραφή:** Σχεδιάζει το παράθυρο αλληλεπίδρασης με το βενζινάδικο. Επιτρέπει στον παίκτη να επιλέξει ποσότητα καυσίμου (slider) και να πληρώσει.
- **Παράμετροι:** player, isActive, screenW, screenH.
- **Επιστρεφόμενη Τιμή:** bool (True αν είναι ενεργό).

7.14. Start Menu (start_menu.c)

Διαχειρίζεται το αρχικό μενού του παιχνιδιού και τη διαδικασία φόρτωσης (Loading Screen).

RunStartMenu_PreLoad

- **Δήλωση:** bool RunStartMenu_PreLoad(int screenWidth, int screenHeight)
- **Περιγραφή:** Η κύρια λειτουργία του μενού.
 - **3D Background:** Δημιουργεί διαδικαστικά μια πόλη στο παρασκήνιο (Diorama) και περιστρέφει την κάμερα γύρω της.
 - **Main Menu:** Επιτρέπει στον χρήστη να επιλέξει "New Game" ή "Continue".
 - **Map Selection:** Αν επιλεγεί νέο παιχνίδι, ο χρήστης διαλέγει μέγεθος χάρτη (Small/Big).
 - **Pre-Loading:** Όταν ο χρήστης ξεκινήσει, εκτελεί μια "ψεύτικη" μπάρα

φόρτωσης ενώ παράλληλα φορτώνει τα assets του μενού και προετοιμάζει το παιχνίδι.

- **Παράμετροι:** screenWidth, screenHeight.
- **Επιστρεφόμενη Τιμή:** bool: Πάντα false (τερματίζει όταν ολοκληρωθεί η φόρτωση και το main αναλαμβάνει).

DrawMenuDiorama

- **Δήλωση:** void DrawMenuDiorama(void)
- **Περιγραφή:** Σχεδιάζει το 3D σκηνικό του μενού χρησιμοποιώντας βελτιστοποιημένη σχεδίαση (DrawCubeBatched) για να αποφύγει καθυστερήσεις. Περιλαμβάνει κτίρια, φώτα και δρόμους.
- **Παράμετροι:** Καμία.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

DrawLoadingInterface

- **Δήλωση:** void DrawLoadingInterface(int screenWidth, int screenHeight, float progress, const char* status)
- **Περιγραφή:** Σχεδιάζει την μπάρα φόρτωσης και τα μηνύματα κατάστασης.
- **Παράμετροι:** progress: Ποσοστό (0.0 - 1.0).
- **Επιστρεφόμενη Τιμή:** Καμία (void).

DeleteSaveData

- **Δήλωση:** void DeleteSaveData(void)
- **Περιγραφή:** Διαγράφει το αρχείο αποθήκευσης (save_data.dat) και το αρχείο ρυθμίσεων χάρτη, επιτρέποντας την εκκίνηση νέου παιχνιδιού από το μηδέν.
- **Παράμετροι:** Καμία.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

7.15. Tutorial (tutorial.c)

Διαχειρίζεται τη διαδικασία εκμάθησης (onboarding) για τους νέους παίκτες. Λειτουργεί ως ένα Finite State Machine που καθοδηγεί τον παίκτη βήμα-βήμα.

InitTutorial

- **Δήλωση:** void InitTutorial(void)
- **Περιγραφή:** Αρχικοποιεί τις μεταβλητές του σεναρίου εκμάθησης και μηδενίζει τα χρονόμετρα.
- **Παράμετροι:** Καμία.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

UpdateTutorial

- **Δήλωση:** bool UpdateTutorial(Player *player, PhoneState *phone, GameMap *map, float dt, bool isRefueling, bool isMechanicOpen)
- **Περιγραφή:** Η κύρια λογική του Tutorial. Ελέγχει την κατάσταση του παίκτη (π.χ. αν οδηγεί, αν άνοιξε το κινητό, αν παρέδωσε την παραγγελία) και προχωράει το σενάριο στο επόμενο βήμα.
 - Μπορεί να "εξαναγκάσει" τη δημιουργία συγκεκριμένων εργασιών (ForceSpawnJob) για εκπαιδευτικούς σκοπούς (π.χ. Fragile Cargo).
 - Επιστρέφει true αν πρέπει να μπλοκάρει την κίνηση του παίκτη (π.χ. όταν εμφανίζεται μήνυμα διαλόγου).
- **Παράμετροι:** player, phone, map, dt, isRefueling, isMechanicOpen.
- **Επιστρεφόμενη Τιμή:** bool (True αν πρέπει να παγώσει το παιχνίδι).

DrawTutorial

- **Δήλωση:** void DrawTutorial(Player *player, PhoneState *phone)
- **Περιγραφή:** Σχεδιάζει τα στοιχεία UI του tutorial.
 - **Overlay Windows:** Παράθυρα με οδηγίες και κουμπιά "Next/Skip".
 - **Floating Text:** Κείμενο στην οθόνη που υποδεικνύει τι πρέπει να κάνει ο παίκτης (π.χ. "PRESS [TAB] TO OPEN PHONE").
 - **App Guide:** Ειδική οθόνη βοήθειας που εξηγεί τις εφαρμογές του κινητού.
- **Παράμετροι:** player, phone.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

SkipTutorial

- **Δήλωση:** void SkipTutorial(Player *player, PhoneState *phone)
- **Περιγραφή:** Σηματοδοτεί το τέλος της εκμάθησης, ενεργοποιεί την κανονική ροή του παιχνιδιού και αποθηκεύει την πρόοδο.
- **Παράμετροι:** player, phone.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

ShowTutorialHelp

- **Δήλωση:** void ShowTutorialHelp(void)
- **Περιγραφή:** Ενεργοποιεί χειροκίνητα τον οδηγό βοήθειας (Help Screen) μέσα από τις ρυθμίσεις του κινητού, ακόμα και αν ο παίκτης έχει ολοκληρώσει το tutorial.
- **Παράμετροι:** Καμία.
- **Επιστρεφόμενη Τιμή:** Καμία (void).

8. ΑΝΑΦΟΡΕΣ

8.1. Βιβλιοθήκες & Εργαλεία (Libraries & Tools)

Raylib Library: <https://www.raylib.com/> - Cheatsheet & Examples.

C Standard Library Documentation: ISO/IEC 9899.

Choose a License: <https://choosealicense.com/licenses/zlib/>

8.2. Αλγόριθμοι & Μέθοδοι (Algorithms & Methods)

A* Pathfinding Algorithm:

Introduction to the A* Algorithm - Red Blob Games

How does A* pathfinding work? - Game Development Stack Exchange

Spatial Hashing & Grid Optimization:

Optimizing 2D Physics Spatial Hashing - Cristiano Politowski

Optimized Spatial Hashing for Collision Detection - GitHub Pages

Polygon Triangulation (Ear Clipping):

Polygon triangulation - Wikipedia

Ear clipping triangulation - Nils Olovsson

mapbox/earcut.hpp - GitHub (Reference Implementation)

8.3. Βελτιστοποίηση & Γραφικά (Optimization & Graphics)

Procedural Mesh Generation:

Procedural Mesh Generation - Deep Dive - YouTube (General Concepts)

Mesh generation of quad with UV repeats in raylib - GitHub Gist

Raylib Batching:

Raylib Examples - Official Examples

8.4. Βοήθεια & Συνεισφορά AI (AI Assistance)

Google Gemini: Χρησιμοποιήθηκε για τη βελτιστοποίηση αλγορίθμων και τη σύνταξη της τεχνικής τεκμηρίωσης.

8.5. Assets:

8.5.1 Μουσική:

The following audio tracks were obtained from Pixabay. They are used under the Pixabay Content License, which allows for free commercial and non-commercial use.

License: Pixabay Content License (Royalty Free)

Source: <https://pixabay.com/service/license-summary/>

Tracks Used:

1. Upbeat Game / Minecraft Background Music (Track ID: 379533)

Source:

<https://pixabay.com/music/upbeat-game-gaming-minecraft-background-music-379533/>

2. Happy Children's Tunes (Track ID: 295075)

Source:

<https://pixabay.com/music/happy-childrens-tunes-kids-game-gaming-background-music-295075/>

3. Beats No Copyright Gaming Music (Track ID: 404995)

Source:

<https://pixabay.com/music/beats-no-copyright-gaming-background-music-for-minecraft-gaming-404995/>

4. Video Games / Minecraft Background Music (Track ID: 372242)

Source:

<https://pixabay.com/music/video-games-gaming-game-minecraft-background-music-372242/>

8.5.2 3D Μοντέλα:

The 3D models are obtained from Kenney.nl and are licensed under CC0 1.0 Universal (Public Domain).

Source: <https://kenney.nl/assets>

License: CC0 1.0 Universal

Used Packs: Coaster Kit, City Kit Roads, Modular Buildings, Space Station Kit, Toy Car Kit

8.5.3 Εικονίδια:

The following icons were obtained from Flaticon.com. They are used under the Flaticon Free License,

which requires attribution to the author.

--- Phone Interface Icons ---

1. Delivery Truck Icon created by Freepik - <https://www.flaticon.com/free-icons/delivery>
2. Cogwheel (Settings) Icon created by Freepik - <https://www.flaticon.com/free-icons/settings>
3. Treasure Map Icon created by Freepik - <https://www.flaticon.com/free-icons/map>
4. Music Icon created by Freepik - <https://www.flaticon.com/free-icons/music>
5. Customization (Car Tune) Icon created by Freepik - <https://www.flaticon.com/free-icons/car-tune>
6. ATM Card Icon created by Freepik - <https://www.flaticon.com/free-icons/bank>

--- Map App Icons ---

7. Fuel Icon created by Freepik - <https://www.flaticon.com/free-icons/fuel>
8. Burger Icon created by Freepik - <https://www.flaticon.com/free-icons/fast-food>
9. Cup of Tea Icon created by Freepik - <https://www.flaticon.com/free-icons/cafe>
10. Cocktails Icon created by Freepik - <https://www.flaticon.com/free-icons/cocktail>
11. Grocery Cart Icon created by Freepik - <https://www.flaticon.com/free-icons/supermarket>
12. Shopping Bag Icon created by Freepik - <https://www.flaticon.com/free-icons/supermarket>
13. Cutlery Icon created by Freepik - <https://www.flaticon.com/free-icons/restaurant>
14. Wrench Icon created by Freepik - <https://www.flaticon.com/free-icons/mechanic>
15. Emergency Sign Icon created by Freepik - <https://www.flaticon.com/free-icons/emergency-sign>