

2D car game- Drunk Drivers 2

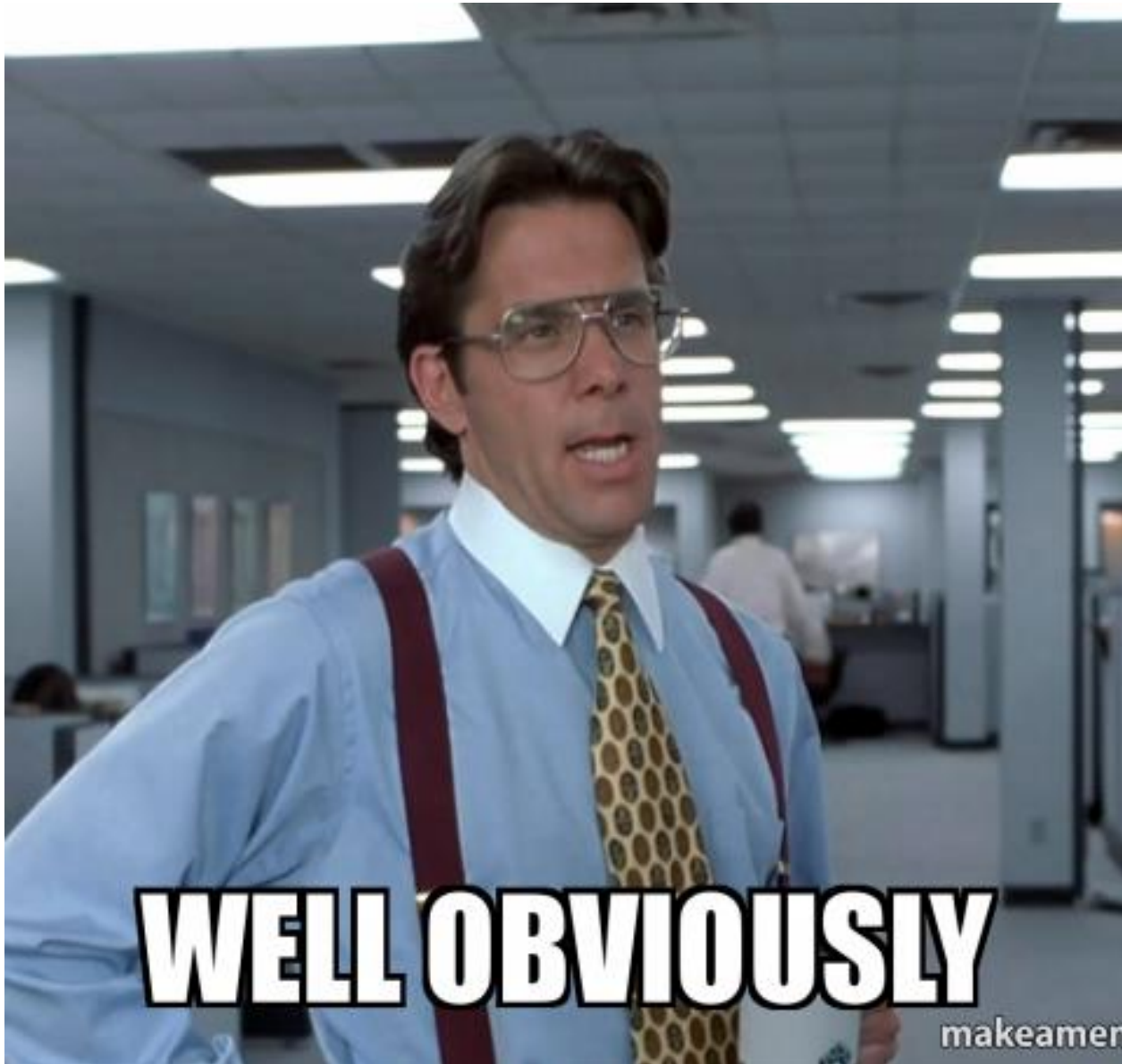
- *Μουστάκας Χρήστος*
- *Τσιάρας Γιώργος*
- *Αλεξανδρίδης Νίκος*

Εισαγωγή & Motivation

-Τι αφορά το project;

- Το project μας αφορά ένα videogame, κατά το οποίο ο παίκτης (Drunk Drivers) έχει μπει στο αντίθετο ρεύμα του δρόμου και πρέπει να αποφύγει τα οχήματα (obstacles) που έρχονται κατά πάνω του, χρησιμοποιώντας τα κατάλληλα πλήκτρα (βελάκια) για τον χειρισμό του οχήματος





Ποιο
προβλημα
λυνει;

Το πρόβλημα το οποίο λύνει
είναι η εξάλειψη της βαρεμάρας

Γιατι το επιλεξατε;

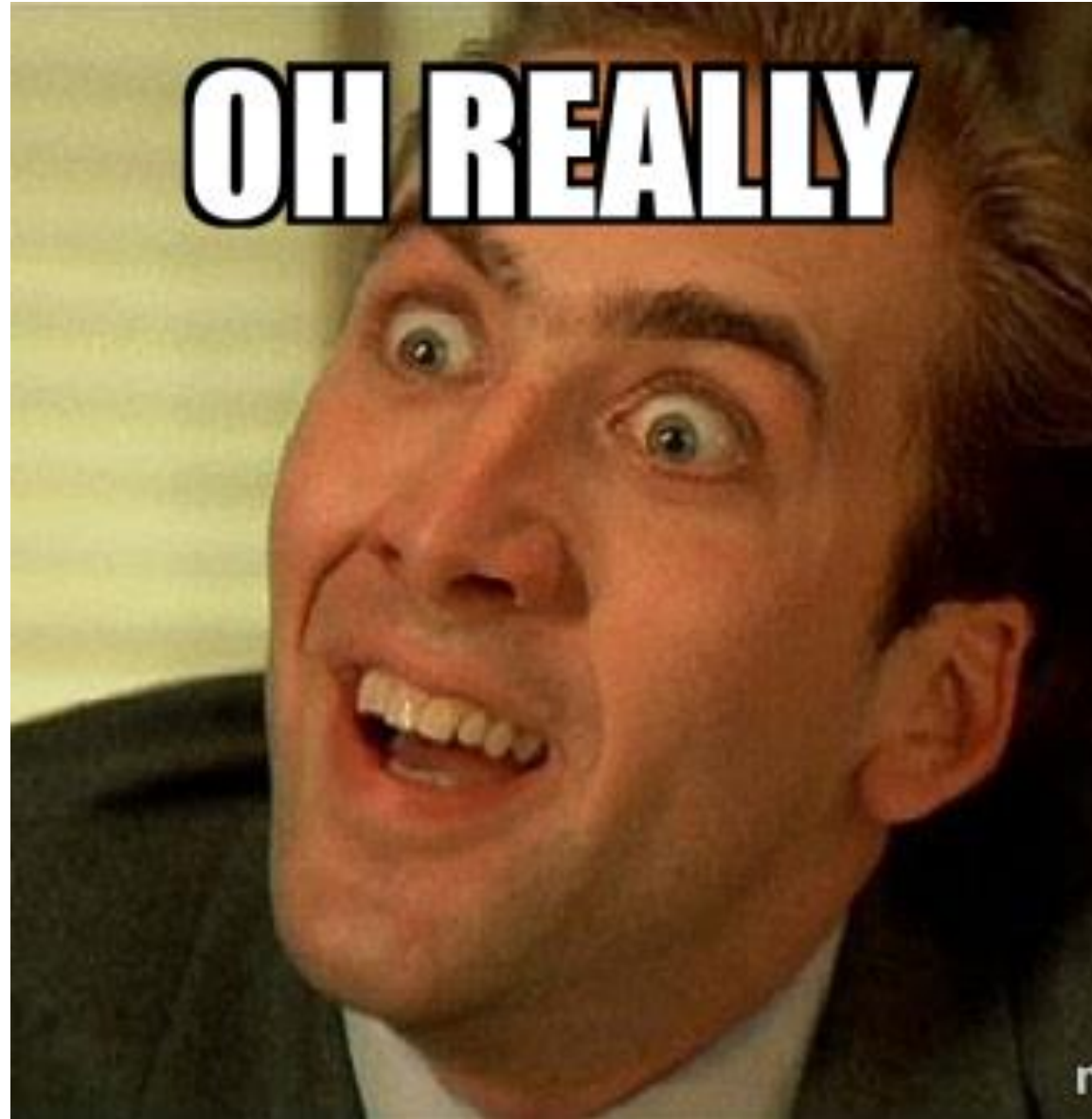
- Η δημιουργία ενός videogame είναι αρκετά ενδιαφέρουσα.
- Ήταν μια καλή πρόκληση.



Objective & Scope

Κυριοι στοχοι του project;

Βασικός στόχος είναι η
ανάπτυξη ενός απλού
videogame.





Δυνατότητες & λειτουργίες που υποστηρίζει

- Το game δίνει τη δυνατότητα επιλογής (1 ή 2 παίκτες), καθώς και τη δυνατότητα επιλογής προτίμησης ανάλυσης (800 x 600 – 1280 x 720 – 1500 x 1000 – 1920 x 1080)

Περιορισμοί (αν υπάρχουν)

*Δυστυχώς δεν μπορείς να
διαλέξεις τι χρώμα θα
είναι το αμάξι σου.*



System Architecture

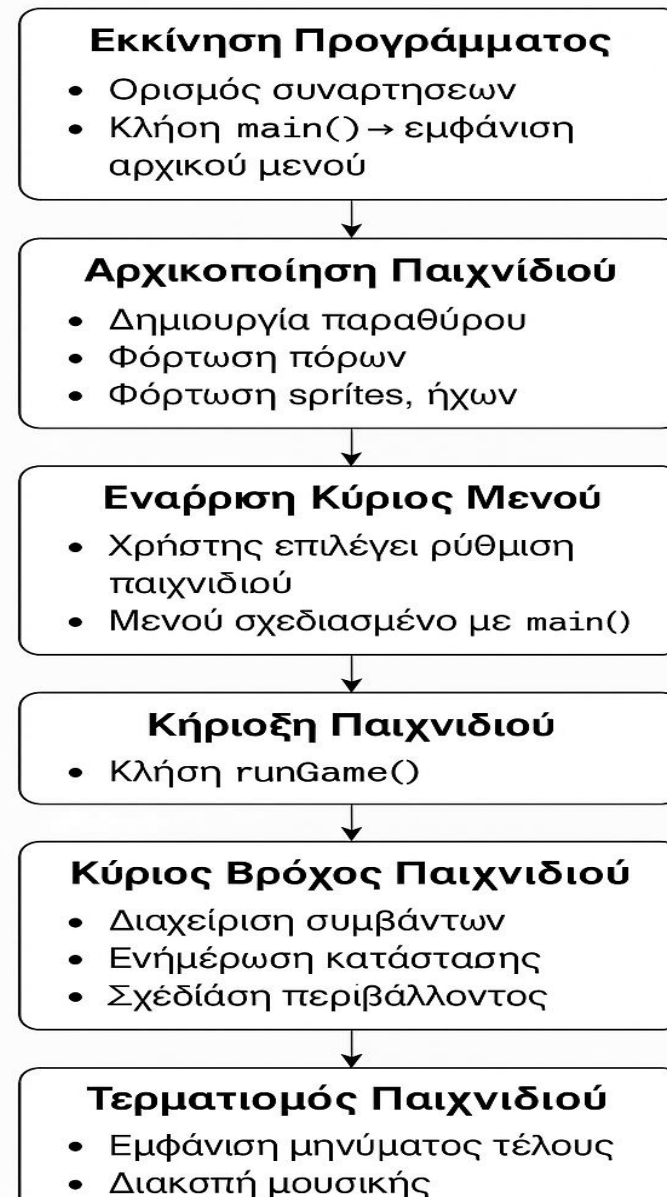
Πώς λειτουργεί το σύστημα;



- Δημιουργείτε το παράθυρο στη main και φορτώνονται εικόνες και ήχοι
- Εμφανίζει μενού επιλογής *resolution & 1 or 2 players*
- Εκτελείται ένας βρόχος που διαχειρίζεται τα *events* του παιχνιδιού

Διάγραμμα ροής

Διάγραμμα Ροής Παιχνιδιού (SFML)



Τεχνολογίες που χρησιμοποιήθηκαν & λόγοι επιλογής

- *Visual studio*
- *SFML*



Κώδικας & υλοποίηση

Πώς υλοποιήθηκε η βασική λειτουργικότητα;

- Φόρτωση *background* (φόντου)
- Φόρτωση *sprites* / ήχου
- Έλεγχος *movement* (κίνησης)
- *Spawn* εμποδίων (*obstacles*)
- Έλεγχος *collision* (σύγκρουσης)

```
int main() {
    Vector2u windowSize(1500, 1000);
    RenderWindow window(VideoMode(windowSize.x, windowSize.y), "Drunk Drivers");

    Texture backgroundTexture, playerTexture, obstacleTexture;
    Font font;
    if (!backgroundTexture.loadFromFile("C:\\Users\\nikos\\OneDrive\\Vnoloyισής\\ChatGPT Image 9 Mar 2025, 10_21_37 μ.μ..png")) ||
        !playerTexture.loadFromFile("C:\\Users\\nikos\\OneDrive\\Vnoloyισής\\ptp3cxnvg00c1.png") ||
        !obstacleTexture.loadFromFile("C:\\Users\\nikos\\OneDrive\\Vnoloyισής\\ptp3cxnvg00c1.png") ||
        !font.loadFromFile("C:\\Users\\nikos\\OneDrive\\Vnoloyισής\\nikscrc.ttf") {
        cerr << "Error: Could not load one or more resources." << endl;
        return -1;
    }

    SoundBuffer engineBuffer, secondBuffer;
    if (!engineBuffer.loadFromFile("C:\\Users\\nikos\\OneDrive\\Vnoloyισής\\audiomass-output.wav"))
        cerr << "Error: Could not load engine sound." << endl;
    if (!secondBuffer.loadFromFile("C:\\Users\\nikos\\OneDrive\\Vnoloyισής\\repeat-output-20s-loopable.wav"))
        cerr << "Error: Could not load second sound." << endl;
    Sound engineSound(engineBuffer), secondSound(secondBuffer);
    engineSound.setLoop(false); engineSound.setVolume(50.f);
    secondSound.setLoop(true); secondSound.setVolume(50.f);
}
```

```
if (Keyboard::isKeyPressed(Keyboard::Left)) player1.move(-300.f * deltaTime, 0.f);
if (Keyboard::isKeyPressed(Keyboard::Right)) player1.move(300.f * deltaTime, 0.f);
if (Keyboard::isKeyPressed(Keyboard::Up)) player1.move(0.f, -300.f * deltaTime);
if (Keyboard::isKeyPressed(Keyboard::Down)) player1.move(0.f, 300.f * deltaTime);
FloatRect p1Bounds = player1.getGlobalBounds();
Vector2f p1Pos = player1.getPosition();
p1Pos.x = max(roadLeft, min(p1Pos.x, roadRight - p1Bounds.width));
p1Pos.y = max(roadTop, min(p1Pos.y, roadBottom - p1Bounds.height));
player1.setPosition(p1Pos);
```

```
for (const auto& obstacle : obstacles) {
    if (player1.getGlobalBounds().intersects(obstacle.getGlobalBounds())) {
        engineSound.stop(); secondSound.stop(); return false;
    }

    if (mode == TWO_PLAYER && player2.getGlobalBounds().intersects(obstacle.getGlobalBounds())) {
        engineSound.stop(); secondSound.stop(); return false;
    }
}
```


Σημαντικοί αλγόριθμοι : rungame()

```
float roadTop = 0.f, roadBottom = static_cast<float>(windowSize.y);

Sprite background1(backgroundTexture), background2(backgroundTexture);
background1.setScale(scaleX, scaleY); background2.setScale(scaleX, scaleY);
background1.setPosition(0.f, 0.f); background2.setPosition(0.f, -static_cast<float>(windowSize.y));

Sprite player1(playerTexture), player2(playerTexture);
player1.setScale(0.17f, 0.17f); player2.setScale(0.17f, 0.17f);
FloatRect p1Bounds = player1.getGlobalBounds();
player1.setPosition(windowSize.x / 2.f - p1Bounds.width / 2.f, windowSize.y * 0.8f - p1Bounds.height / 2.f);
FloatRect p2Bounds = player2.getGlobalBounds();
player2.setPosition(windowSize.x / 2.f - p2Bounds.width / 2.f, windowSize.y * 0.9f - p2Bounds.height / 2.f);

vector<Sprite> obstacles;
Clock spawnClock, gameClock, deltaClock;
float spawnInterval = 1.0f, obstacleSpeed = 400.f, speedIncreaseRate = 0.f;
Text counterText;
counterText.setFont(font); counterText.setCharacterSize(24);
counterText.setFill(Color::White); counterText.setPosition(10.f, 10.f);

float backgroundSpeed = 100.f, backgroundSpeedMax = 500.f, backgroundAcceleration = 200.f;

while (window.isOpen()) {
    Event event;
    while (window.pollEvent(event)) {
        if (event.type == Event::Closed) { engineSound.stop(); secondSound.stop(); window.close(); return false; }
    }

    float deltaTime = deltaClock.restart().asSeconds();
    if (backgroundSpeed < backgroundSpeedMax)
        backgroundSpeed = min(backgroundSpeed + backgroundAcceleration * deltaTime, backgroundSpeedMax);

    background1.move(0.f, backgroundSpeed * deltaTime);
    background2.move(0.f, backgroundSpeed * deltaTime);
    if (background1.getPosition().y >= static_cast<float>(windowSize.y))
        background1.setPosition(0.f, background2.getPosition().y - static_cast<float>(windowSize.y));
    if (background2.getPosition().y >= static_cast<float>(windowSize.y))
        background2.setPosition(0.f, background1.getPosition().y - static_cast<float>(windowSize.y));
```

```
        p2Pos.x = max(roadLeft, min(p2Pos.x, roadRight - p2Bounds.width));
        p2Pos.y = max(roadTop, min(p2Pos.y, roadBottom - p2Bounds.height));
        player2.setPosition(p2Pos);
    }

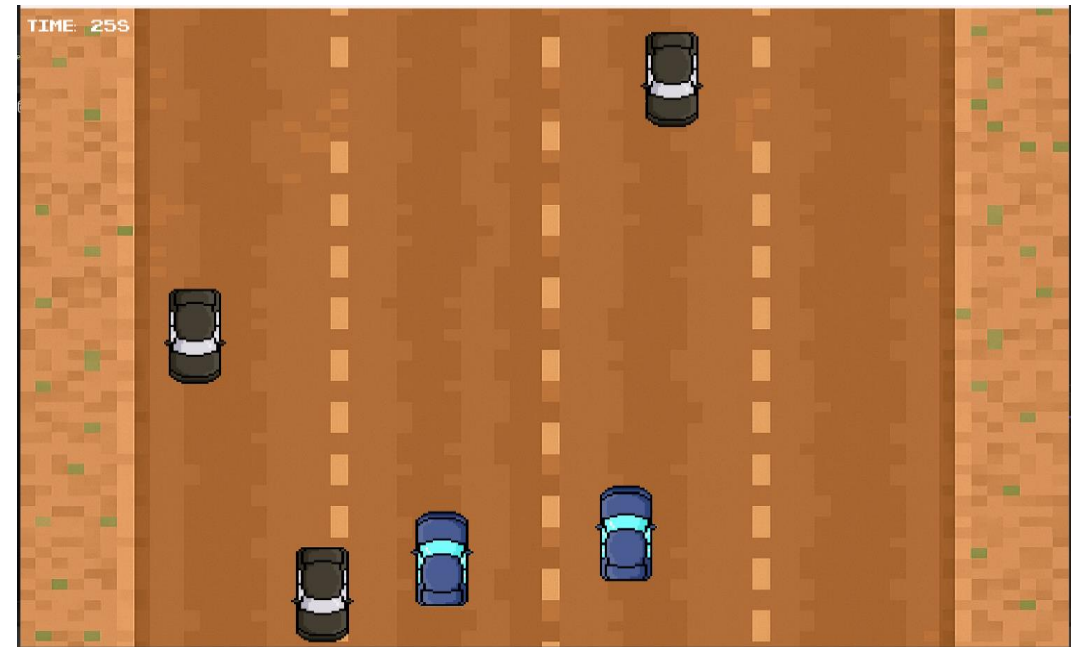
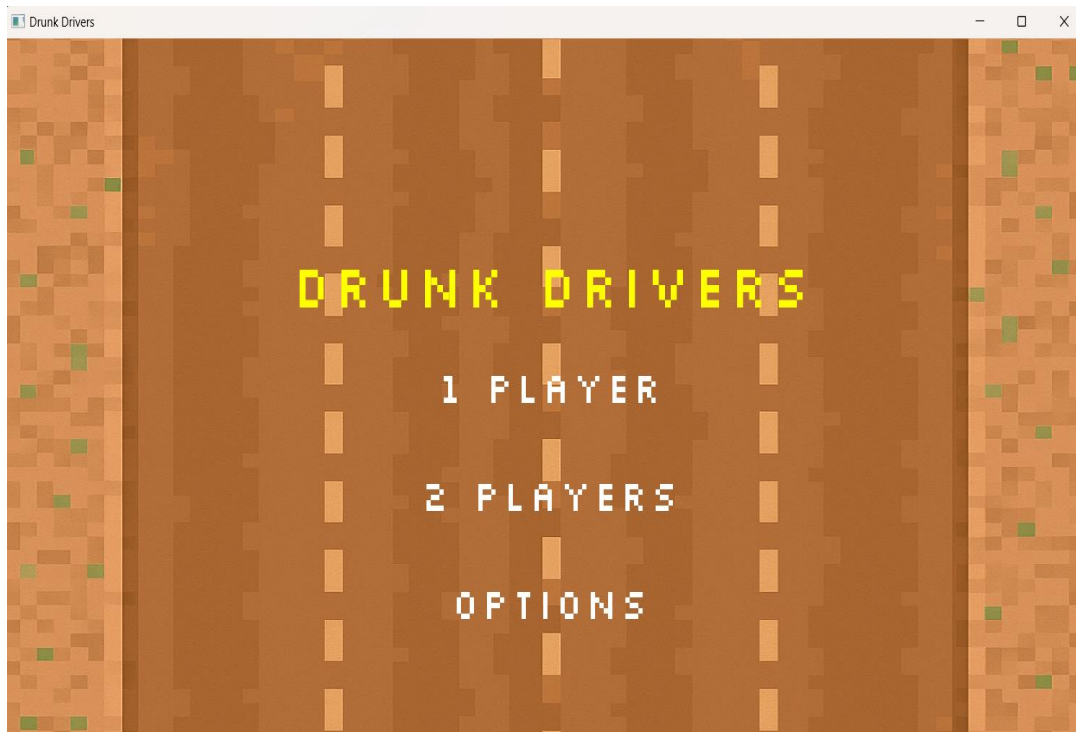
    if (spawnClock.getElapsedTime().asSeconds() > 1.0f) {
        Sprite obstacle(obstacleTexture);
        float minX = roadLeft, maxX = roadRight - obstacle.getGlobalBounds().width;
        obstacle.setPosition(minX + static_cast<float>(rand()) / RAND_MAX * (maxX - minX), -50.f);
        obstacle.setScale(0.17f, 0.17f);
        obstacles.push_back(obstacle);
        spawnClock.restart();
    }

    obstacleSpeed += speedIncreaseRate * deltaTime;
    for (auto& obstacle : obstacles) {
        obstacle.move(0.f, obstacleSpeed * deltaTime);
        FloatRect oBounds = obstacle.getGlobalBounds();
        Vector2f oPos = obstacle.getPosition();
        oPos.x = max(roadLeft, min(oPos.x, roadRight - oBounds.width));
        obstacle.setPosition(oPos);
    }

    for (const auto& obstacle : obstacles) {
        if (player1.getGlobalBounds().intersects(obstacle.getGlobalBounds())) {
            engineSound.stop(); secondSound.stop(); return false;
        }
        if (mode == TWO_PLAYER && player2.getGlobalBounds().intersects(obstacle.getGlobalBounds())) {
            engineSound.stop(); secondSound.stop(); return false;
        }
    }
}
```

Αποτελέσματα & demo

screenshots



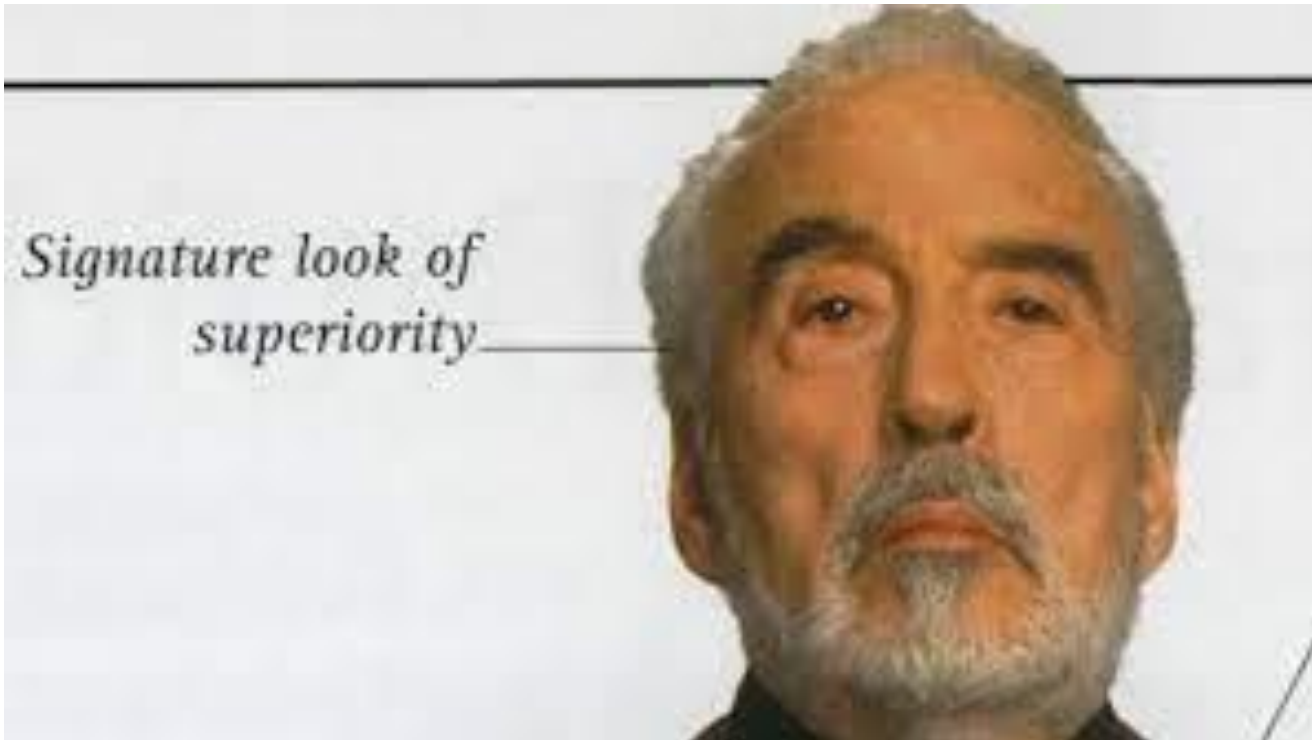
Σύγκριση με AI-generated code



**Χρησιμοποιήσατε
AI (π.χ. ChatGPT,
Copilot) για
κώδικα;**

Φυσικά και ΟΧΙ

Διαφορές στον κώδικα που γράψατε εσείς vs AI



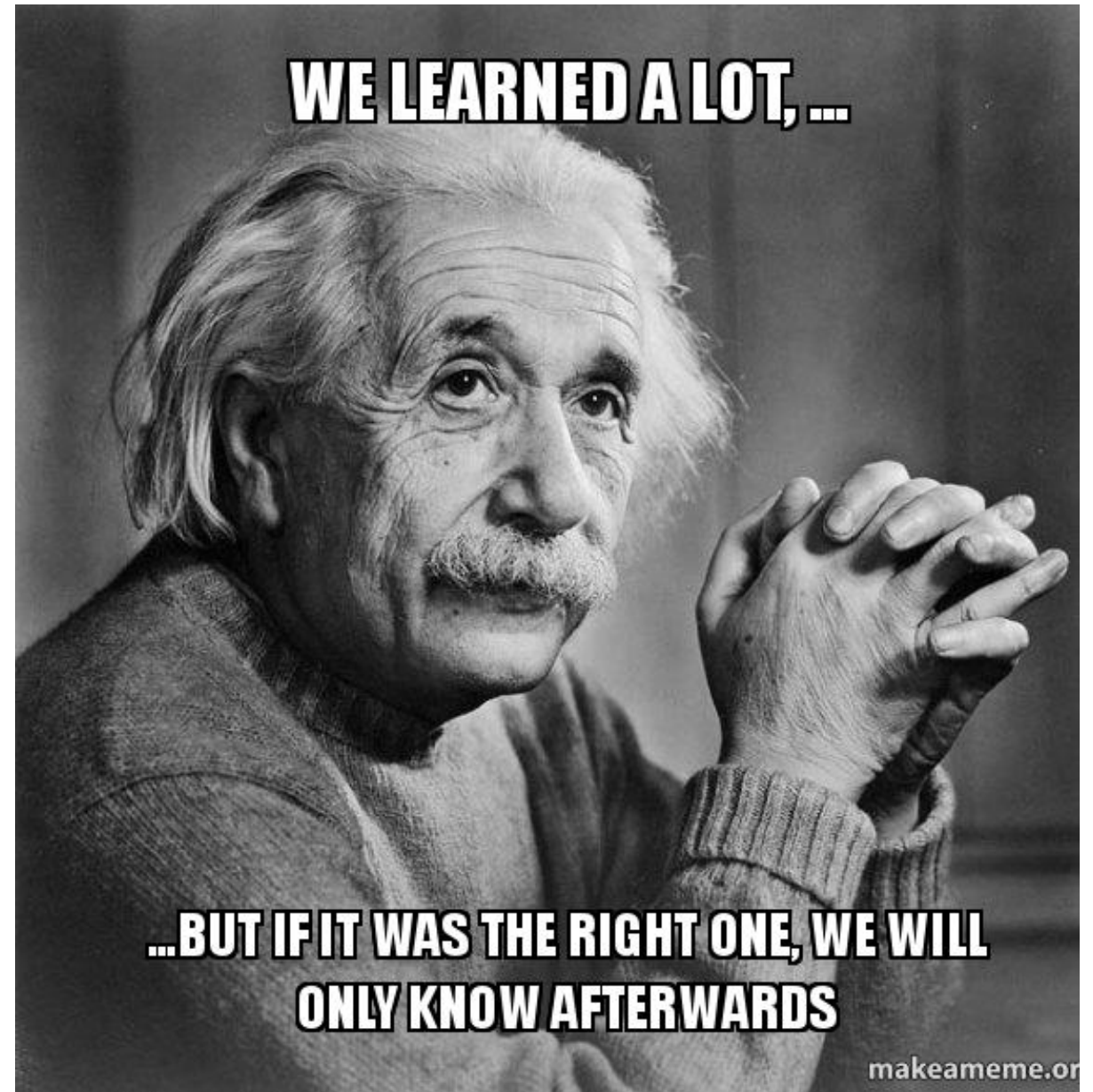
-
- Ο κώδικας του AI ήταν πιο οργανωμένος, σωστός και λειτουργικός

<-- chat gpt probably

Συμπεράσματα & lessons learned

Τι μάθατε από το project;

- Μάθαμε πώς να χειριζόμαστε το *Visual Studio* και την *SFML*
- Πώς να βάζουμε ήχους, εικόνες κλπ
- Ο προγραμματισμός δεν είναι εύκολος



Τι θα μπορούσε να βελτιωθεί;

- *(ΤΙΠΟΤΑ ΕΙΝΑΙ ΤΕΛΕΙΟ)*
- *Dynamic resolution*
- *Sound effects*
- *animation*



Μελλοντικές προεκτάσεις του project

- *Multiple car colours (mandatory)*
- *Add Story*
- *Ο δρομος να εχει και στροφες*

Me looking at my future



Τελος

