

UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAȘI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DOMENIUL: Calculatoare și Tehnologia Informației
SPECIALIZAREA: Tehnologia Informației

LUCRARE DE DIPLOMĂ

Coordonator științific:
Prof. dr. ing. Mitică Craus

Absolvent:
Florin Nistor

Iași, 2024

UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAȘI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DOMENIUL: Calculatoare și Tehnologia Informației
SPECIALIZAREA: Tehnologia Informației

Cat's Valley

LUCRARE DE DIPLOMĂ

Coordonator științific:
Prof. dr. ing. Mitică Craus

Absolvent:
Florin Nistor

Iași, 2024

DECLARAȚIE DE ASUMARE A AUTENTICITĂȚII PROIECTULUI DE DIPLOMĂ

Subsemnatul _____,
legitimat cu _____ seria _____ nr. _____, CNP _____
autorul lucrării _____

elaborată în vederea susținerii examenului de finalizare a studiilor de licență, programul de studii _____ organizat de către Facultatea de Automatică și Calculatoare din cadrul Universității Tehnice „Gheorghe Asachi” din Iași, sesiunea _____ a anului universitar _____, luând în considerare conținutul Art. 34 din Codul de etică universitară al Universității Tehnice „Gheorghe Asachi” din Iași (Manualul Procedurilor, UTI.POM.02 - Funcționarea Comisiei de etică universitară), declar pe proprie răspundere, că această lucrare este rezultatul propriei activități intelectuale, nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române (legea 8/1996) și a convențiilor internaționale privind drepturile de autor.

Data

Semnătura

Cuprins

Introducere	1
1 Fundamentarea teoretică și documentarea bibliografică	5
1.1 Java ca limbaj de programare pentru jocuri	5
1.1.1 Context	5
1.1.2 Portabilitate	6
1.1.3 Garbage Collection	6
1.2 Ce este un Engine de Joc?	7
1.2.1 Concepte relevante în cadrul Engine-urilor de Joc	8
1.2.1.1 Scena (Scene) și Obiecte (Objects)	8
1.2.1.2 Sistem de Entități și Componente (Entity-Component System - ECS)	9
1.2.1.3 Inteligența Artificială (AI)	10
1.2.1.4 Pathfinding	10
1.3 Map (harta)	11
1.4 Dungeon and Procedural Generation (Generare Procedurală)	11
2 Proiectarea aplicației	13
2.1 "	13
2.1.1 Biblioteci utilizate	13
2.1.2 Proiectare si metodologie	13
3 Implementarea aplicației	15
3.1	15
4 Testarea aplicației și rezultate experimentale	17
Concluzii	19
Bibliografie	21
Anexe	23
1 Componente Software	23
2 Componente Hardware	24
3 Codul funcției xyz()	25
4 Tabel lung pe mai multe pagini	27
5 Tabel simplu	29
6 Cod în Python	30
7 Cod în Kotlin (mai lung de o pagină)	33
8 Cod în XML (preluat din fisier)	35

Cat's Valley

Florin Nistor

Rezumat

Lucrarea de diplomă/disertație va conține un scurt rezumat (abstract – termenul în engleză) de maximum o pagină, care expune ideile principale ale tezei, contribuția autorului la realizarea temei propuse precum și concluziile obținute. Formulați ideile principale ale lucrării în propoziții sau fraze cât mai simple, fără amănunte nesemnificative.

Recomandare neoficială: Rezumatul se compune, de obicei, la finalizarea redactării lucrării de licență/disertație, după finalizarea capitolului Concluzii

- Reflectă capacitatea absolventului de a prezenta sintetic tema abordată, obiectivul, proiectarea aplicației, modul de implementare și rezultatele obținute;
- Rezumatul lucrării în format pdf va fi publicat pe site-ul facultății, înaintea susținerii publice.

Introducere

Introducerea va avea 2–3 pagini care vor conține motivația alegerii temei, relevanța și contextul temei alese, obiectivele generale ale lucrării, metodologia și instrumentele utilizate și o scurtă descriere a structurii lucrării (titlul capitolelor, o scurtă descriere și legătura dintre acestea).

În acest capitol nu se introduc figuri, table sau listing-uri de cod. Pot fi în schimb referite!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit.

Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit.

Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Capitolul 1. Fundamentarea teoretică și documentarea bibliografică

Jocurile 2D sunt fundația industriei de jocuri video încă de la început. Chiar și în prezent, unde industria este dominată de jocuri 3D avansate, jocurile 2D continuă să fie îndrăgite și jucate, datorită simplității lor, accesibilității și mai ales capacitatea de a crea povești, experiențe și un design aparte, fiind un principal to-go pentru a îți da frâu liber imaginației.

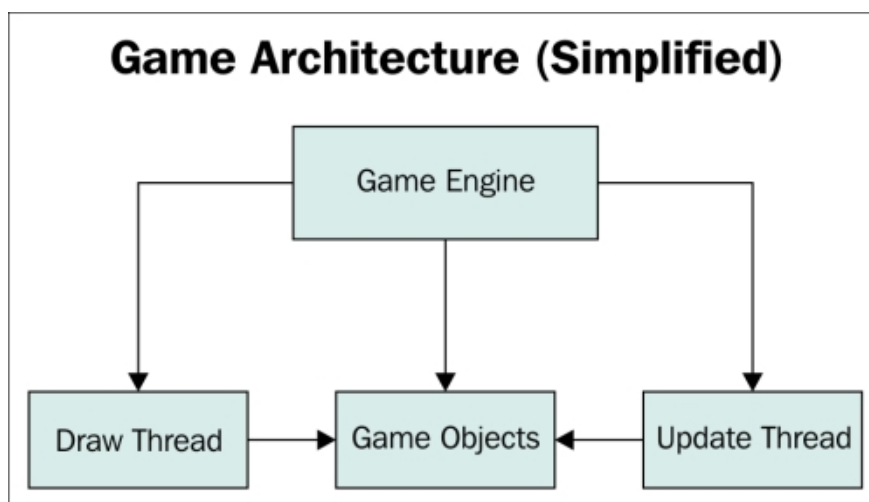


Figura 1.1. O arhitectura simplificata a codului sursa¹

Această lucrare își propune implementarea unui engine de la zero. În prezent, există deja pe piață o varietate de Engine-uri deja complete, fiind unul pentru fiecare stil de joc. Acestea, mi-ar ușura lucrarea, dar nu este ceea ce îmi doresc. Caut să înțeleg cât mai bine întreg procesul, de la zero. "We take the piano-playing example: one day, you are playing scales and simple melodies, but you dedicate one hour a day to practice, and years later, you can play grade 8 Bach without really remembering how. Learning is incremental, gradual, iterative, and transformational; it is a process and an outcome."²

1.1. Java ca limbaj de programare pentru jocuri

1.1.1. Context

Java, este un limbaj de programare orientat pe obiecte, făcând astfel posibilă o ușoară programare a jocurilor. Este cunoscut pentru portabilitatea sa și performanță. Până în prezent, Java a fost folosit pentru a dezvolta o varietate de aplicații și jocuri precum: Minecraft, Runescape și multe altele. În contextul dezvoltării de jocuri 2D, Java oferă câteva avantaje.

¹Preluat de pe <https://subscription.packtpub.com/book/game-development/9781783551774/1/ch01v11sec10/game-architecture>

²Preluat de la Hainey, T., Baxter, G. (2022). Writing Successful Undergraduate Dissertations in Games Development and Computer Science. Routledge Taylor Francis Group. <https://www.routledge.com/Writing-Successful-Undergraduate-Dissertations-in-Games-Development-and/Hainey-Baxter/p/book/9780367517083>

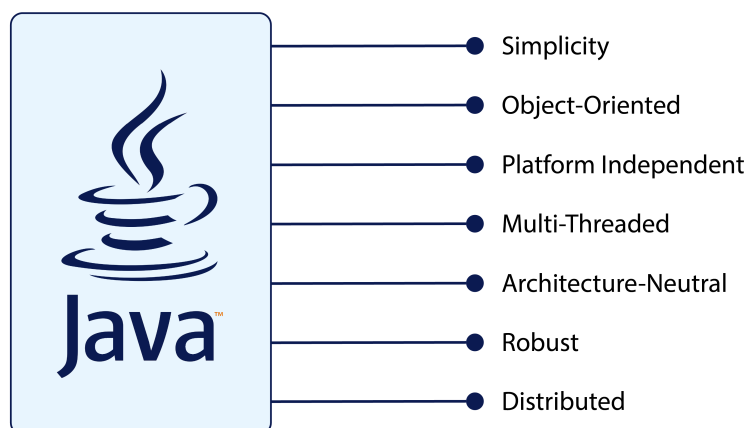


Figura 1.2. Java Documentation³

1.1.2. Portabilitate

Datorită Java Virtual Machine (JVM), jocurile în Java pot rula pe orice platformă care suportă JVM precum Windows, Mac, Linux, Android, iOS, Raspberry Pi, Browsers Web, făcând astfel o ușoară tranziție de lansare MultiPlatform.

1.1.3. Garbage Collection

Garbage Collection este un proces automatizat de gestiune a memoriei folosit în Java. Are ca rol identificarea și eliminarea memoriei care nu mai este utilizată, prevenind astfel scurgerile de memorie și alte probleme ce apar la gestiunea manuală a memoriei.

În dezvoltarea de jocuri, gestiunea eficientă a memoriei este crucial de important pentru a asigura performanță constantă și pentru a evita probleme ce pot apărea foarte ușor dacă manipulăm noi memoria fără grijă.

³Informații preluate de pe <https://docs.oracle.com/javase/8/docs/>

1.2. Ce este un Engine de Joc?

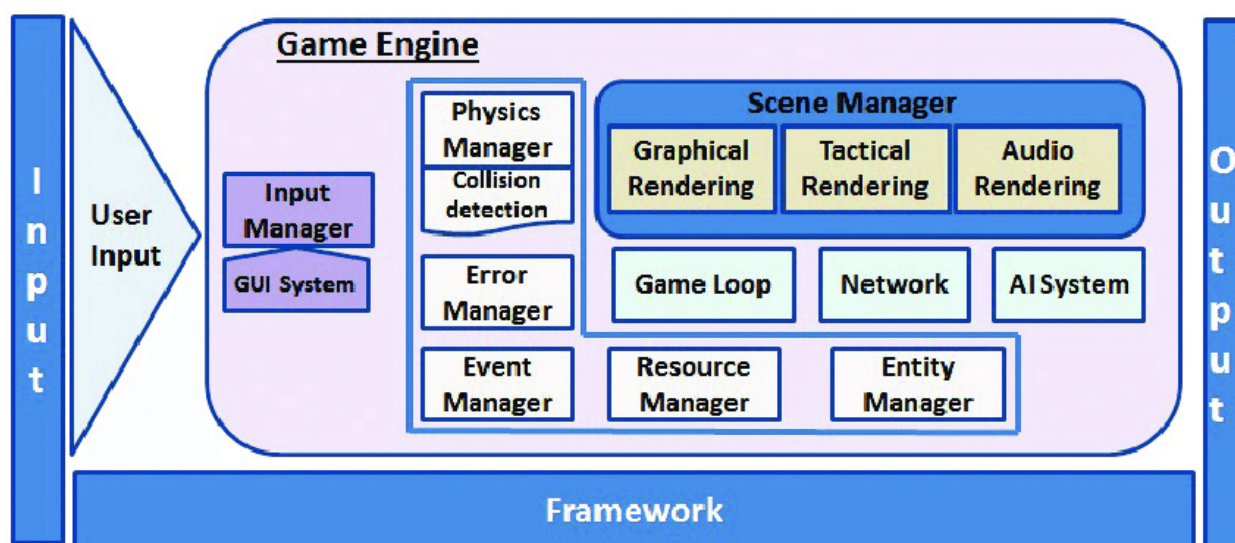


Figura 1.3. Game Engine Solutions⁴

Un engine de joc (sau motor de joc) este un software care oferă o bază pentru dezvoltarea și rularea jocurilor video. Simplifică și eficientizează procesul de dezvoltare, furnizând o serie de componente esențiale, precum redarea graficii, fizica jocului, animații, sunet și multe altele.

- **Grafică:**

- 2D și 3D Rendering: Capabilitatea de redare a graficii 2D și 3D. Aceasta include rasterizarea, iluminarea, umbrirea și alte efecte vizuale.

- **Fizică:**

- **Simularea Dinamicii:**
 - * Calcularea mișcărilor pe o axă de coordonate și interacțiunilor fizice ale obiectelor.
 - * Aceasta include gravitația, coliziunile și interacțiunile cu ele.
- **Rigid Body Physics:**
 - * Simularea mișcărilor și interacțiunilor cu obiecte solide.
- **Soft Body Physics:**
 - * Simularea obiectelor deformabile, cum ar fi țesuturile sau gelatinele (slime).

- **Gestionarea Sunetului (Motor Audio):**

- Constă în redarea efectelor sonore pentru interacțiuni cu obiecte și muzică ambientală de fundal. În cadrul unui motor 3D, există posibilitatea redării sunetului într-un spațiu tridimensional, ajustând volumul și direcția sunetului în funcție de poziția jucătorului.

- **Logica Jocului:**

- Fiecare joc are o logică diferită. Poate un joc se axează doar pe control de mouse și este un joc de simulare. Alt joc poate să fie cu alte controale unice și se bazează pe Event-uri și story.

⁴Informații preluate de pe https://www.researchgate.net/publication/323179176_Game_Engine_Solutions

- Fiecare logică a jocului reflectă gândirea programatorului asupra proiectului.
- Spre exemplu, jocul Undertale este un joc story based și are un dialog ce poate să fie influențat de alegerile jucătorului. Jocul este unul foarte popular și jucat iar logica din spate a dialogului este o succesiune de 2000 de if-uri.

- **Animații:**

- Gestionarea și redarea animațiilor pentru personaje și obiecte. În majoritatea cazurilor, în special la un motor 2D, animațiile sunt o succesiune de imagini ce simulează prin ciclitatea imaginilor o animație.
- Pentru motoarele 3D, există o opțiune mai realistă. Animația personajelor utilizând un sistem de oase și articulații.

- **Gestionarea Resurselor:**

- Managementul Asset-urilor: Organizarea și gestionarea resurselor de joc, cum ar fi texturi, modele 3D, sunete, script-uri dacă este cazul. Tot în cadrul gestionării resurselor există termenul de Loading/Streaming prin care se încearcă gestionarea dinamică a resurselor pe măsură ce jucătorul navighează prin joc.

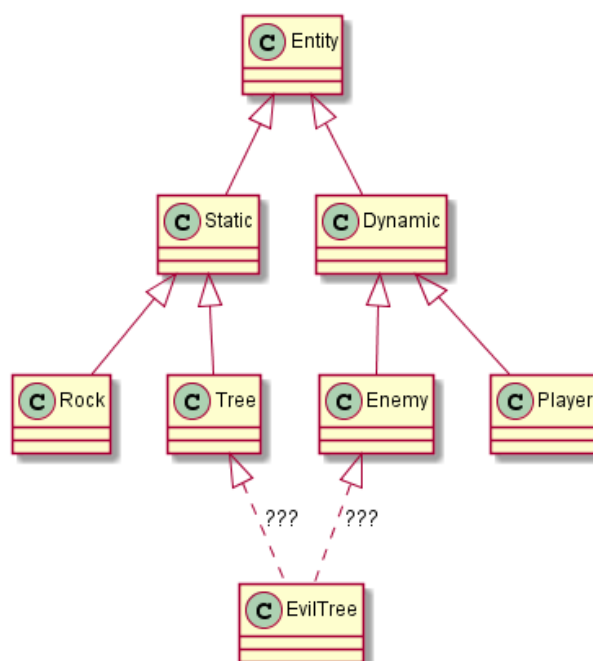
Un engine de joc este esențial pentru o bună dezvoltare a jocurilor, oferind o infrastructură pentru construcția complexă și pentru performanțe. Acestea includ componente pentru redare grafică, fizică, sunet, animații, gestiunea resurselor, rețea și interfața utilizatorului. Utilizarea unui engine de joc nu doar accelerează procesul de dezvoltare, dar și asigură că jocurile pot ajunge la un nivel înalt de calitate și performanță.

1.2.1. Concepte relevante în cadrul Engine-urilor de Joc

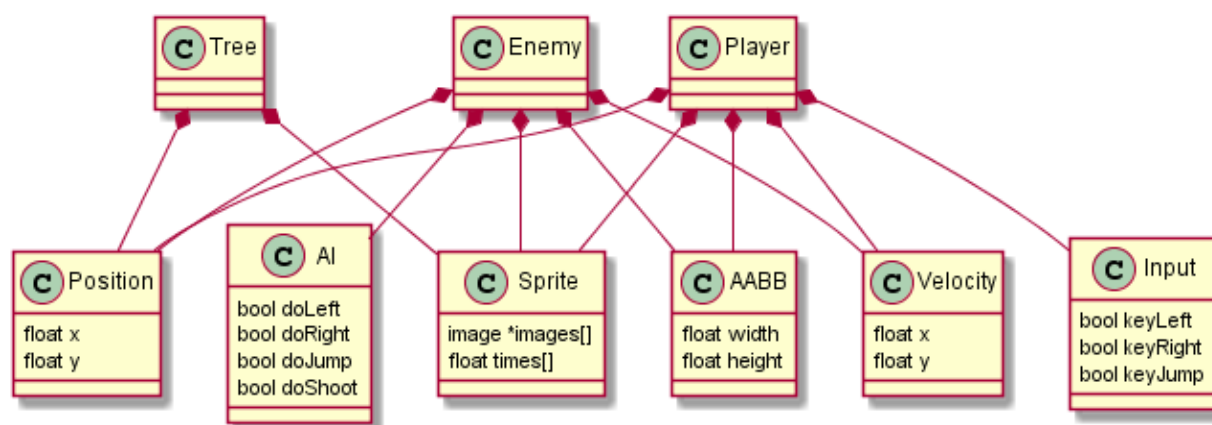
1.2.1.1. Scena (Scene) și Obiecte (Objects)

Scena reprezintă un container pentru toate obiectele dintr-un anumit context sau nivel al jocului. În cadrul unei scene, avem definit mediul, personajele, obiectele și regulile de joc specifice segmentului. Obiectele sunt entități individuale dintr-o scenă, acestea pot include NPC-uri (Non-Player Character), elemente de mediu (copaci, apă, case, pietre), proiectile, etc. Fiecare obiect având proprietățile și comportamentele specifice.

1.2.1.2. Sistem de Entități și Componente (Entity-Component System - ECS)

Figura 1.4. Entity-Component System Exemple⁵

Entitățile sunt unități care reprezintă obiecte de joc și care sunt identificate printr-un ID unic. Din clasa principală entitate care manageruește fiecare obiect/entitate, derivă câte o clasă ce are proprietățile obiectului. Dacă urmărim exemplul din figura 1.4, un copac (Tree) este un obiect static și fix. Inamicul (Enemy) este o unitate dinamică, ceea ce înseamnă că se poate deplasa, ataca, fugii, etc. Acum, dorim un copac, care atunci când se încearcă a fi tăiat, acesta să prindă viață și din entitate Tree să devină entitate Enemy. În cazul ăsta, păstrăm proprietățile Tree (spre exemplu dacă este distrus să scape (drop) bucăți de lemn, dar în același timp să aibă și proprietățile enemy precum deplasare, atac, HP (Health Points).

Figura 1.5. Component System Exemple⁶

⁵Preluat de pe <https://www.gamedev.net/articles/programming/general-and-gameplay-programming/understanding-component-entity-systems-r3013/>

⁶Preluat de pe <https://www.gamedev.net/articles/programming/general-and-gameplay-programming/understanding-component-entity-systems-r3013/>

Componentele sunt blocuri de date care sunt atașate entităților pentru a defini atributele acestora, un exemplu îl avem în figura 1.5. Un Tree este Static deci are doar o poziție pe axă (x și y) și are un Sprite ce conține aspectul copacului. Un inamic are toate componentele pe care copacul le are și multe în plus precum AI, prin care inamicul se poate mișca aleatoriu, o dimensiune a lui, poate un inamic ocupa o tilă dar un inamic de tip Boss ocupă 4 tile. Velocitate prin care parametrizăm viteza cu care acesta se deplasează, etc. Componentele joacă un rol important în dezvoltarea unui game engine.

Sistemele sunt funcționalități care operează pe entități care conțin anumite componente, implementând logica jocului. De exemplu sistemul de mișcare, sistem de coliziune, etc.

1.2.1.3. Inteligența Artificială (AI)

Prin inteligență artificială putem să ajungem ușor cu gândul la un model de antrenat. Din păcate pentru industria de jocuri, nu e practic să scrii un model neuronal pentru a observa o varietate de jucători și de a ajunge în punctul de a învăța singur întreg mecanismul. Deoarece jocul fiind în producție nu are nici un jucător și lansând un joc cu gândul că la un moment dat AI-ul o să învețe și-ar pierde mulți jucători din prima, aceștia considerând că este un joc nefinalizat.

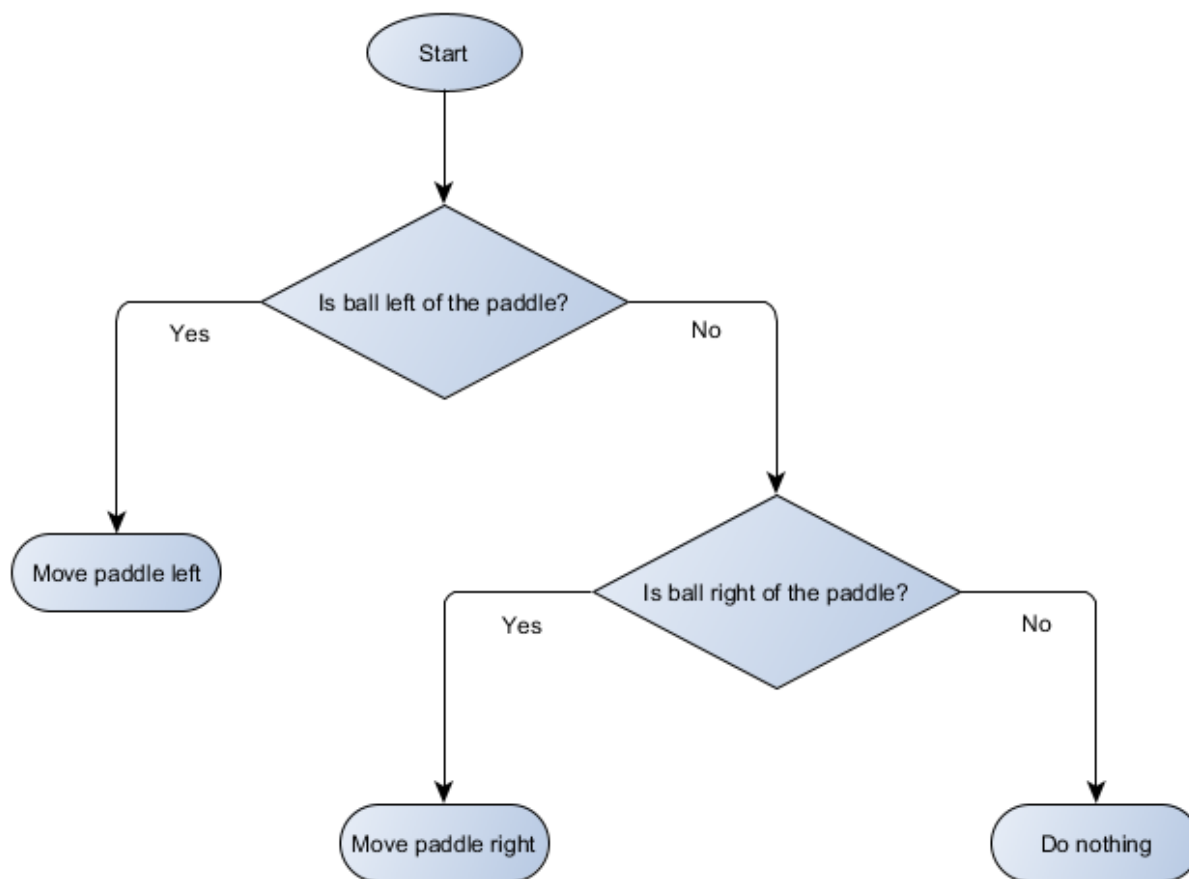


Figura 1.6. AI Decision Tree Example⁷

În schimb, putem implementa un Decision Tree ce poate simula AI-ul.

1.2.1.4. Pathfinding

⁷Preluat de pe <https://www.gamedev.net/articles/programming/artificial-intelligence/the-totalbeginners-guide-to-game-ai-r4942/>

1.3. Map (harta)

1.4. Dungeon and Procedural Generation (Generare Procedurală)

Capitolul 2. Proiectarea aplicației

Modulele generale ale aplicației sunt:

-
-
-

2.1. "

2.1.1. Biblioteci utilizate

-
-
-
-
-

2.1.2. Proiectare si metodologie

Capitolul 3. Implementarea aplicației

3.1.

Capitolul 4. Testarea aplicației și rezultate experimentale

Concluzii

Concluziile lucrării (1–2 pagini) în care se regăsesc cele mai importante concluzii din lucrare, care pornind de la obiectivele propuse în introducerea lucrării vor conține opinia personală a autorului privind rezultatele obținute în lucrare, o comparație a acestora cu alte proiecte/produse similare precum și posibile direcții de dezvoltare.

- Gradul în care s-a realizat tema propusă (motivarea eventualelor obiective modificate);
- Evidențierea concisă a contribuțiilor/soluțiilor personale (dacă este cazul);
- Comparație cu alte proiecte similare;
- Posibile direcții de dezvoltare.

În acest capitol nu se introduc figuri, table sau listing-uri de cod. Pot fi în schimb referite!

Concluziile finale au un rol semnificativ în cadrul lucrării de diplomă, care trebuie pus în evidență în special în momentul susținerii acesteia în fața comisiei. Aici trebuie prezentate, în câteva pagini, cele mai importante rezultate ale muncii depuse de candidat și potențiale direcții de dezvoltare ulterioară a temei abordate. Acest capitol oferă o sinteză a realizărilor obținute, precum și avantajele și neajunsurile soluțiilor alese. Concluziile nu trebuie să aducă rezultate sau interpretări noi, ci doar să le evidențieze pe cele din conținutul lucrării.

Direcții viitoare de dezvoltare

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Lecții învățate pe parcusul dezvoltării lucrării de diplomă

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Bibliografie

Anexe

Anexe prezintă doar elementele specifice proiectului!

Anexele (dacă este cazul) constituie o secțiune separată a lucrării care nu se numerează ca și capitol. Anexele se numerează crescător cu numere arabe (ex.: Anexa 1, Anexa 2 etc.). Anexele vor conține scheme, diagrame, secvențe din codul sursă.

Anexa 1. Componente Software

- diagramele UML care referă numai la componentele dezvoltate de student și care datorită complexității pot fi listate pe o foaie de tip A3 sau A2.
- cod sursă numai pentru componentele dezvoltate de către student.

Anexa 2. Componente Hardware

- schemele electrice finale realizate într-un CAD de profil;
- schemele cablajelor realizate pentru implementare, realizate într-un CAD de profil;
- informații suplimentare despre implementarea și testarea aplicației (de ex. capturi de osciloscop);
- schemele standard ce vor fi folosite pentru testare (pseudocod sau schemă logică);
- extrase din foi de catalog.

Anexa 3. Codul funcției xyz()

În Figura A.1 este prezentat unul dintre cele mai impunătoare vehicule ale Imperiului.

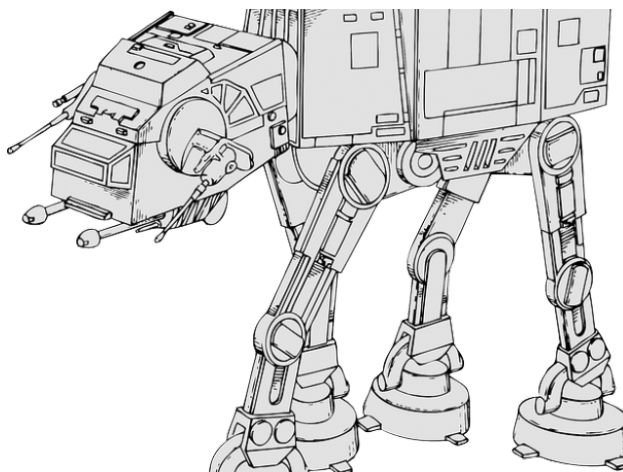


Figura A.1. AT⁸

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue

⁸ imagine preluată de pe un site web care nu „merită” trecut la bibliografie <https://www.pngitem.com/>

quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus. (Figura A.2)

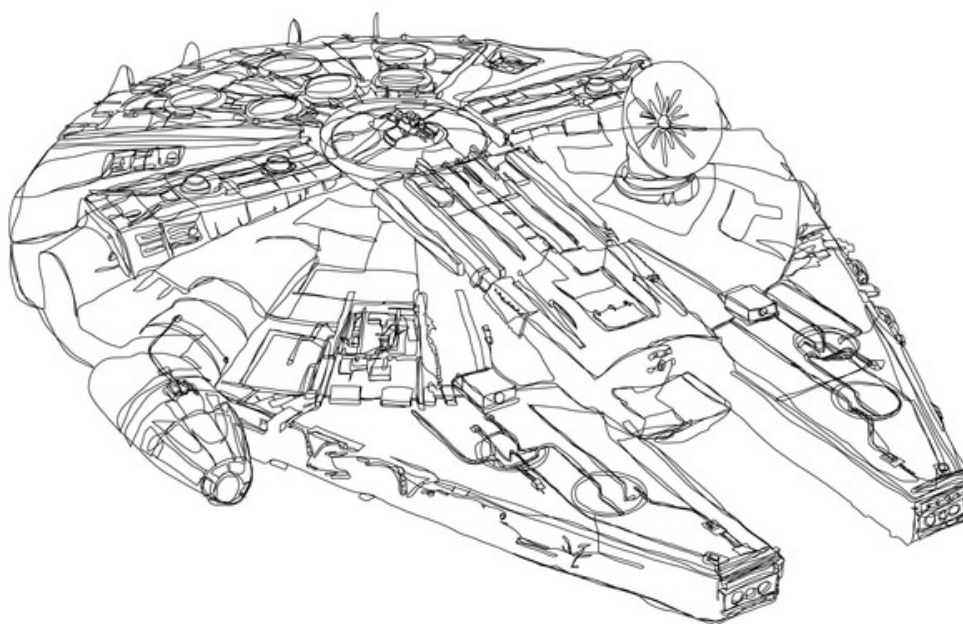


Figura A.2. Millennium Falcon⁹

⁹ imagine preluată de pe un site web care nu „merită” trecut la bibliografie <http://clipart-library.com/>

Anexa 4. Tabel lung pe mai multe pagini

Malware (software rău intenționat) este un termen folosit pentru a descrie orice program sau cod care ocolește procesele de control al accesului, este creat cu intenția de a ataca un sistem de calcul sau rețea, pentru a provoca daune sau a compromite acel sistem. În categoria malware se includ: viruși, *ransomware*, *keyloggers*, troieni, viermi, *spyware* etc (Tabelul A.1). În vocabularul curent, termenul de "virus" este folosit abuziv pentru orice tip de malware.

Tabelul A.1. Tipuri de malware

<i>Tip</i>	<i>Descriere</i>	<i>Exemple elocvente</i>
Virus	cod executabil malițios, atașat la fișiere sau programe legitime. Majoritatea virușilor necesită declanșarea de către utilizatorul final. Se răspândește prin intermediul unităților USB, discurilor optice, partajărilor în rețea sau e-mail-ului	Brain, Michelangelo, Melissa
Ransomware	conceput pentru a ține captiv un sistem informatic, de obicei prin criptarea datelor esențiale, dezactivează accesul victimei la date până la efectuarea unei plăți către atacator	WannaCry, Petya
Fileless Malware	face modificări la fișierele native ale sistemului de operare	Astaroth
Spyware	proiectat pentru a urmări acțiunile utilizatorului și a colecta date despre activitatea victimei, fără cunoștința acesteia	DarkHotel
Adware	livrează anunțuri în mod automat. Unele sunt însoțite de programe <i>spyware</i>	Fireball
Troian	execută operațiuni rău intenționate disimulate ca operațiuni dorite. Se deghizează în codul unui program cunoscut și se atașează de fișiere non-executabile	Emotet
Worms	se reproduce singur, se răspândește într-o rețea prin replicare, prin exploatarea vulnerabilităților din rețele. Au tipare similare, inclusiv o vulnerabilitate activă, o modalitate de a se propaga singuri și conțin un <i>payload</i>	Stuxnet

Continuarea tabelului A.1

<i>Tip</i>	<i>Descriere</i>	<i>Exemple elocvente</i>
Rootkit	conceput pentru a modifica sistemul de operare, astfel încât computerul să poată fi accesat de la distanță, printr-o porțiță de acces. Rootkit-urile modifică privilegiile de acces, fișierele de sistem și instrumentele de monitorizare a sistemului, ceea ce le face foarte greu de detectat și eliminat	Zacinlo
Keyloggers	monitorizează apăsările de taste ale victimei	Olympic Vision
Bots	lansează automat atacuri concertate și concentrate asupra sistemului țintă	Echobot
Mobile Malware	conțin cod specific (Android, iOS) infectării dispozitivele mobile	Triada

Anexa 5. Tabel simplu

Pentru a ușura lecturarea lucrării de diplomă recomandăm ca tabelele (dacă nu sunt prea mari) să nu fie împărțite (sparte) pe mai multe pagini.

Tabelul A.2. Rezultatele simulării

<i>Tipul semnalului</i>	<i>Durata</i>	<i>Randamentul</i>
sinusoidal	10s	99%
dreptunghiular	12s	81%
triunghiular	15s	89%

În Tabelul A.2 este prezentat un exemplu de formatare pentru un tabel cu cap de tabel orizontal.

Anexa 6. Cod în Python

```

1  def batch_process(self, df, create_embeddings_fn, index_num,
    ↪ batch_size=100):
2      total_items = len(df)
3      num_batches = (total_items + batch_size - 1) // batch_size #
    ↪ Calculate the number of batches
4
5      print(f'Start processing {num_batches} batches for {total_items}
    ↪ total '
6          f'items on index {self.index_names[index_num]}...')
7
8      for i in range(num_batches):
9          start = i * batch_size
10         end = min((i + 1) * batch_size, total_items)
11         df_batch = df[start:end]
12
13         print(f'Processing batch {i+1} with {len(df_batch)} items')
14
15         if index_num == 3 or index_num == 4:
16             embeddings =
17                 ↪ create_embeddings_fn(df_batch['text'].tolist())
18         else:
19             embeddings =
20                 ↪ create_embeddings_fn(df_batch['text'].tolist(),
21                 ↪ index_num)
22
23         to_upload = [{
24             'id': row['ID'],
25             'values': embeddings[batch_idx],
26         } for batch_idx, (df_idx, row) in
    ↪ enumerate(df_batch.iterrows())]
27
28         response = self.indexes[index_num].upsert(vectors=to_upload,
    ↪ namespace=self.index_names[index_num])
29         print(response)

```

Listing 1. Codul funcției de inserare a embedding-urilor în baza de date

```

1  class DeepgramTranscriber:
2  def __init__(self, api_key):
3      self.api_key = api_key
4      self.deepgram = DeepgramClient(api_key)
5
6  def transcribe_prerecorded(self, preprocessed_audio_file, language):
7      try:
8          with open(preprocessed_audio_file, "rb") as file:
9              buffer_data = file.read()
10
11         payload: FileSource = {
12             "buffer": buffer_data,
13         }
14
15         options = PrerecordedOptions(

```



```

16         model="nova-2",
17         smart_format=True,
18         language=language
19     )
20
21     print("\nTranscribing audio...")
22     response =
23         ↪ self.deepgram.listen.prerecorded.v("1").transcribe_file(payload,
24         ↪ options)
25     return response.to_json(indent=4)
26
27 except Exception as e:
28     print(f"\nException: {e}")
29     return None

```

Listing 2. Trimiterea unui request catre API-ul Deepgram

```

1  def is_offensive(self, query_emb, index_num):
2  query_resp =
3      ↪ self.profanity_indexes[index_num].query(vector=query_emb,
4      namespace=INDEX_NAMES[index_num],
5      top_k=1)
6
7  if len(query_resp.matches) > 0:
8      top_score = query_resp.matches[0].score
9      if top_score > 0.9: # (Arbitrary) threshold for determining
10         ↪ if a word is offensive
11         return True
12     return False
13
14 def get_offensive_words_timestamps(self, transcription_str,
15     ↪ create_embeddings_fn, index_num):
16     transcription_json = json.loads(transcription_str)
17     words_timestamps, all_original_words =
18     ↪ self.extract_words_timestamps(transcription_json)
19
20     offensive_words_timestamps = defaultdict(list)
21     offensive_words_list = []
22
23     unique_words = list(words_timestamps.keys())
24
25     if index_num == 3 or index_num == 4:
26         word_embeddings = create_embeddings_fn(unique_words)
27     else:
28         word_embeddings = create_embeddings_fn(unique_words,
29         ↪ index_num)
30
31     print(f'\nExtracting offensive word timestamps for model
32     ↪ {index_num+1}...')
33
34     for unique_word, embedding in zip(unique_words, word_embeddings):
35         if self.is_offensive(embedding, index_num):
36             offensive_words_timestamps[unique_word] =
37             ↪ words_timestamps[unique_word]

```

```
31         offensive_words_list.append(unique_word)
32
33     original_len = len(offensive_words_list)
34     for word in offensive_words_list[:original_len]:
35         offensive_words_list.remove(word)    # Original words in list
36         ↪ are not counted again
37
38     while word in all_original_words:
39         offensive_words_list.append(word)
40         all_original_words.remove(word)
41
42     return offensive_words_timestamps, offensive_words_list
```

Listing 3. Identificarea cuvintelor malițioase

Anexa 7. Cod în Kotlin (mai lung de o pagină)

```

1  internal inner class ScheduleAdapter :
2      RecyclerView.Adapter<CalendarSessionViewHolder>(),
3      StickyHeaderHandler {
4
5      private var schedule: MutableList<ScheduleItem> = mutableListOf()
6      var data: List<SessionDataGroup> = emptyList()
7          set(value) {
8              updateSchedule(value)
9              field = value
10         }
11
12     override fun onCreateViewHolder(
13         parent: ViewGroup,
14         viewType: Int): CalendarSessionViewHolder {
15         val view = when (viewType) {
16             ScheduleItem.TYPE_SMALL ->
17                 R.layout.item_schedule_session_header_small
18             ScheduleItem.TYPE_LARGE ->
19                 R.layout.item_schedule_session_header_large
20             else -> R.layout.item_schedule_session_card
21         }
22
23         val holder = layoutInflater.inflate(view, parent, false)
24         return CalendarSessionViewHolder(holder, _favoriteNeedSync)
25     }
26     override fun getItemCount(): Int = schedule.size
27
28     override fun onBindViewHolder(
29         holder: CalendarSessionViewHolder, position: Int) {
30         holder.show(schedule[position])
31     }
32
33     override fun getItemViewType(position: Int): Int {
34         return schedule[position].type
35     }
36     private fun updateSchedule(groups: List<SessionDataGroup>) {
37         val result = mutableListOf<ScheduleItem>()
38         for (group in groups) {
39             if (group.type == 0) {
40                 result += ScheduleItem.SmallHeader(
41                     group.title, R.color.dark_grey_40
42                 )
43                 continue
44             }
45             if (group.type == 1) {
46                 result += ScheduleItem.LargeHeader(group)
47                 continue
48             }
49
50             result += ScheduleItem.LargeHeader(group)
51             result += group.sessions.map { ScheduleItem.Card(it) }
52         }

```

```
53
54         schedule = result
55     }
56     override fun getAdapterData(): MutableList<*> = schedule
57 }
```

Listing 4. *clasa ScheduleAdapter*

Anexa 8. Cod în XML (preluat din fisier)

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
5          https://maven.apache.org/xsd/maven-4.0.0.xsd">
6      <modelVersion>4.0.0</modelVersion>
7      <parent>
8          <groupId>org.springframework.boot</groupId>
9          <artifactId>spring-boot-starter-parent</artifactId>
10         <version>2.3.4.RELEASE</version>
11         <relativePath/> <!-- lookup parent from repository -->
12     </parent>
13     <groupId>local.pos.examples</groupId>
14     <artifactId>CalculatorWS</artifactId>
15     <version>0.0.1-SNAPSHOT</version>
16     <name>CalculatorWS</name>
17     <description>Simple WS: Stateless Calculator</description>
18
19     <properties>
20         <java.version>11</java.version>
21     </properties>
22
23     <dependencies>
24         <dependency>
25             <groupId>org.springframework.boot</groupId>
26             <artifactId>spring-boot-starter-web</artifactId>
27         </dependency>
28         <dependency>
29             <groupId>org.springframework.boot</groupId>
30             <artifactId>spring-boot-starter-web-services</artifactId>
31         </dependency>
32
33         <dependency>
34             <groupId>org.springframework.boot</groupId>
35             <artifactId>spring-boot-starter-test</artifactId>
36             <scope>test</scope>
37             <exclusions>
38                 <exclusion>
39                     <groupId>org.junit.vintage</groupId>
40                     <artifactId>junit-vintage-engine</artifactId>
41                 </exclusion>
42             </exclusions>
43         </dependency>
44
45         <dependency>
46             <groupId>wsdl4j</groupId>
47             <artifactId>wsdl4j</artifactId>
48         </dependency>
49
50         <dependency>
51             <groupId>javax.xml.bind</groupId>
52             <artifactId>jaxb-api</artifactId>

```

```
53     </dependency>
54
55     <dependency>
56         <groupId>javax.activation</groupId>
57         <artifactId>activation</artifactId>
58         <version>1.1</version>
59     </dependency>
60
61     <dependency>
62         <groupId>org.glassfish.jaxb</groupId>
63         <artifactId>jaxb-runtime</artifactId>
64     </dependency>
65 </dependencies>
66
67 <build>
68     <plugins>
69         <plugin>
70             <groupId>org.springframework.boot</groupId>
71             <artifactId>spring-boot-maven-plugin</artifactId>
72         </plugin>
73
74         <plugin>
75             <groupId>org.codehaus.mojo</groupId>
76             <artifactId>jaxb2-maven-plugin</artifactId>
77             <version>2.5.0</version>
78             <executions>
79                 <execution>
80                     <id>xjc</id>
81                     <goals>
82                         <goal>xjc</goal>
83                     </goals>
84                 </execution>
85             </executions>
86             <configuration>
87                 <sources>
88                     <source>
89 ${project.basedir}/src/main/resources/Calculator.xsd
90                     </source>
91                 </sources>
92             </configuration>
93         </plugin>
94     </plugins>
95 </build>
96
97 </project>
```

Listing 5. Exemplu de cod XML preluat din fișier extern