

Candy Rush Adventure

EMANUEL-CRISTIAN BARBU AUTOR-1
COLEGIUL NAȚIONAL "VASILE LUCACIU" BAIA MARE
Specializarea: Mate-Info Bilingv Engleză cls. a XII-a A
Email: emanuelbarbu2004@gmail.com
Profesori coordonatori: Filip Adela, Radu Margareta

Abstract

Nowadays, in a digital age, the task of using technology in teaching is a priority. In this context, 'Candy Rush Adventure' comes as an educational software aimed at bridging the gap between students and chemistry. Created in Unity, the game integrates an efficient dialogue system developed with the help of the Ink script language. Therefore, 'Candy Rush Adventure' is a platformer reminiscent of classic games such as 'Super Mario Bros.' and manages to help students learn chemistry in a simple and fun way.

Keywords: educational software, chemistry, interactive



Introducere

Pentru mulți elevi, există materii mai mult sau mai puțin accesibile, acestea fiind dificile din punctul de vedere al înțelegerii conceptelor de bază. Printre aceste subiecte se numără și chimia.

Din dorința de a crea o modalitate prin care elevii să se familiarizeze cu noțiuni elementare de chimie și nu numai, am decis să combin această pasiune cu informatica, creând astfel un soft educațional care va putea fi folosit drept un instrument util în cadrul orelor de specialitate.

Astfel, inspirat și din *platformere*, am realizat un proiect care are la bază energia și dorința de a crea, intitulat *Candy Rush Adventure*, joc care abordează un aspect esențial din chimie, fără de care viața nu ar fi posibilă: zaharidele.

Secțiunea preliminară

Softul educațional *Candy Rush Adventure* a fost dezvoltat cu ajutorul aplicației Unity, a limbajului de programare C# prin intermediul editorului Visual Studios și a programului INKY și a limbajului de programare asociat acesteia, INK.

Unity este un engine¹ și framework² destinat creării de jocuri, aplicații și animații destinate mediilor 2D, 2.5D și 3D. Avantajul utilizării softului Unity este caracterul intuitiv al acesteia, permițându-i

¹ Engine= o aplicație care are ca scop executarea sarcinilor fundamentale pentru alte programe; de exemplu, într-un sistem de operare, subsistem sau aplicație, un engine organizează și supraveghează execuția și rularea programelor aferente

² Framework= unelte care permite construirea unei aplicații pe baza unor structuri specifice și special dedicate, având din start funcționalități predefinite, soft reutilizabil (care se poate modifica ulterior), având ca scop ușurarea creării unei aplicații și a dezvoltării sale

utilizatorului să interacționeze cu proiectul nu numai prin intermediul codului, ci și vizual, de exemplu prin procedee de tip drag&drop. În plus, un alt avantaj este posibilitatea portării unei aplicații construite în Unity într-o gamă variată de platforme, de la sisteme Windows, Mac sau Android până la aplicații web.

Softul Unity, lansat în 2005, a fost destinat încă de la început pentru a facilita programarea de jocuri, fapt pentru care deși aplicația este codată în C++, codul aferent unui program are posibilitatea de a fi scris în limbaje de programare larg utilizate, mai exact C# sau Java. Proiectul *Candy Rush Adventure* a fost dezvoltat prin intermediul C#.

C# este un limbaj de programare high-level³, static⁴, de uz general și a fost creată drept o versiune modernă a limbajului C++, un limbaj de tip OOP⁵, fiind de asemenea un limbaj sigur din punctul de vedere al memoriei⁶. În ceea ce privește sintaxa C#, aceasta este asemănătoare cu cea din Java.

Un element important din dezvoltarea aplicației *Candy Rush Adventure* este Visual Studios, un IDE⁷ însoțit de extensia UnityVS prin care se face legătura între o secvență de cod și un element din proiectul din Unity.

În plus, în proiectarea softului educațional a fost folosit și INK, un limbaj de programare folosit pentru a concepe sisteme de dialog pentru jocuri și nu numai. Cu o sintaxă ușor de înțeles, INK este una dintre cele mai populare alegeri în rândul dezvoltatorilor de aplicații. Deși conceptele elementare sunt ușor de înțeles, INK posedă un caracter complex, oferind posibilitatea de a crea povești complicate, cu multiple fire narative, dar și efecte în funcție de alegerile utilizatorului. Pentru a putea modela dialogul s-a utilizat aplicația INKY, un IDE pentru INK, iar pentru conectarea aplicației INKY cu elementele din Unity și secvențele de cod C# a fost necesar instalarea unui pachet de compatibilitate între cele două medii de lucru.

După instalarea aplicației INKY și a modulului de legătură între aceasta și Unity, programarea în ea este facilă. Se creează un fișier INK din Unity care va fi editat ulterior din INKY și un fișier JSON atașat⁸. După finalizarea codului, dialogul va putea fi ulterior testat în Unity. În alte cazuri, acesta poate fi exportat într-un format JSON din cadrul aplicației.

La fel ca și orice limbaj de programare, INK are programe și subprograme, variabile de diferite tipuri și structuri logice. În cadrul proiectului s-au folosit subprograme pentru a prezenta teoria, structuri de tip *if* pentru a verifica condiții(de exemplu, dacă jucătorul a obținut un număr de puncte în cadrul unui test), și alegeri⁹.

³ High-Level=mediu de programare care facilitează înțelegerea codului de către oameni, folosind simboluri, litere și expresii pentru a reda

⁴ Limbaj de programare static=înaintea compilarea codului sursă acesta este tradus într-un format care poate fi procesat de către calculator, rezultând în faptul că tuturor variabilelor folosite le trebuie atribuit un tip înainte de rularea programului

⁵ Limbaj de programare OOP(Object-Oriented Programming/Programare pe bază de obiecte)=model de programare care organizează informația cu ajutorul unor structuri de date/obiecte în schimbul utilizării programării logice bazate pe funcții

⁶ Limbaj de programare sigur din punctul de vedere al memoriei(Memory Safe Language)=limbaj de programare care previne introducerea accidentală a unui bug/probleme în ceea ce privește memoria

⁷ IDE(Integrated Development Environment)=aplicație software care ajută la crearea și testarea codului într-un mod simplu și eficient

⁸ Fișier INK asociat=Aplicația INKY și limbajul de programare INK sunt folosite *strict* la crearea unei povești. Un fișier INK nu poate interacționa direct cu un script asociat unui proiect din Unity. Din acest motiv, odată cu crearea unui fișier INK, se creează automat un fișier de tip JSON(ume.json) care poate fi utilizat de către Unity și alte secvențe de cod din C#. Un aspect ce trebuie menționat este că, în timp ce modificarea codului INK aduce cu sine o schimbare *automată* a fișierului JSON, modificarea fișierului JSON nu schimbă codul INK.

⁹ Alegeri în INK=În cadrul aplicației INKY și a limbajului de programare INK se regăsesc alegeri care pot influența firul narativ, aceste alegeri fiind de multiple feluri. Alegerea de bază este *(poate fi aleasă o singură dată), iar în

Crearea aplicației

- Grafica

Întrucât Unity nu este o aplicație special destinată creării de resurse grafice, este necesară importarea unor astfel de elemente. În acest context este utilă platforma Unity Asset Store care pune la dispoziție o gamă largă de resurse, de la sprite-uri (imagini 2D care pot fi folosite ulterior pentru personajul principal, decor, fundal etc), sunete, dar și unelte și framework-uri.

După importarea texturilor, jocul poate prinde viață. Prin introducerea unui *grid* (obiect destinat cu organizarea și afișarea pe ecran a unor elemente precum fundalul, platformele, dialogul etc.), se poate lucra cu imaginile dorite. Acest *grid* are, pentru simplitate, forma unui caiet de pătrățele, fapt ce va facilita lucrul cu texturile în funcție de rezoluția dorită.

În continuare, se va crea un *TileMap* pentru fundal, respectiv pământ și alte elemente.

Pentru a evoca atmosfera jocurilor clasice, imaginile au fost puse într-o paletă după ce au fost tăiate și secționare pentru a putea folosi mai ușor anumite părți dintr-o imagine (în acest caz imaginile au fost împărțite în pătrate de 16x16 pixeli). În final, se poate decora fundalul. Procesul se va repeta pentru pământ.

Pentru a putea respecta o anumită ordine (fundalul va fi în spatele pământului, decorul în față sau în spatele jucătorului, se va utiliza din Inspector opțiunea *Layer* pentru a atribui fiecărui obiect pe parcurs o anumită prioritate la afișare. Pământul va primi un *RigidBody(Static)* pentru a permite jucătorului să se miște și să interacționeze cu lumea.

- Jucătorul

Pentru a introduce jucătorul, se va crea un nou obiect și atribui o imagine acestuia.

Pentru a da viață jucătorului și pentru a-l ajuta să interacționeze cu lumea, acesta va primi un *RigidBody(Dynamic)* - astfel, lumea rămâne nemișcată, în timp ce jucătorul se poate mișca).

Pentru a-i permite jucătorului să se deplaseze, se va crea un nou element, în acest caz un *script-PlayerMovement* - care îi va permite jucătorului să se miște.

În acest moment, jucătorul se poate mișca, dar nu sunt prezente animații pentru a sugera în ce stare se află. Astfel, se vor lua *sprite-urile* asociate personajului (acestea nu sunt întotdeauna incluse, în cadrul aplicației *Candy Rush Adventure* s-au folosit texturi și imagini deja pregătite din Unity Asset Store). Se accesează opțiunea *Animation* și i se adaugă animațiile corespunzătoare, după care, cu ajutorul interfaței *Animator*, se realizează tranzițiile între stări (repaus, alergare, săritură, cădere, moarte) alături de câteva instrucțiuni din *script-ul* de mișcare a jucătorului.

- Obstacole

Pentru a reproduce atmosfera și dinamica specifică unui platformer s-au introdus elemente care cresc dificultatea jocului.

Pentru a crea un obstacol, la fel ca și în cadrul jucătorului, i se va atribui un *texture* și i se va adăuga un *RigidBody(Static)*.

Pentru a gestiona relația și interacțiunea dintre jucător și obstacol se va crea un nou *script(PlayerDeath)*. În plus, atât jucătorului, cât și obstacolului, li se vor atribui din meniul Inspector al interfaței Unity câte un tag prin care secvența de cod va ști ce elemente pot interacționa (jucătorul va primi tag-ul 'Player', iar obstacolul tag-ul 'Death').

cadrul jocului s-au folosit alegeri de tip + ('Sticky Choice'; astfel, alegerile pot fi alese din nou și din nou, aspect util când utilizatorul dorește să revină la o parte de teorie).

Nu în ultimul rând, după crearea primului obstacol, se va crea un folder intitulat Prefab(denumirea asset-urilor care, odată create, pot fi reutilizate, având opțiunea de modifica eventual fiecare element individual după cerințele aplicației), iar obstacolul, prin drag&drop, va fi atribuit folder-ului.

- Platforme

Orice platformer are nevoie de platforme. Asemenea cazurilor anterioare, se va crea un obiect căruia îi va fi atribuit un sprite(sau animație dacă ea este existentă) și va deveni parte din folder-ul Prefab, iar platformei i se va atribui un Rigidbody(Dynamic).

Mișcarea va fi executată cu ajutorul unui script(Waypoint Follower).

Pentru a determina drumul platformelor, se pot introduce câteva puncte care determină direcția platformei în final, puncte/obiecte care sunt prelucrate la nivelul secvenței de cod menționate anterior.

- Dialogul

Dialogul funcționează cu ajutorul mai multor subprograme:

Input Manager

O metodă prin care se poate supraveghea execuția programului și al input-ului utilizatorului, oferind dezvoltatorului de a oferi și dezvolta funcționalități multiple în Unity. În cadrul proiectului *Candy Rush Adventure*, programul Input Manager este utilizat pentru a determina tastele utilizate pentru a determina direcția jucătorului, dacă acesta sare sau nu și dacă el interacționează cu elemente specifice sistemului de dialog(începerea dialogului, alegerea unei opțiuni etc.).

Dialogue Loader

Un script prin care dialogul este accesat și încărcat în cadrul aplicației. În mod normal, după un build(construirea aplicației efective și exportarea acesteia din Unity), *toate elementele acesteia* fac parte din aplicație, modificarea acestora fiind imposibilă(un element de acest fel este denumit și *embeded*). Metoda Dialogue Loader este formată din 2 pași:1) verifică dacă la nivelul calculatorului există un folder în care sunt incluse fișierele de tip JSON care constituie baza poveștii, în caz negativ, se trece la al doilea pas; 2) creează un folder separat în cadrul calculatorului în care sunt precizate și incluse fișierele de tip JSON care vor fi utilizate pe parcursul desfășurării programului.

În acest fel, script-ul Dialogue Loader contribuie la crearea unei copii permanente a datelor de tip JSON care sunt folosite. De asemenea, faptul că aceste fișiere nu sunt parte a programului constituie un avantaj și din punctul de vedere al dezvoltării aplicației. Astfel, chiar dacă aplicația a fost exportată din Unity, dialogul ei poate fi schimbat prin exportarea unui fișier INK într-unul de tip JSON, urmată de înlocuirea *conținutului* fișierului JSON inițial. Înlocuirea fișierului inițial ar duce la o corupere a datelor softului, ceea ce determină prăbușirea aplicației.

Dialogue Trigger

O secvență de cod importantă care declanșează dialogul. Odată ce jucătorul intră în proximitatea unui hitbox(zonă în care jucătorul/obiectele pot interacționa), opțiunea de a începe dialogul devine posibilă, activându-se metoda EnterDialogueMode din script-ul DialogueManager, aceasta începând odată cu apăsarea unei taste stabilite dinainte cu ajutorul InputManager-ului. Această metodă este folosită în două instanțe:1)prezentarea unor informații, teste de recapitulare a cunoștinței; 2)interacțiunea cu unele personaje care nu pot fi controlate de către utilizator, cunoscute și sub numele de NPC(Non-Playable Character). Ea funcționează pe baza faptului că jucătorul are tag-ul 'Player', iar obiectul care interacționează 'Interactable'.

Dialogue Manager

Script-ul principal din cadrul sistemului de dialog, acesta având în subordine toate celelate secvențe de cod asociate.

Mod de funcționare:

1. Programul verifică dacă povestea asociată fișierului INK a ajuns la final. În caz afirmativ, programul determină ieșirea din rutina¹⁰ ContinueStory. Altfel, se continuă povestea.
2. Povestea se continuă, apăsarea tastei atribuite dinainte(în cadrul acestei aplicații este vorba de tasta 'C')determinând trecerea la următoarea linie din fișierul INK.
3. Dacă se ajunge în cadrul unei alegeri se utilizează metoda DisplayChoices care va determina apariția pe ecran a alegerilor și posibilitatea utilizatorului de a alege una dintre opțiuni în acest caz.
4. După ce o variantă este aleasă se continuă povestea, repetându-se ciclul de instrucțiuni până la finalul fișierului INK.

Dezvoltarea proiectului continuă odată cu introducerea teoriei care va fi expusă utilizatorului și care va putea fi folosită de către cadrul didactic cu ocazia orelor de curs. Softul *Candy Rush Adventure* prezintă materia sub două forme: imagini de tip pop-up și secvențe dialogate, amândouă având nevoie de o interfață de tip *Canvas* prin care informația va fi afișată pe ecran.

Pentru crearea unei element de tip pop-up se creează, în cadrul obiectului *Canvas* din ierarhia obiectelor dintr-o scenă din Unity, un obiect de tip *Image*. În cadrul acestuia se pot edita poziția, transparența imaginii, dar se pot adăuga și instrucțiuni. Acestui obiect i se va adăuga un Collider2D prin care, cu ajutorul script-ului UIAppear, imaginea va apărea când jucătorul este în hitbox-ul realizat și va dispărea în caz contrar

Pentru crearea secvențelor dialogate, se va crea mai întâi un panou de dialog(Dialogue Panel)care va constitui fundalul dialogului. De asemenea, se creează un obiect text(Dialogue Text)unde va fi afișat textul dialogului. În acest sens, se pot modifica fontul, grosimea sau culoarea textului. Nu în ultimul rând, se va crea un buton pentru alegerea opțiunilor din cadrul dialogului(Choices), buton care va fi atribuit folder-ului Prefab pentru a putea fi reutilizat în funcție de numărul de alegeri pe care dezvoltatorul le-a introdus în script-ul INK. Interfața. Pentru ecranul de pornire se va crea o nouă scenă, iar ea va fi amplasată înaintea primelor două prin intermediul managerului de scene din interfața Unity(Scene Manager). Se vor crea un *canvas* și două obiecte de tip text: una va fi folosită pentru a menționa titlul aplicației, iar a doua pentru a detalia comenzile necesare interacțiunii cu produsul final. În plus, se va adăuga un buton denumit 'Start' la care i se va atașa un script prin care aplicația va porni. Nu în ultimul rând, va fi adăugat la nivelul acestei scene un obiect 'Loader' care va avea atașat script-ul Dialogue Loader. Astfel, aplicația va avea acces la toate fișierele JSON înainte de rularea efectivă a aplicației, evitându-se bug-uri legate de funcționarea jocului și de interacțiunea între elementele softului educațional.

Pentru ecranul de oprire se va adăuga încă o scenă din Scene Manager, alături de un *Canvas* și un buton 'Ieși din joc' care va avea o secvență de cod care va determina închiderea aplicației.

Rularea aplicației

Mai întâi, utilizatorul va fi întâmpinat cu ecranul de pornire cu butonului 'Start', dar și de o serie de instrucțiuni în ceea ce privește utilizarea aplicației. În cadrul acestui *screen* rulează prima metodă folosită, Dialogue Loader. Odată cu apăsarea butonului 'Start', utilizatorul își poate începe procesul de învățare. În timp ce explorează lumea, jucătorul va întâmpina primul NPC(în acest caz un cavaler)care îi oferă jucătorului ocazia de a-și spori cunoștințele în domeniul chimiei.

După ce jucătorul a ajuns în pădure, respectiv în tunel, acesta se va întâlni cu mai multe locații în care vor fi prezente diferite informații despre subiectul discutat, puncte semnalate prin intermediul unor indicii asemănătoare cu cel folosit în cadrul cavalerului în nivelul anterior. De asemenea, sunt prezente și imagini de tip pop-up, acestea fiind de obicei în proximitatea locațiilor unde este expusă teoria. În plus, faptul că nu sunt semnalate va contribui la evocarea unei atmosfere asemenea jocurilor clasice. Astfel, utilizatorul va avea parte de o experiență intrigantă.

¹⁰ Rutină=secvență de cod care rulează cât timp sunt îndeplinite unele condiții

Dacă jucătorul intră în imediata apropiere a unui indiciu și alege să acceseze meniul de dialog, acesta va avea posibilitatea de a aprofunda teoria și de a alege una dintre opțiunile prezentate în dreapta chenarului de informații, având posibilitatea de a sări în orice moment între o parte de teorie sau alta și chiar să iasă din interfața dialogului. De asemenea, în cadrul unor locații, sunt prezente și teste de verificare ale cunoștinței care pot evalua nivelul de fixare a materiei. Există și locații în care softul educațional are structura unui platformer, fapt ce reiese din prezența unor provocări.

Ultimul test de recapitulare reia noțiunile predate și le verifică. La final, jucătorul părăsește harta, iar ecranul de părăsire a aplicației este afișat odată cu butonul 'Ieși din joc' care, odată apăsat, închide aplicația.

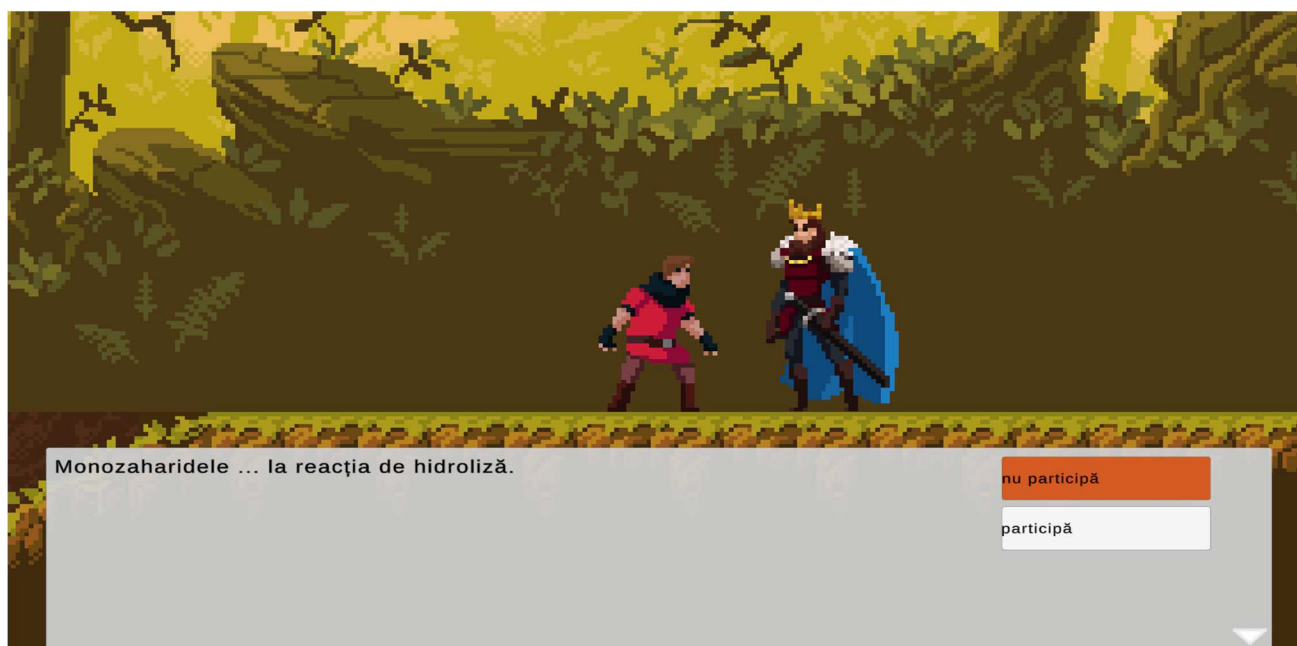


Fig. 1 Interfața din modul dialog din jocul *Candy Rush Adventure* de Barbu Emanuel

Concluzii

Candy Rush Adventure este un proiect complex. Jocul oferă o interfață simplă, interactivă și distractivă prin care elevii, indiferent de nivelul lor, își pot consolida noțiunile legate de chimie și zaharide. Asemănarea cu un platformer determină schimbarea modului în care este privită învățarea: ca pe o activitate care, deși presupune efort, poate aduce satisfacție.

Teoria este predată într-un mod unic, fiind evidențiate pe parcursul utilizării softului elemente de teorie esențiale, acestea putând fi elaborate de către un profesor la nevoie. În plus, întrebările sunt variate, ele dovedindu-se o modalitate eficientă pentru elevi de a-și testa cunoștințele, de la persoane care abia își încep drumul în chimie la altele care au aprofundat mult acest subiect, precum elevii care studiază în vederea admiterii în cadrul universităților și facultăților de medicină și nu numai.

De asemenea, crearea aplicației în softul Unity permite dezvoltarea ulterioară a acesteia în viitor, adăugându-se mai multe elemente care pot facilita procesul de învățare și experiența utilizatorului: 1) o interfață de editare a întrebărilor pentru profesori pentru folosirea jocului drept verificare a cunoștințelor în cadrul orelor de curs; 2) posibilitatea de a alege un *skin* pentru jucător; 3) o gamă variată de subiecte care pot fi abordate, alături de noi elemente de predare (de exemplu videoclipuri, prezentări PowerPoint integrate, jocuri cu o limită de timp, puzzle-uri etc.).

În concluzie, printr-un caracter simplu, interactiv și inovator, *Candy Rush Adventure* poate deveni un instrument de bază pentru oricine care vrea să învețe, înțeleagă și să se bucure de chimie.

Bibliografie

- [1] <https://learn.microsoft.com/en-us/archive/msdn-magazine/2014/august/unity-developing-your-first-game-with-unity-and-csharp>
]
- [2] <https://www.spiceworks.com/tech/devops/articles/what-is-software-engine/#:~:text=An%20engine%20in%20computer%20programming,around%20functions%20of%20other%20programs>
]
- [3] <https://www.netsolutions.com/insights/what-is-a-framework-in-programming/>
]
- [4] <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
]
- [5] <https://www.youtube.com/watch?v=ravLFzIguCM>
]
- [6] <https://www.baeldung.com/cs/statically-vs-dynamically-typed-languages#:~:text=Statically%20Typed%20Languages&text=This%20means%20that%20before%20source,Go%2C%20Pascal%2C%20and%20Swift>
]
- [7] [https://www.techtarget.com/searcharchitecture/definition/object-oriented-programming-OOP#:~:text=Object%2Doriented%20programming%20\(OOP\)%20is%20a%20computer%20programming%20model,rather%20than%20functions%20and%20logic.](https://www.techtarget.com/searcharchitecture/definition/object-oriented-programming-OOP#:~:text=Object%2Doriented%20programming%20(OOP)%20is%20a%20computer%20programming%20model,rather%20than%20functions%20and%20logic.)
]
- [8] <https://www.memorysafety.org/docs/memory-safety/>
]
- [9] [https://aws.amazon.com/what-is/ide/#:~:text=An%20integrated%20development%20environment%20\(IDE,easy%2Dto%2Duse%20application.](https://aws.amazon.com/what-is/ide/#:~:text=An%20integrated%20development%20environment%20(IDE,easy%2Dto%2Duse%20application.)
]
- [10] <https://www.youtube.com/watch?v=KSRpcftVyKg>
]