

TASTARE RAPIDĂ

# Introducere

## Ce este touchtyping?

Cu cât tehnologia avansează, aceasta devine din ce în ce mai predominantă în viețile noastre de zi cu zi. Calculatoarele și-au găsit utilitate, într-un mod sau altul, în aproape fiecare domeniu, de la medicină la jurnalism și multe altele. De aceea, a știi să folosim și navigăm în mod eficient internetul este o necesitate.

Touchtyping este o metodă de tastare care dă utilizatorului abilitatea de a scrie repede, corect, și fără a privi tastatura sau a căuta individual fiecare tastă.

În loc să își îndrepte privirea spre tastatură, utilizatorul ține mâinile pe poziția de bază (de obicei, cu degetele de mijloc pe tastele F și J, care au marcaje tactuale pentru ghidare) și apasă tastele cu degetele corespunzătoare (fiecare deget are atribuit câteva taste), având încredere în memorie musculară pentru a localiza literele și simbolurile.

Această tehnică asigură eficiență maximă a tastării, întrucât este folosit fiecare deget al mâinii. Cu destul antrenament se poate ajunge la viteze de chiar 100 – 150 WPM. (Words per Minute)

## Ce utilitate are jocul Fox Racing?

Până acum, singură metodă de a învăța metoda touchtyping era prin diferite site-uri educative, constând din multe exerciții de tastare. Acestea, într-adevăr, sunt necesare pentru a putea învăța touchtyping, dar ele pot fi foarte plictisitoare și greu de parcurs.

Jocul **Fox Racing** își propune să facă procesul de creștere a vitezei de scriere, folosind această metodă, unul mai plăcut și mai accesibil pentru publicul larg. Cu toate că este potrivit pentru toate vârstele, este în special destinat elevilor tineri, întrucât învățarea acestei tehnici de la o vârstă fragedă poate aduce numeroase beneficii în viitor.

Astfel, scopul general este de a oferi utilizatorilor un intermediu nou, unic și distractiv, sub forma unui joc video de tip Racing Game, prin care aceștia să își poată îmbunătăți viteza de scriere.



Figure 1. Ecranul de start

# Structura și modul de utilizare

În **Fox Racing** utilizatorul joacă rolul unei vulpi care se întrece cu șase alte vietăți ale naturii, pentru a deveni cea mai sprintenă vulpe a regatului animal.

**Fox Racing** are trei moduri esențiale de joc:

1. RACING
2. FREE ROAM
3. LESSONS



Figure 2. Meniul principal

Putem de asemenea observa, pe ecran, un contor “Max WPM” care ne indică viteza de scriere maximă atinsă până acum în timpul jocului. Acest detaliu dă utilizatorului o idee legată de capabilitățile lui.

Trecând mouse-ul peste fiecare buton din stânga, putem schimba modul de joc.

O caracteristică importantă este salvarea automată. Astfel, după fiecare cursă sau lecție completată și orice altceva, jocul se salvează automat, iar progresul nu este pierdut după închiderea jocului.

## RACING

În acest mod de joc, utilizatorul se întrece umăr la umăr cu fiecare dintre cei șase adversari, *Loki the Wolf* fiind cel mai rapid.

Utilizatorul controlează viteza vulpii prin viteza (WPM) la care tastează cuvintele date pe ecran. Fiecare cursă ia loc prin pădure, pe un traseu prestabilit.

Se poate observa în partea de sus o bară de progres, în stânga un “vitezometru” care ne indică viteza vulpii, iar în partea de jos a ecranului bara ce sugerează cuvintele pe care trebuie să le tastăm.



Figure 3. În timpul cursei



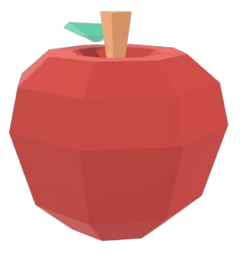
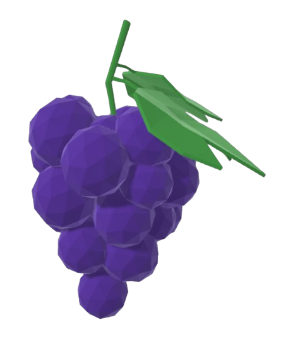
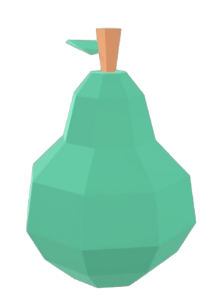
Figure 4. You Won

Figure 5. You Lost

## FREE ROAM



Figure 6. Free Roam

În acest mod de joc, utilizatorul se poate plimba liber prin pădure, culegând fructe pe drumul său.



Pentru a începe fiecare cursă este nevoie de un număr minim de fructe adunate în modul **Free Roam**. Astfel, utilizatorul are șansa de a își exersa tastarea, fără a avea presiune de timp sau viteză, navigând întreaga pădure.

Folosind tasta (Down Arrow) jucătorul poate schimba direcția de mers a vulpii.

De asemenea, la apropierea unei intersecții, pe ecran apar două săgeți roșii (exemplu în figura 6), indicând că utilizatorul poate folosi tastele săgeată stânga și săgeată dreapta pentru a selecta direcția de mers dorită.

## LESSONS

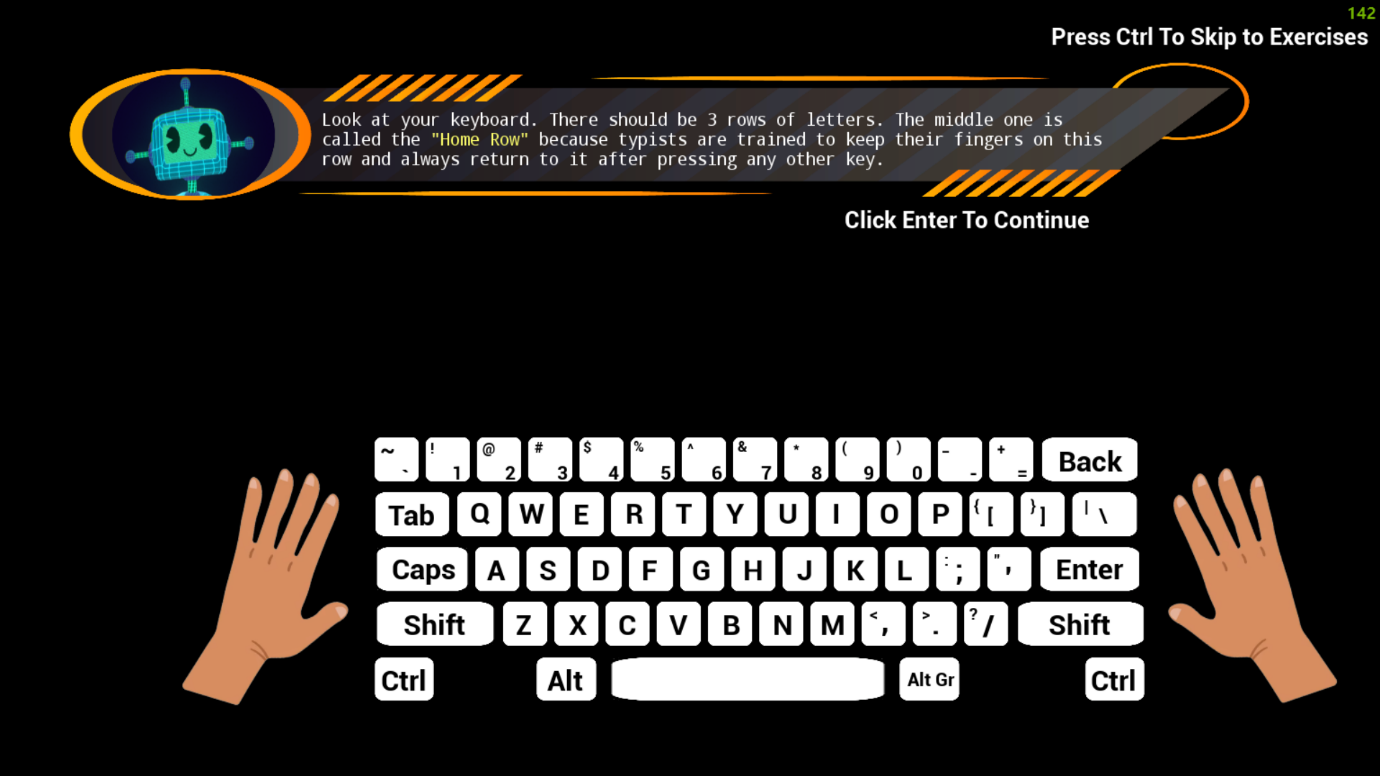


Figure 7. Lessons - explicație

În acest mod, utilizatorul poate învăța metoda touchtyping. Există 13 lecții, fiecare vorbind despre alt grup de litere. La sfârșitul celor 13 lecții utilizatorul va știi sa tasteze rapid, fără a-și lua privirea de la ecran.

La începutul lecției se explică ce degete se folosesc pentru a tasta tastele respective. (figura 7)

*Exemplu: To press W & O you use the ring fingers from your left and right hands respectively, and then return to the Home Row.*

Figure 8. Lessons - Exerciții

În continuare, sunt date câteva exerciții care să pună în aplicare ce tocmai am învățat. Utilizatorul trebuie să facă mai puțin de 20 de greșeli pentru a completa lecția. Atunci când este tastată greșit o literă, aceasta poate fi ștearsă. (figura 8)

La fiecare literă este evidențiat pe ecran poziția acesteia pe tastatură, cât si degetul cu care trebuie apăsată aceasta. Astfel, utilizatorul se obișnuiește să nu își ia privirea de la ecran atunci când scrie. (figura 8)

Cuvintele care trebuie tastate pentru a te deplasa în modurile **Free Roam** și **Racing** vor folosi numai tastele învățate până acum în modul **Lessons**.

Fiecare cursă necesită un număr de lecții completate înainte de a începe cursa. Astfel, utilizatorul are motivația de continua să învețe metoda touchtyping.

# Detalii tehnice

## 1. Descriere generală și Game Engine

Pentru a crea Fox Racing, am ales Unreal Engine 5 (UE5), deoarece acest Game Engine este gratis, cunoscut, și oferă multe posibilități.

Unreal Engine 5 este, la bază, construit în limbajul C++ (programare orientată pe obiecte), iar multe dintre funcționalitățile mai avansate ale jocului sunt scrise în C++, folosind diferite clase puse la dispoziție de UE5.

UE5 mai oferă, de asemenea, modul *“Blueprints”.* *Blueprints* sunt o modalitate grafică de a crea logică și funcționalitate pentru jocurile tale fără a fi nevoie să scrii cod în limbajul de programare C++. Sunt ca un set de piese LEGO pe care le poți asambla pentru a construi tot felul de lucruri în jocul tău. C++ și Blueprints pot fi folosite împreună și pot interacționa, pentru a face crearea jocului cât mai ușoară.

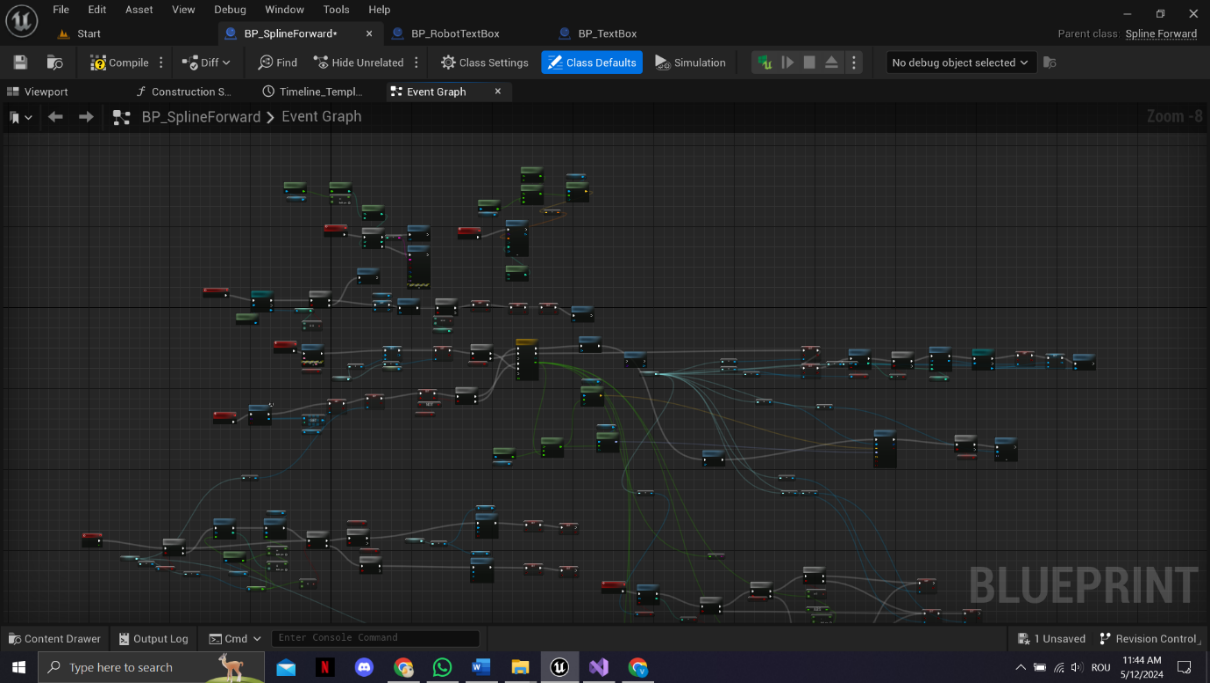


Figure 9. Blueprints (implementare SplineForward - FreeRoam)

Figure 10. Exemplu Blueprint (implementare traseu pentru modul Free Roam)

## 2. Secvențe de cod și funcționalități complicate

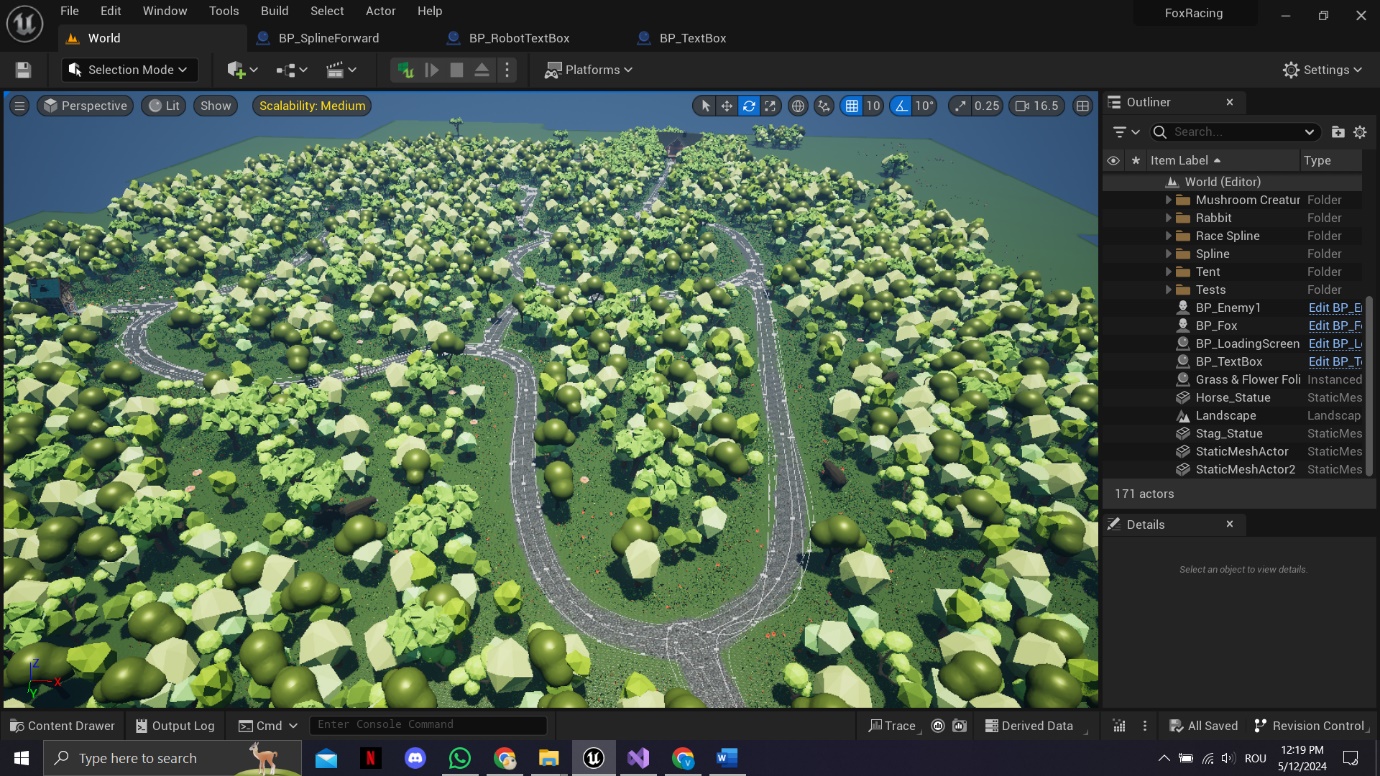
* Traseele predefinite în modurile Racing și Free Roam

Figure 10. Splines (liniile albe)

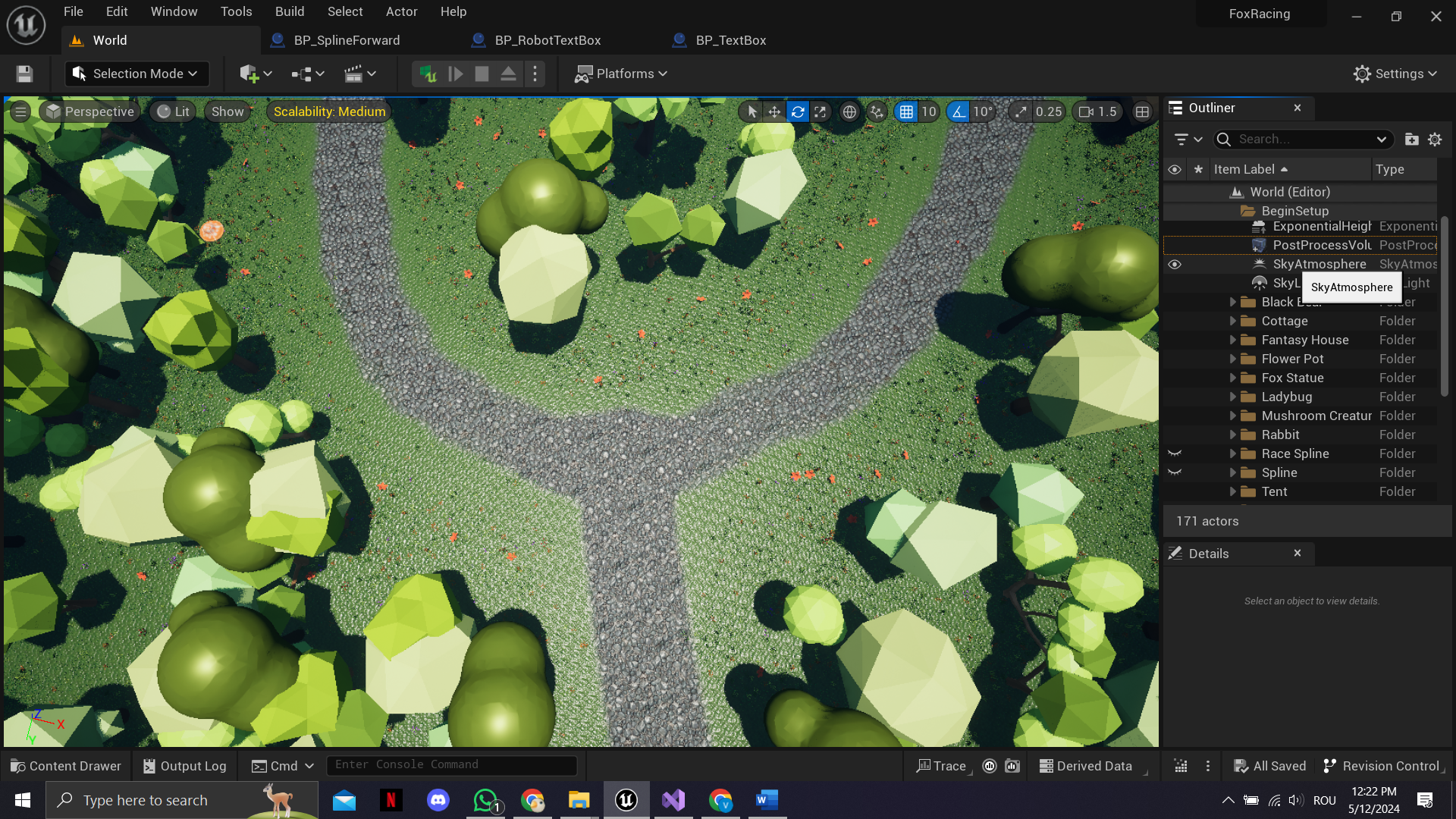
Întrucât drumul caracterului este predefinit (utilizatorul controlează numai viteza sau direcția) a fost nevoie de trasarea unor drumuri prin pădure, pe care vulpea să le urmeze.

Acest lucru poate fi îndeplinit folosind Splines.

În UE5, Splines sunt ca niște linii curbate sau trasee imaginare pe care le poți crea și manipula pentru a controla mișcarea sau comportamentul diferitelor obiecte în jocul tău. Un Spline e ca o bandă flexibilă sau o cale pe care o poți contura și îndoi în orice direcție dorești. (figura 10 – Splines folosite în modul Racing și Free Roam)

Pentru modul Racing, utilizarea obișnuită a unui Spline e destul, întrucât, după ce s-a ajuns la sfârșitul unui Spline se termină cursa.

Totuși, nu există capabilitatea de “bifurca” un Spline – adică de a crea intersecții. Această problemă a necesitat multe idei pe care le voi prezenta în continuare.



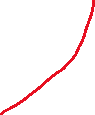
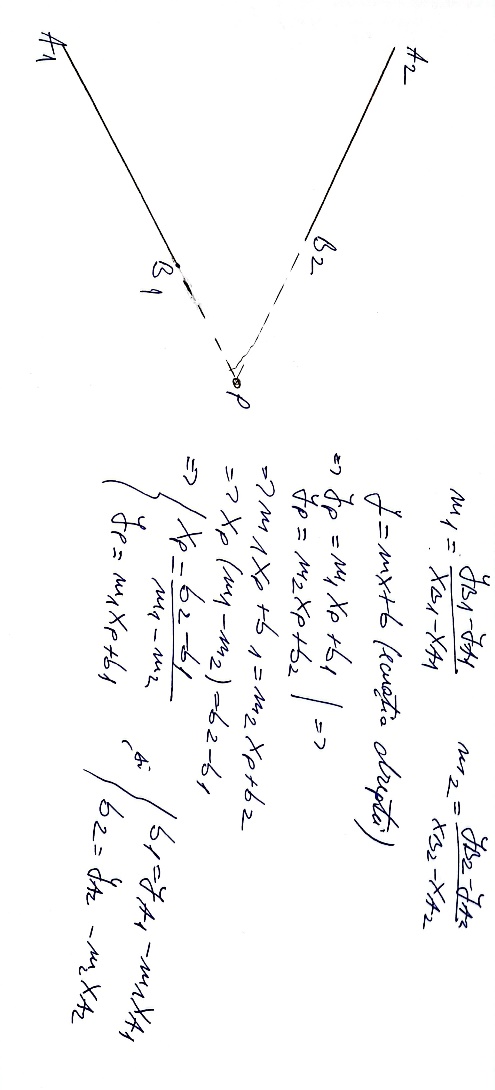


Figure 11. SplineForward și SplineTurn

Am creat 2 tipuri de Spline: SplineForward și SplineTurn.

SplineForward reprezintă un drum fără intersecții, un drum continuu, cel marcat cu roșu în figura 11.

În momentul în care 3 drumuri SplineForward se întâlnesc, este nevoie de drumuri SplineTurn (marcate cu albastru în figura 11), care creează legătura dintre fiecare SplineForward. Când caracterul ajunge la finalul unui SplineForward, acesta se mută pe un SplineTurn, și după la următorul SplineForward, în funcție de direcția de mers aleasă.

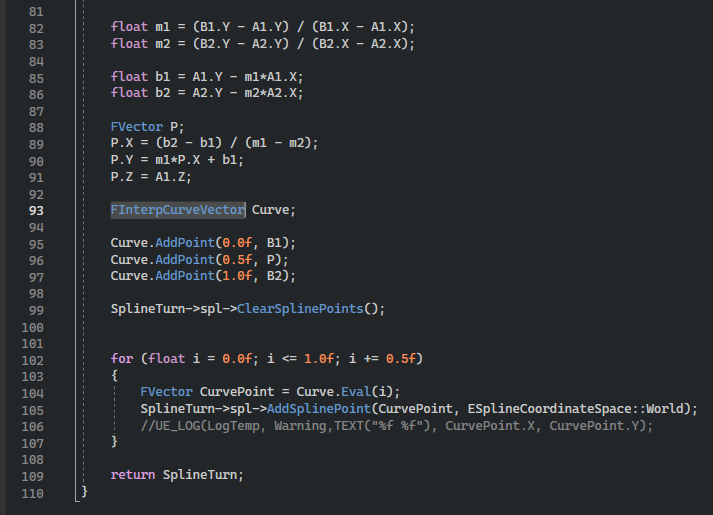
Traseele SplineForward sunt puse manual pe mapă, dar cele SplineTurn sunt generate în funcție de intersecția a celor două SplineForward pe care le unește. Mai întai, aflăm folosind calcule cu pante, punctul P(x, y) de intersecție a celor 2 SplineForward. (figura 12)

Figure 12. Calcule intersecție

După care, folosim un tip de date specific Unreal Engine 5 numit *FinterpCurveVector* și creăm variabila “Curve”*.*

Cu *FInterpCurveVector*, poți defini un set de puncte de control (precum locația sau scalarea unui obiect în diferite momente de timp) și apoi să interpolezi valorile între aceste puncte într-un mod fluid. Astfel poți crea animații sau efecte de tranziție lină între diferite stări ale obiectului sau ale jocului tău.

În alte cuvinte, adăugam punctele B1, P și B2 (din figura 12), iar acest tip de date va crea o mulțime de puncte care să traseze curba formată din punctele B1, P și B2. SplineTurn este format din această mulțime de puncte.



De asemenea, orice Spline are un Start și un End fixate, adică sensul de mers trebuie specificat, ceea ce adaugă la dificultatea implementării.

Un alt detaliu este faptul că la un Spline trebuie precizat timpul în care se va parcurge acel Spline (pentru 2 Splines care vor fi parcurse în 2 secunde, pe Spline-ul mai lung viteza va fi mai mare), deci pentru a avea o viteză constantă care să depindă de WPM trebuie să setăm Play Rate-ul la WPM / SplineLength. (play rate se referă la viteza de redare a unei animații sau a unei secvențe de acțiuni în joc)

* Comportamentul adversarului în curse

Comportamentul adversarului în curse este un detaliu important pentru a controla dificultatea fiecărei curse și pentru a face cursa competitivă.

De aceea, am optat pentru un algoritm de tip *Rubberbanding* (întalnit adesea în jocurile video de tip Racing), care variază viteza adversarului în funcție de progresul jucătorului.

În esență, rubberbanding-ul funcționează astfel:

* dacă jucătorul se află în spatele adversarului în cursă, jocul îi va oferi un impuls artificial pentru a-i permite să recupereze sau să ajungă mai aproape de lider. Acest impuls, în cazul jocului Fox Racing, constă din încetinirea adversarului;
* pe de altă parte, dacă un jucător se află în frunte, sistemul de rubberbanding va crește viteza adversarului considerabil, cu intenția de a deveni o amenințare pentru jucător (totuși, fiecare adversar are o viteză maximă prestabilită, conform dificultății fiecărei curse);

Astfel, sistemul Rubberbanding este conceput pentru a ține concurenții apropiați, pentru “a înfășura o bandă de cauciuc imaginară” în jurul lor. El încearcă să mențină un nivel optim de provocare pentru fiecare jucător prin ajustarea variabilelor din joc în timp real.

* Viteza caracterului și calcularea WPM

Viteza caracterului principal, vulpea, se schimbă în timp real în funcție de WPM. (Words per Minute)

Pentru a calcula cat mai corect WPM, în mod uzual, se înregistrează numărul de cuvinte scrise timp de un minut, dar în acest caz avem nevoie de un WPM dinamic, instant, de la începutul tastării, care să se actualizeze în funcție de utilizator.

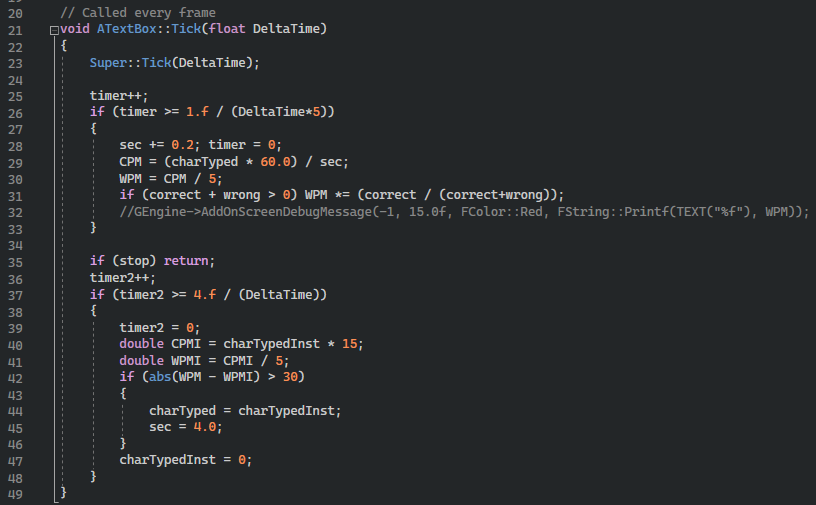
Astfel, se calculează, mai întai, CPM (Characters per Minute), *CPM = (charTyped \* 60.0) / sec*, unde *charTyped* e numărul de caractere tastate până la momentul curent iar *sec* este numărul de secunde trecute. Întrucât un cuvânt are, în medie, 5 caractere, WPM = CPM / 5.

De asemenea, pentru a lua în considerare și greșelile de până acum, înmulțim WPM cu *(correct / (correct+wrong))*, unde *correct* și *wrong* sunt numărul de caractere tastate corect, respectiv greșit de până acum.

Totuși, apare o problemă: WPM-ul scade odată cu viteza de scriere, dar nu la același ritm. În alte cuvinte, dacă oprim tastarea, caracterul nostru se va opri de abia după 10-15 secunde (ca și cum ar avea prea multă inerție).

Pentru a rezolva această problemă, am implementat un WPMI (WPM “instantaneu”), care reprezinta WPM-ul exclusiv pentru ultimele patru secunde. Astfel, dacă se detectează o diferență prea mare (> 30) între WPM si WPMI, înseamnă ca a avut loc o schimbare bruscă, iar viteza caracterului va fi actualizată corespunzător.

Verificarea si actualizarea WPM-ului se întâmplă la fiecare Tick (adică la fiecare dintre cele 60 de frame-uri pe secundă).



# Cerințe tehnice

System requirements:

OS: Windows 10 64-bit or higher

Processor: Quad-core Intel or AMD, 2.5 ghz

Memory: 16 GB RAM

Graphics Card: 4 GB VRAM, DirectX 11 or 12 compatible

# Bibliografie

[Sketchfab - 3d meshes](https://sketchfab.com/stefan.vasilache2005/collections/touch-typing-game-b4ae26b1d26e424cb985f1ef1d86977c)

[Poly.pizza - 3d meshes](https://poly.pizza/l/3mPBTHtYDq)

[opengameart.org - Low Poly Nature Pack 3d meshes](https://opengameart.org/content/lowpoly-nature-pack)

[Design - Logo](https://www.design.com/maker/logo/gear-shift-fox-67551?text=Fox+Racing&isSearch=True)

[nectanebo.itch - Menu Buttons](https://nectanebo.itch.io/menu-buttons)

[stackoverflow - scop informativ](https://stackoverflow.com/)

[UE5 Forums](https://forums.unrealengine.com/)

[UE5 Documentation](https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-5-4-documentation)

[Fonts](https://www.1001fonts.com/)

Contents

[Introducere 2](#_Toc166450574)

[1. Ce este touchtyping? 2](#_Toc166450575)

[2. Ce utilitate are jocul Fox Racing? 3](#_Toc166450576)

[Structura și modul de utilizare 4](#_Toc166450577)

[1. RACING 5](#_Toc166450578)

[2. FREE ROAM 6](#_Toc166450579)

[3. LESSONS 7](#_Toc166450580)

[Detalii tehnice 8](#_Toc166450581)

[1. Descriere generală și Game Engine 8](#_Toc166450582)

[2. Secvențe de cod și funcționalități complicate 9](#_Toc166450583)

[ Traseele predefinite în modurile Racing și Free Roam 9](#_Toc166450584)

[ Comportamentul adversarului în curse 12](#_Toc166450585)

[ Viteza caracterului și calcularea WPM 12](#_Toc166450586)

[Cerințe tehnice 14](#_Toc166450587)

[Bibliografie 14](#_Toc166450588)