**PROIECT PENTRU OBTINEREA ATESTARII PROFESIONALE IN INFORMATICA**

**Titlul lucrarii:**

ClimateSpy

Bacau

Mai, 2023

Cuprins

1.Introducere

2.Generalitati despre Godot Game Engine si GDScript

3.Cerinte de sistem

4.Structura si continutul proiectului

5.Bibliografie

**Introducere**

Lucrarea de fata reprezinta o aplicatie care arata vremea in momentul curent, pentru coordonatele introduse in aplicatie, scris in GDScript.

In mod implicit,aplicatia arata vremea pentru judetul Bacau, dar asta se poate schimba in casutele „Latitudine” si „Longitudine”.

In cazul unei erori, aplicatia poate fi repornita cu ajutorul butonului de refresh, care va reincarca programul, dar de asemenea va si reseta valorile pentru latitudine si longitudine, la valorile implicite.

A screen shot of a computer

Description automatically generated with low confidence

**Generalitati**

Godot Engine este un motor de joc pentru multiple platforme, plin de caracteristici, pentru a crea jocuri 2D și 3D dintr-o interfață unificată. Oferă un set cuprinzător de instrumente comune, astfel încât utilizatorii să se poată concentra pe realizarea jocurilor fără a fi nevoie să reinventeze roata. Jocurile pot fi exportate cu un singur clic către o serie de platforme, inclusiv principalele platforme desktop ( Linux, macOS, Windows ), platforme mobile ( Android, iOS ), precum și platforme și console bazate pe Web.

Godot este complet gratuit și open source sub licența MIT permisivă. Jocurile utilizatorilor sunt ale lor, până la ultima linie de cod a motorului. Dezvoltarea Godot este pe deplin independentă și orientată către comunitate, împuternicind utilizatorii să ajute la modelarea motorului lor pentru a se potrivi cu așteptările lor. Este susținut de Fundația Godot fără scop lucrativ.

In mod implicit este folosit limbajul de programare GDScript, se pot utiliza si C# si C++. Cu ajutorul contribuitorilor, se pot folosi mult mai multe limbaje de programare precum: Rust, JavaScript, Lua, Nim, dar cu suport neoficial, depinde numai de contribuitori pentru compatibilitatea acestor limbaje de programare cu motorul de jocuri.

A screenshot of a computer

Description automatically generated with medium confidence

În Godot, îți descompui jocul în scene reutilizabile. O scenă poate fi un personaj, o armă, un meniu în interfața de utilizator, o singură casă, un nivel întreg sau orice vă puteți gândi. Scenele lui Godot sunt flexibile; ele completează atât rolul prefabricatelor, cât și al scenelor în alte motoare de joc.

A screenshot of a video game

Description automatically generated

Puteți cuiba și scene. De exemplu, puteți pune personajul dvs. la un nivel și puteți trage și arunca o scenă ca un copil al acesteia.

A screenshot of a computer screen

Description automatically generated with low confidence

O scenă este compusă dintr-unul sau mai multe noduri. Nodurile sunt cele mai mici blocuri de construcție ale jocului pe care le aranjați în copaci. Iată un exemplu de noduri ale unui personaj.

A screenshot of a computer

Description automatically generated

Este format dintr-un nod CharacterBody2D numit „Character”, un Sprite2D, un Camera2D și un CollisionShape2D.

A screenshot of a computer program

Description automatically generated with medium confidence

Godot oferă o bibliotecă extinsă de tipuri de noduri de bază pe care le puteți combina și extinde pentru a construi altele mai puternice. 2D, 3D sau interfață de utilizator, veți face cele mai multe lucruri cu aceste noduri.

A screenshot of a computer

Description automatically generated with medium confidenceToate scenele jocului se reunesc în arborele scenei, literalmente un copac de scene. Și cum scenele sunt copaci de noduri, arborele de scenă este și un copac de noduri. Dar este mai ușor să te gândești la jocul în ceea ce privește scenele, deoarece acestea pot reprezenta personaje, arme, uși sau interfața de utilizator.

A screenshot of a computer program

Description automatically generated with medium confidenceNodurile emit semnale atunci când are loc un eveniment. Această caracteristică vă permite să faceți ca nodurile să comunice fără a le conecta cu coduri. Oferă multă flexibilitate în modul în care vă structurați scenele.

GDScript este un limbaj de programare la nivel înalt, orientat pe obiecte, imperativ și tastat treptat, construit pentru Godot. Utilizează o sintaxă bazată pe indentare similară cu limbile precum Python. Scopul său este de a fi optimizat și integrat strâns cu Godot Engine, permițând o mare flexibilitate pentru crearea și integrarea conținutului.

GDScript este complet independent de Python și nu se bazează pe acesta.

GDScript este limbajul de script integrat al lui Godot. Acesta a fost construit de la sol pentru a maximiza potențialul lui Godot în cea mai mică cantitate de cod, oferind atât dezvoltatorilor novici cât și experți să valorifice cât mai repede punctele forte ale lui Godot. Dacă ai mai scris ceva într-o limbă precum Python înainte, atunci te vei simți drept acasă. Pentru exemple și o imagine de ansamblu completă a puterii pe care vi le oferă GDScript, consultați ghidul de scripturi GDScript.

Există mai multe motive pentru a utiliza GDScript, mai ales atunci când prototipați, în etapele alfa / beta ale proiectului dvs. sau nu creați următorul titlu AAA. Motivul cel mai important este reducerea generală a complexității.

Intenția inițială de a crea un limbaj de script personalizat, integrat, pentru Godot, a fost de două ori: în primul rând, reduce timpul necesar pentru a se ridica și a alerga cu Godot, oferind dezvoltatorilor o modalitate rapidă de a se expune motorului cu accent pe productivitate; în al doilea rând, reduce sarcina generală a întreținerii, atenuează dimensionalitatea problemelor, și permite dezvoltatorilor motorului să se concentreze pe erorile de squashing și îmbunătățirea funcțiilor legate de miezul motorului, în loc să petreci mult timp încercând să obții un set mic de funcții incrementale care funcționează într-un set mare de limbi.

Întrucât Godot este un proiect open source, a fost imperativ din start să acordăm prioritate unei experiențe mai integrate și mai perfecte pentru atragerea de utilizatori suplimentari prin sprijinirea limbajelor de programare mai familiare, mai ales atunci când sprijinirea acestor limbi mai familiare ar duce la o experiență mai proastă. Înțelegem dacă preferați să utilizați o altă limbă în Godot.

O prezentare a sintaxei GDScript

# Everything after "#" is a comment.

# A file is a class!

# (optional) icon to show in the editor dialogs:

@icon("res://path/to/optional/icon.svg")

# (optional) class definition:

class\_name MyClass

# Inheritance:

extends BaseClass

# Member variables.

var a = 5

var s = "Hello"

var arr = [1, 2, 3]

var dict = {"key": "value", 2: 3}

var other\_dict = {key = "value", other\_key = 2}

var typed\_var: int

var inferred\_type := "String"

# Constants.

const ANSWER = 42

const THE\_NAME = "Charly"

# Enums.

enum {UNIT\_NEUTRAL, UNIT\_ENEMY, UNIT\_ALLY}

enum Named {THING\_1, THING\_2, ANOTHER\_THING = -1}

# Built-in vector types.

var v2 = Vector2(1, 2)

var v3 = Vector3(1, 2, 3)

# Functions.

func some\_function(param1, param2, param3):

const local\_const = 5

if param1 < local\_const:

print(param1)

elif param2 > 5:

print(param2)

else:

print("Fail!")

for i in range(20):

print(i)

while param2 != 0:

param2 -= 1

match param3:

3:

print("param3 is 3!")

\_:

print("param3 is not 3!")

var local\_var = param1 + 3

return local\_var

# Functions override functions with the same name on the base/super class.

# If you still want to call them, use "super":

func something(p1, p2):

super(p1, p2)

# It's also possible to call another function in the super class:

func other\_something(p1, p2):

super.something(p1, p2)

# Inner class

class Something:

var a = 10

# Constructor

func \_init():

print("Constructed!")

var lv = Something.new()

print(lv.a)

**Cerințe de sistem**

Folosind la baza OpenGL3 ca API grafic, fără efecte speciale, programul nu necesita foarte multe resurse, așadar cerințele de sistem sunt următoarele:

* Sistem de operare win8
* 70MB stocare
* 70MB RAM liberi

Precizez ca doar binara (fișierul cu extensia .exe) necesita minim Windows 8, programul poate fi construit din sursa pentru a rula pe Linux si Android.

**Structura si conținut proiect**

Programul a fost realizat in întregime folosind limbajul de programare GDScript, integrat in motorul de jocuri Godot (versiunea 4.0.2), care este capabil de a realiza si simple aplicații.

A screen shot of a computer

Description automatically generated with low confidence

Primul nod încărcat este rădăcina acestuia, care are ca fiu Node2D,caruia ii este atașat un script. Acest script se va ocupa de mare parte a logicii aplicației.

Mai avem si butonul de refresh căruia ii este atașat un script, el pur si simplu reîncarcă scena după ce este apăsat:

extends Button

func \_on\_pressed():

get\_tree().reload\_current\_scene()

**Codul principal al programului:**

extends Node2D

#Initializarea variabilelor

var meteo

var cod

var vreme

#Implicit se va initializa cu coordonatele judetului Bacau

var long="26.916025"

var lat="46.568825"

var url="https://api.open-meteo.com/v1/forecast?latitude="+lat+"&longitude="+long+"&hourly=precipitation\_probability,weathercode,is\_day&daily=temperature\_2m\_max,temperature\_2m\_min&current\_weather=true&forecast\_days=1&timezone=auto"

func \_ready():

#Rulare cand programul este incarcat

cerere()

await $Network/HTTPRequest.request\_completed

codul\_vremii()

fundal()

func cerere():

#Cerere HTTP catre Open Meteo pentru a obtine prognoza meteo

#Descarca datele in variabila meteo

$Network/HTTPRequest.request\_completed.connect(\_on\_request\_completed)

$Network/HTTPRequest.request(url)

await $Network/HTTPRequest.request\_completed

func fundal():

#Verifica daca este zi sau noapte

#Schimba fundalul si culoarea textului in functie de codul vremii si ora

if(meteo["current\_weather"]["is\_day"]==1):

$Text/StareaVremii.set("theme\_override\_colors/font\_color",Color(0,0,0))

$Text/Temperatura.set("theme\_override\_colors/font\_color",Color(0,0,0))

$Reincarcare.icon=ResourceLoader.load("res://Images/darkrefresh.png")

match (vreme):

0:

$Afisare/Fundal.texture=ResourceLoader.load("res://Images/Decent\_Day.png")

$Afisare/luminaSAUintuneric.texture=ResourceLoader.load("res://Images/Sunny.png")

1:

$Afisare/Fundal.texture=ResourceLoader.load("res://Images/Cloudy\_Day.png")

2:

$Afisare/Fundal.texture=ResourceLoader.load("res://Images/Rainy\_Day.png")

3:

$Afisare/Fundal.texture=ResourceLoader.load("res://Images/Snowy\_Day.png")

4:

$Afisare/Fundal.texture=ResourceLoader.load("res://Images/Stormy\_Day.png")

$Reincarcare.show()

else:

$Text/StareaVremii.set("theme\_override\_colors/font\_color",Color(255,255,255))

$Text/Temperatura.set("theme\_override\_colors/font\_color",Color(255,255,255))

$Reincarcare.icon=ResourceLoader.load("res://Images/lightrefresh.png")

match (vreme):

0:

$Afisare/Fundal.texture=ResourceLoader.load("res://Images/Decent\_Night.png")

$Afisare/luminaSAUintuneric.texture=ResourceLoader.load("res://Images/Moon.png")

1:

$Afisare/Fundal.texture=ResourceLoader.load("res://Images/Cloudy\_Night.png")

2:

$Afisare/Fundal.texture=ResourceLoader.load("res://Images/Rainy\_Night.png")

3:

$Afisare/Fundal.texture=ResourceLoader.load("res://Images/Snowy\_Night.png")

4:

$Afisare/Fundal.texture=ResourceLoader.load("res://Images/Stormy\_Night.png")

$Reincarcare.show()

$Text/Temperatura.text=str(meteo["current\_weather"]["temperature"])+"°C"

func \_on\_request\_completed(\_result, \_response\_code, \_headers, body):

meteo = JSON.parse\_string(body.get\_string\_from\_utf8())

func codul\_vremii():

#Procesare cod vreme

if(meteo):

cod=int(meteo["current\_weather"]["weathercode"])

match (cod):

0,1,2,3:

$Text/StareaVremii.text="Senin"

vreme=0

45,48:

$Text/StareaVremii.text="Ceata"

vreme=1

51,53,55,56,57:

$Text/StareaVremii.text="Burnita"

vreme=1

61,63,65,66,67:

$Text/StareaVremii.text="Ploaie usoara"

vreme=2

71,73,75,77,85,86:

$Text/StareaVremii.text="Ninsoare"

vreme=3

80,81,82:

$Text/StareaVremii.text="Averse de ploaie"

vreme=2

95,96,97:

$Text/StareaVremii.text="Furtuna"

vreme=4

#Schimbare locatie pentru care sa afiseze vremea curenta pentru locatia respectiva

func \_on\_lat\_text\_submitted(\_test):

lat=$Lat.text

url="https://api.open-meteo.com/v1/forecast?latitude="+lat+"&longitude="+long+"&hourly=precipitation\_probability,weathercode,is\_day&daily=temperature\_2m\_max,temperature\_2m\_min&current\_weather=true&forecast\_days=1&timezone=auto"

cerere()

await $Network/HTTPRequest.request\_completed

codul\_vremii()

fundal()

func \_on\_lon\_text\_submitted(\_test):

long=$Lon.text

url="https://api.open-meteo.com/v1/forecast?latitude="+lat+"&longitude="+long+"&hourly=precipitation\_probability,weathercode,is\_day&daily=temperature\_2m\_max,temperature\_2m\_min&current\_weather=true&forecast\_days=1&timezone=auto"

cerere()

await $Network/HTTPRequest.request\_completed

codul\_vremii()

fundal()

Nodurile sunt aranjate in acest mod:

A screenshot of a computer

Description automatically generatedNode are ca scop doar de a fi rădăcina. Node2D are ca scop rularea logicii principale a jocului. Afisare are 2 fii, Fundal si luminaSAUintuneric, au ca scop afișarea fundalului in funcție de vreme iar luminaSAUintuneric sa arate soarele sau luna, in funcție de ora si condițiile meteo.

Network are ca fiu HTTPRequest, care se ocupa de cererea HTTP către Open Meteo, pentru a obține prognoza meteo.

Text are ca fii nodurile de tip Labe, texte ce urmează a fi afișate pe ecran. Culorile lor se vor schimba in funcție de ora si prin intermediul lor vom afișa temperatura si vremea curenta.

Nodurile Lat si Lon sunt de tipul LineEdit, folosite pentru a primi sau edita text. In acest caz ne folosim de aceste noduri pentru a introduce latitudinea si longitudinea locului căruia vrem sa primim vremea curenta.

Imaginile de fundal au fost create folosind programul Krita.

**Bibliografie:**

[Godot Docs – 4.0 branch](https://docs.godotengine.org/en/stable/index.html)