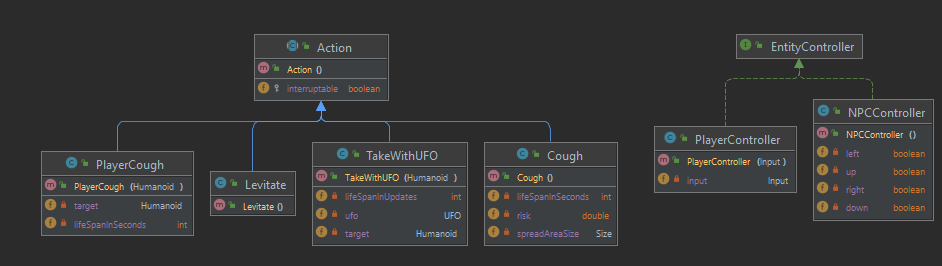
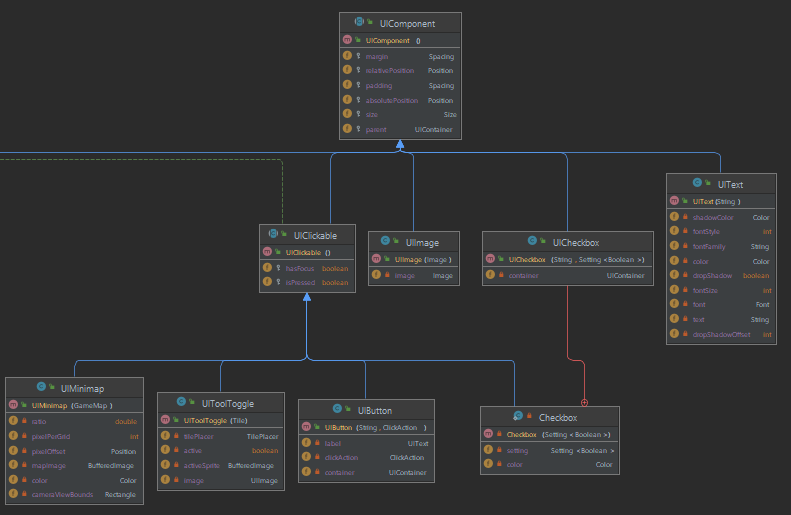


In diagrama de mai sus sunt reprezentate clasele ce sunt GameObject. Toate implementeaza GameObject , clasa de baza a oricarei enitatei etc. De aici avem mosteneste direct MovingEntity si Selection Circle. MovingEntity reprezinta clasa din care orice obiect ce se poate misca , are o directive , un motion etc. si un controller Pentru a fi miscat. Humanoid este un MovingEntity dar care are la randul lui o lista de efecte si actiuni (aceasta clasa reprezinta bazele jucatorului : Player si al NPC-urilor). Player si NPC sunt clase abstarcte pentru a se putea folosi sabloul de factory , cu clasele concrete .

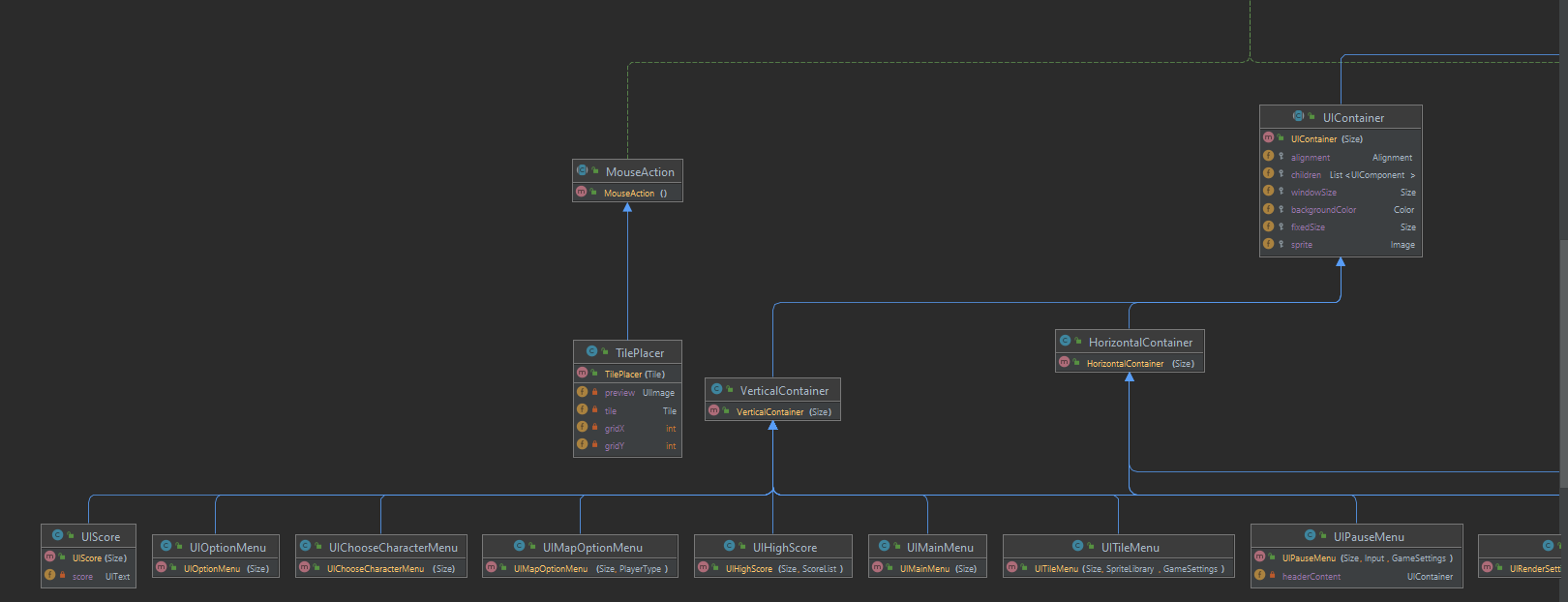


Acestea sunt clase ajutatoare Pentru MovingEntity : action unde se pot crea ai multe tipuri de actiuni : Pentru a face playerul/NPC-ul sa tuseasca, sa leviteze etc.

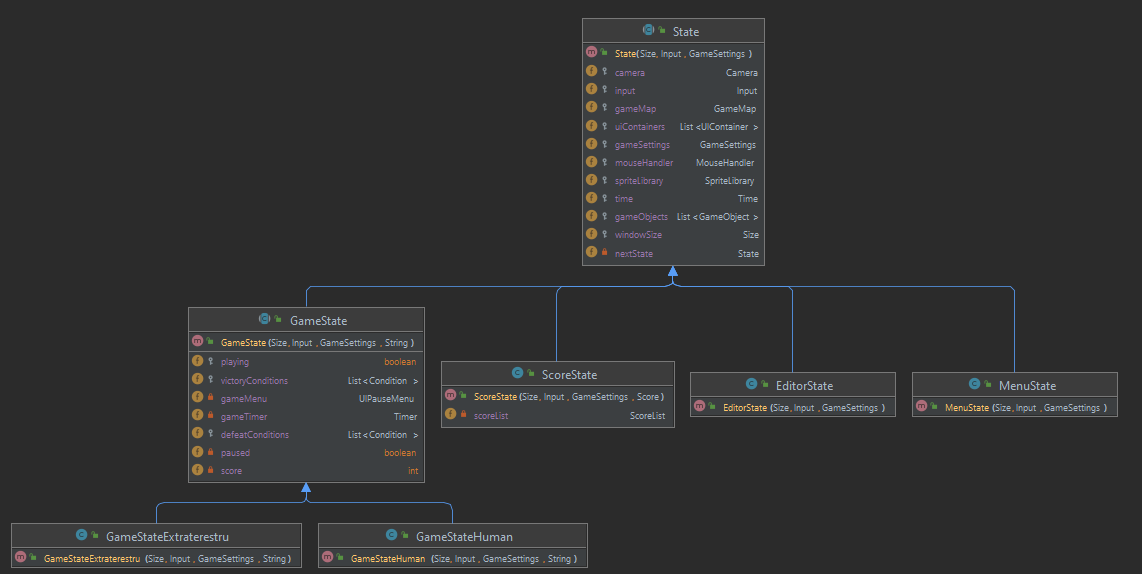
Controllerul de entitati este de 2 tipuri : Pentru player constructorul ia ca argument un input Pentru a puea “asculta” tastatura , iar NPC controller este “controlat” de catre program (unui NPC I se da o pozitie random de pe harta la care trebuie sa ajunga , iar prin intermediul controllerului acesta ajunge la pozitia x,y -> direct , fara un alg de pathfining (se putea folosi si un alg de pathfining printr-un graf ponderat, -> tileurile un-walkable vor avea ponderi inifnte, si tileurile de drum vor avea cele mai mici ponderi)). AiState are 2 stari -> atunci cand s-a gasit pozitia intra intr-o stare de Stand (cu propria animatie), iar cand inca este in cautare -> Wander .



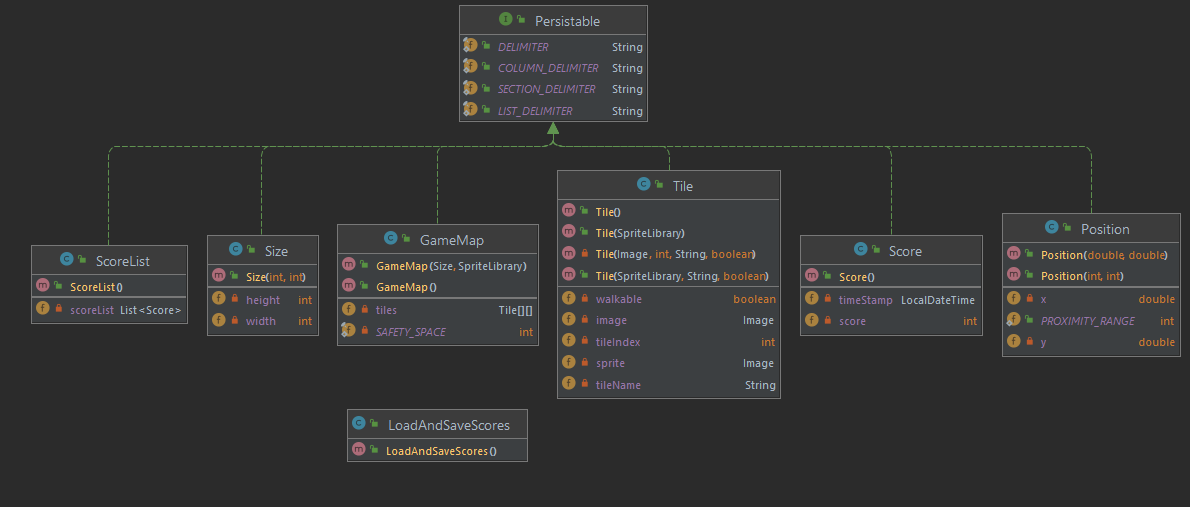
Aici avem structura UIComponent , ce se asemana cu componente dintr-o pagina web . UIComponennt contine o pozitie , un padding si un margin (de tipul spacing), dar si un UIContainer ca si parinte -> care va fi analizat in urmatoarea diagrama. UIComponent este abstract , fiind implementat de mai multe elemente ce vor fi folosite in interiorul containereleor : de ex UIButton, UIText etc.



Aici se afla Containerele propriu zise : VerticalContainer si HorizontalContainer -> difera modul in care adaugam la ele prin metoda addUIComponent: vertical creste pe verticala si horizontal pe orizontala. Exista mai multe Containere ce implementeaza VerticalContainer: de sc UIScore , UIOption menu etc.



Rularea jocului se bazeaza pe multiple stari prin care se trec Pentru a se face tranzitii de la (de exemplu), un meniu la altul , sau un meniu la jocul propriu zis. Asa ca exista mai multe Staturi : scoreState(se arata highScoreul), EditorState si rularea jocului prin GameState -> care este o clasa abstracta fiindca exista 2 jocuri: Pentru Extraterestru si Pentru Om (cu conditii de castig diferite , cu spriteul de baza diferit etc.).



Aceste serii de clase au fost folosite Pentru parsarea datelor (fie intr-o data de baza, prin LoadAndSaveScores Pentru highScoruri, fie intr-un text file, de exemplu cum s-a salvat GameMap).Fiecare clasa care implementa aceasta interfata trebuia sa contine 2 metode: serialize prin care se specifica modul de constructie al dateleor ce se vor salva , si apply-serialization (un parser) prin care se citeau datele dintr-un fisier si se faceau obiecte noi cu datele respective.