**Team : Black Box  
Ion Daniela Alexandra 333CB**

**Țeculescu Octavian Alexandru 335CC**

**Antiquarian**

1. ***Introducere***

Jocul nostru se incadreaza in subgenul survival horror game.Dezvoltarea acestuia se va face cu ajutorul game enginului Godot folosind o abordare hibrida: inglobam elemente 3D dar ne folosim de o grafica 2D. Gameplayul va fi axat in jurul experientei unui singur jucator, care isi va controla caraterul intr-o lume bidimensionala.

* 1. **Scopul sistemului**

Sistemul are ca scop relaxarea jucatorului, permitandu-i acestuia sa evadeze din cotidian intr-o lume noua, asumandu-si un caracter si actionand intr-un mod cu totul diferit . Pe masura ce petreci mai mult timp in lumea misterioasa iar povestea incepe sa se desfasoare vei observa o dorinta de a descoperi fiecare coltisor al mapei si de a acumula cat mai multe informatii despre noua lume. Perspectiva asupra jocului ti se va schimba si vei realiza ca nu iti mai doresti atat de mult sa duci misiunea la bun sfarsit ci vei dori sa faci din calatoria ta in lumea plina de monstri fiorosi si pericole ce te pandesc la tot pasul una cat mai palpitanta , profitand intr-un mod cat mai creativ de resursele intalnite si obiectele adunate.

* 1. **Documente referinte**
* Documentul de specificare a cerintelor

1. ***Obiective de proiectare***

Obiectivul principal al proiectarii este realizarea unei arhitecturi modulare care sa faciliteze implementarea functionalitatilor descrise in documentul de specificatie.

1. ***Arhitectura propusa***

Arhitectura pe care ne-am propus sa o folosim este MVC (Model – View – Controller).

* 1. **Prezentarea generala a arhitecturii sistemului**

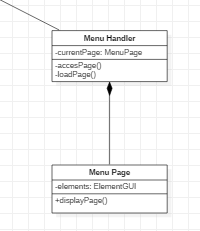
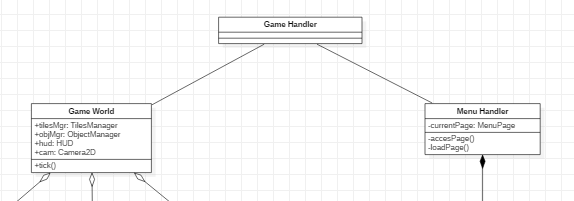
Arhitectura va fi construita peste enginul Godot si va urma modelul MVC (Model - View - Controller), fiind un model foarte popular in lumea jocurilor video.

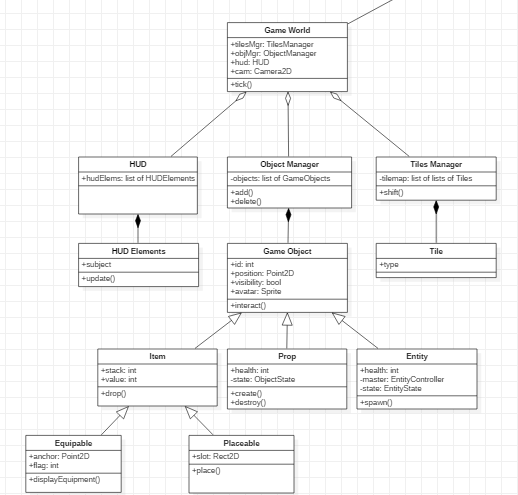
Acesta va folosi lumea jocului (care include un numar de obiecte "ascunse" pentru a invarti rotiile jocului) ca Model in care se vor petrece toate actiunile cu adevarat. Aceste actiuni includ orice interactiune intre jucator si joc.

View-ul va fi constituit din camera care prezinta lumea jocului dintr-un anume unghi, oferind o viziune de ansamblu asupra modelului, totodata fara a dezvalui detalii de functionare si logica din spatele evenimentelor. In categoria aceasta intra si HUD-ul in sine, care reprezinta o interfata de vizualizare a datelor din spate pentru ochii utilizatorului. Diverse interactiuni se vor desfasura prin intermediul ferestrelor de HUD, acestea fiind un hibrid intre actiune pe model (Controller) si vizualizari ale acestuia (View-uri).

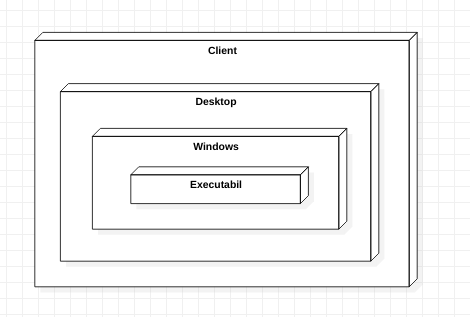
Controllerul este insusi caracterul jucatorului, dupa care se orienteaza Viewul si care (din motive de optimizare) activeaza in jurul sau diferite evenimente in asteptare din Modelul jocului. Caracterul jucatorului este aproape exclusiv singurul mod de interactiune a jucatorului cu mediul, exceptie facand meniurile si ferestrele de HUD care cer interactiuni directe prin intermediul mouseului.

* 1. **Decompozitia in subsisteme si responsabilitatile fiecarui subsistem**





* 1. **Distributia subsistemelor pe platforme hardware/software (**diagrama de distributie**)**



* 1. **Managementul datelor persistente (**fiṣiere, sistemul de baze de date folosit, schema conceptuala a bazei de date)
* **Fisierul de save** va fi salvat pe disc si actualizat periodic in timpul jocului. Daca nu exista se va considera ca este primul joc al jucatorului.
* **Asseturile** sunt incarcate dinamic in timpul jocului.
* **Fisierul de configuratii** va exista numai daca utilizatorul a facut cel putin o modificare peste configuratiile default
  1. **Controlul accesului utilizatorilor la sistem**

Data fiind natura pur recreativa a programului si lipsa accesului acestuia la un set de date ce pot fi folosite in scopuri maliţioase, consideram ca nu este necesara implementarea unui mecanism de securizare a jocului sau a accesului la datele generate de utilizator in acesta.

* 1. **Fluxul global al controlului**

Obiectul de tip player isi controleaza propriul comportament. Acestuia ii este permisa efectuarea unei singure actiuni intr-o perioada de timp. El poate efectua atat actiuni care blocheaza controlul pana cand acestea sunt finalizate (gather, pick, attack) dar si actiuni care pot fi intrerupte (navigarea) . De asemenea, fluxul jocului poate fi oprit de player cand este apasat butonul de pauza (care deschide un mic meniu de asemenea).

* 1. **Conditiile limita (**cazurile de utilizare limită**)**
* Plasarea de obiecte (de catre jucator) ce conduce la incarcarea unui numar prea mare de entitati in spatiul in lumii activ la un moment dat (spatiul din jurul jucatorului).
* Consumarea resurselor neregenerabile (de catre jucator) poate conduce intr-un punct in care jucatorului nu mai poate invinge.