### **Joc 3D Horror - "Asylum Escape"**

### **1. Analysis**

#### **1.1. Product Vision**

„*Asylum Escape*” urmărește să ofere o experiență imersivă de horror, unde jucătorii trebuie să scape dintr-o clădire bântuită.

* **FOR** jucătorii pasionați de jocuri horror care caută o experiență captivantă și tensionată.
* **WHO** doresc să exploreze medii imersive și să rezolve puzzle-uri provocatoare într-un cadru înfricoșător.
* **The "Asylum Escape"** este un joc 3D horror interactiv.
* **THAT** oferă o combinație unică de explorare, rezolvare de puzzle-uri și interacțiune cu un AI adaptiv, pentru a oferi o experiență captivantă și intensă.
* **UNLIKE** jocurile de groază obișnuite care oferă scenarii previzibile și interacțiuni limitate,
* **OUR PRODUCT** creează o atmosferă dinamică și personalizată, în care acțiunile jucătorului influențează direct dificultatea și comportamentul AI-ului.

#### **1.2. Product Features, Scenarios și User Stories**

**Requirements:**

* Un mediu 3D realist, cu elemente interactive.
* Sistem de navigare bazat pe **NavMesh** pentru entitatea malefică.
* Puzzle-uri progresive cu dificultate dinamică.

**Features:**

* Explorare a clădirii bântuite.
* Interacțiune cu obiecte (uși, obiecte colectionabile).
* AI avansată pentru entitatea malefică.

**User Stories:**

* „Ca jucător, vreau să navighez în clădire pentru a descoperi indicii.”
* „Ca jucător, vreau să fiu urmărit de o creatură malefică în timpul jocului.”
* „Ca jucător, vreau să aud sunete ambientale.”
* „Ca jucător, vreau să rezolv puzzle-uri pentru a avansa în poveste.”
* „Ca jucător, vreau să evit entitatea malefică pentru a supraviețui.”
* „Ca jucător, vreau să am o atmosfera horror și tensionată care să mă inducă în atmosferă.”

**Backlog:**

* **Sprint 1:** Configurarea scenei de bază, implementarea NavMesh-ului, adăugarea funcționalităților de bază pentru jucător.
* **Sprint 2:** Optimizarea AI-ului entității.
* **Sprint 3:** Testarea sunetelor și adăugarea detaliilor vizuale.
* **Sprint 4:** Implementarea interacțiunilor complexe și animațiilor pentru entitate.
* **Sprint 5:** Optimizare Gameplay și introducerea de mecanici noi.
* **Sprint 6:** Crearea de puzzle-uri specifice pentru fiecare nivel.
* **Sprint 7:** Pregătirea pentru livrarea finală.

### **2. Planning**

#### **2.1. Backlog prioritizat din User Stories**

* Fiecare User Story este spart în task-uri gestionate prin GitHub Projects.
* Exemple de prioritizare:
  + **Task:** Configurarea NavMesh-ului. **Prioritate:** Înaltă.
  + **Task:** Crearea de puzzle-uri interactive. **Prioritate:** Medie.
  + **Task:** Adăugarea detaliilor audio-vizuale. **Prioritate:** Scăzută.

#### **2.1. Sprint 1:** Configurarea scenei de bază, implementarea NavMesh-ului, adăugarea funcționalităților de bază pentru jucător.

* **Task:** Setarea terenului și a structurii clădirii
* **Task:** Iluminare
* **Task:** Camere și uși
* **Task:** Setare de bază pentru player
* **Task:** Configurarea NavMesh-ului în Scenă
* **Task:** Crearea și Configurarea Entității (Creaturii)
* **Task:** Implementarea Comportamentului de Urmărire
* **Task:** Testarea și Ajustarea Comportamentului AI

#### **2.2. Sprint 2:** Optimizarea AI-ului entității.

* **Task:** Ajustarea NavMesh-ului pentru obstacole dinamice
* **Task:** Adăugarea comportamentului de staționare temporară
* **Task:** Optimizarea performanței AI-ului

#### **2.3. Sprint 3:** Testarea sunetelor și adăugarea detaliilor vizuale.

* **Task:** Adăugarea detaliilor vizuale: Încorporarea de elemente detaliate, cum ar fi mobilier sau decor, pentru a adăuga autenticitate scenei.
* **Task:** Iluminare avansată: Configurarea surselor de lumină pentru a îmbunătăți atmosfera și pentru a evidenția elementele importante.
* **Task:** Adăugarea efectelor sonore: Integrând sunete ambientale și efecte sonore pentru interacțiunile principale.
* **Task:** Testare sunet: Verificarea calității și sincronizării sune

#### **2.4. Sprint 4:** Implementarea interacțiunilor complexe și animațiilor pentru entitate.

* **Task:** Adăugarea patrulei pentru monstru , vrem ca acesta sa patruleze holurile atunci cand nu ne urmărește
* **Task:** Setarea animațiilor corecte pentru diferitele interacțiuni
* **Task:** Posibilitatea interacțiunii cu player-ul, nu doar urmărirea acestuia, alături de soundFX-urile aferente

#### **2.5. Sprint 5:** Optimizare Gameplay și introducerea de mecanici noi.

* **Task:** Optimizarea mecanicilor de mișcare ale jucătorului: ajustarea vitezei de deplasare și adăugarea de animații
* **Task:** Testarea mecanicilor existente
* **Task:** Puzzle-uri avansate: integrarea unui mecanism care necesită completarea unui puzzle sub presiunea demonului
* **Task:** Evenimente declanșate de acțiuni: declanșează evenimente speciale (ex. apariția entității, activarea unui puzzle nou)

#### **2.6. Sprint 6:** Crearea de puzzle-uri specifice pentru fiecare nivel.

* **Task:** Implementarea puzzle-urilor simple: de tip „găsește și folosește obiectul” sau bazat pe secvențe
* **Task:** Crearea puzzle-urilor avansate: cu mai multe etape
* **Task:** Integrarea puzzle-urilor în gameplay
* **Task:** Testare și ajustare

#### **2.7. Sprint 7:** Pregătirea pentru livrarea finală.

* **Task:** Optimizarea performanței jocului
* **Task:** Rezolvarea ultimelor bug-uri
* **Task:** Refactorizarea codului
* **Task:** Finalizarea tuturor animțiilor și efectelor
* **Task:** Testarea scenariilor finale

### **3. Design**

#### **3.1. Diagrame pentru arhitectura dorită**

* **Diagrama Contextuală:** Prezintă componentele principale ale aplicației (jucător, entitate AI, mediu 3D).
* **Diagrama de Componente:** Ilustrează legăturile dintre modulele aplicației (controler AI, motor de joc, mecanici de interacțiune).
* **Diagrama de Flux:** Detaliază interacțiunea dintre utilizator, AI, și puzzle-uri.

### 

### **4. Tehnologii utilizate**

#### **4.1. Engine de Dezvoltare: Unity**

* **Unity** este motorul de dezvoltare principal utilizat pentru acest proiect, datorită flexibilității și capabilităților sale puternice de a crea jocuri 3D

#### **4.2. Programare și AI**

* **C#** este limbajul principal folosit în Unity pentru scripting. Prin C#, s-au implementat sistemele de algoritmi de pathfinding, State Machine, Trigger-uri și Evente.

#### **4.3. Grafică și Design 3D**

* **Blender**: Folosit pentru crearea și animarea modelelor 3D (ex. entitatea malefică, obiectele esențiale, decorul clădirii). Modelele sunt exportate în Unity și integrate în scenă.
* **Shader Graph (Unity)**: Utilizat pentru a crea efecte de lumini și umbre dinamice care cresc tensiunea în joc. De asemenea, shader-ele sunt folosite pentru a adăuga efecte vizuale speciale, cum ar fi distorsionarea vizuală în prezența creaturii.

#### **4.4. Sunet și Atmosferă**

* **FMOD**: Integrat pentru gestionarea sunetelor de ambianță și a efectelor sonore dinamice, FMOD oferă flexibilitate pentru a sincroniza sunetele cu evenimentele din joc și pentru a crea o atmosferă realistă.
* **Audacity**: Folosit pentru editarea și prelucrarea sunetelor, asigurându-ne că fiecare element audio se încadrează perfect în atmosferă (sunete de pași, uși care scârțâie, respirația jucătorului, etc.).

#### **4.5. Gestionarea Proiectului și Controlul Versiunilor**

* **Git și GitHub**: Pentru managementul codului și colaborarea eficientă în echipă. Toate modificările sunt versiuni controlate, iar proiectul este sincronizat constant între membrii echipei pentru a asigura o dezvoltare fără conflicte.
  + **Link GitHub**: <https://github.com/sumithesum/Asylum-Escape>

### **5. Livrabil intermediar**

Documentul este conceput pentru a reflecta progresul și planurile proiectului și a fost completat pe baza analizei și documentației furnizate. În repository-ul GitHub sunt incluse:

* Product Vision.
* Backlog detaliat.
* Diagrame și descrieri arhitecturale.

[**Link GitHub**](https://github.com/sumithesum/Asylum-Escape)**:** <https://github.com/sumithesum/Asylum-Escape>