

Software Architecture Report

1. What the purpose of the software project is

Proiectul "Assassin's Creed: Back To The Future" își propune să redefinească experiența de joc prin îmbinarea a două francize iconice, Assassin's Creed și Back To The Future. Scopul nostru fundamental este de a oferi jucătorilor o incursiune captivantă și inovatoare în lumea călătoriilor în timp și a asasinilor, unde pot explora epoci istorice variate și pot influența evenimente cruciale care au modelat istoria umanității.

Până în acest moment, proiectul a atins un punct de referință semnificativ prin implementarea unui stadiu minim viabil (MVP). A fost creat un prototip funcțional, oferind jucătorilor o privire asupra mecanicilor de bază, inclusiv călătoria în timp și abilitățile de asasinare. Primul stagiul în secolul 15 a fost realizat, prezentând un boss emblematic și introducând jucătorii în lumea fascinantă a jocului.

Ce urmează până la indeplinirea completă a proiectului:

1. Extinderea universului jocului:
 - Adăugarea de alți bossi reprezentativi din diferite secole pentru a diversifica și îmbogăți experiența de joc.
 - Introducerea de noi setări și epoci istorice pentru a oferi jucătorilor o gamă extinsă de locații de explorat.
2. Rafinarea gameplay-ului:
 - Continuarea rafinării mecanicilor de bază, asigurându-se că călătoria în timp și abilitățile de asasinare sunt fluide și captivante.
 - Identificarea și corectarea eventualelor bug-uri pentru a garanta o experiență de joc stabilă.
3. Extinderea poveștii și caracterizării:
 - Dezvoltarea și extinderea poveștii personajului principal, Alex Mercer, pentru a aduce mai multă profunzime și conexiune emoțională.
 - Introducerea de elemente narrative noi pentru a menține interesul jucătorilor pe durata întregii aventuri.
4. Optimizare și perfecționare vizuală:
 - Îmbunătățirea calității grafice și a designului vizual pentru a oferi o experiență uimitoare jucătorilor.
 - Asigurarea unei optimizări adecvate pentru a asigura o performanță stabilă pe o varietate de platforme.
5. Planificarea viitoarelor lansări și actualizări:
 - Colectarea feedback-ului jucătorilor pentru a identifica zonele de îmbunătățire și pentru a planifica actualizări viitoare.

- Pregătirea pentru lansarea publică, inclusiv strategii de marketing și distribuție.

Prin atingerea acestor obiective, ne propunem să oferim jucătorilor o experiență de joc excepțională, plină de aventură, acțiune și explorare într-o lume fascinantă unde trecutul și viitorul se intersectează într-un mod captivant.

2. Guides on how to ...

Run the project locally

1. Instaleaza Unity Hub cu Unity v2022.3.10f1
2. Setup Visual Studio 2022 pentru Unity Development
3. Instaleaza Git si Git LFS. Daca nu iti plac terminalele, poti oricand sa faci setup la GitHub Desktop
4. Cere access la git repository ul nostru
5. Cere access la Discord server si la diferite chat grupuri

Design patterns utilizate:

1. Component Pattern - este pattern-ul folosit in mod implicit in Unity Engine. Toate Game Objects sunt containere pentru componente, care adauga functionalitate.

2. Strategy Pattern - Este un behavioral pattern prin care sunt definiti mai multi algoritmi interschimbabili, alesi la runtime in functie de mai multi factori. Este utilizat in cadrul NPC-urilor, care au idle behavior, chase behavior si attack behavior, in functie de pozitia fata de player.

3. State pattern - Schimbarea starii NPC-urilor se realizeaza prin intermediul State Pattern-ului, in cadrul componentei Animator, in functie de anumite variabile.

3. Application entry points:

Main Menu:

Meniul principal de unde jucătorii pot începe un joc nou, încărca un joc salvat, accesa setările și ieși din joc.

Pause Menu:

Este disponibil oricand in timpul jocului, pentru a permite jucatorilor sa isi salveze progresul, modifica setarile sau iesi din joc.

a) Data sources:

Asset libraries:

În procesul de dezvoltare a primului nivel al jocului, am recurs la utilizarea unui pachet de resurse digitale (asset pack) specializat pe tematica castelurilor medievale. Acest pachet, ales pentru autenticitatea și detaliile arhitecturale pe care le oferă, a fost o sursă valoroasă în replicarea și adaptarea anumitor structuri medievale caracteristice. Această abordare ne-a permis să infuzăm nivelul cu o atmosferă istorică credibilă și să oferim o experiență vizuală îmbogățită jucătorilor. ([Link medieval castle asset pack](#))

Pe lângă elementele vizuale, a fost esențial să asigurăm și o interfață utilizator (UI) eficientă și intuitivă. În acest sens, am ales să implementăm un pachet de meniu predefinit, achiziționat din Unity Store. Acest pachet de meniu a fost selectat datorită designului său sofisticat și ușor de navigat, facilitând astfel o experiență de utilizare fluidă și accesibilă. Acesta integrează elemente de navigare esențiale, precum și funcții de personalizare, contribuind la o interfață generală mai atractivă și funcțională. ([Link menu asset pack](#))

b) Data inputs:

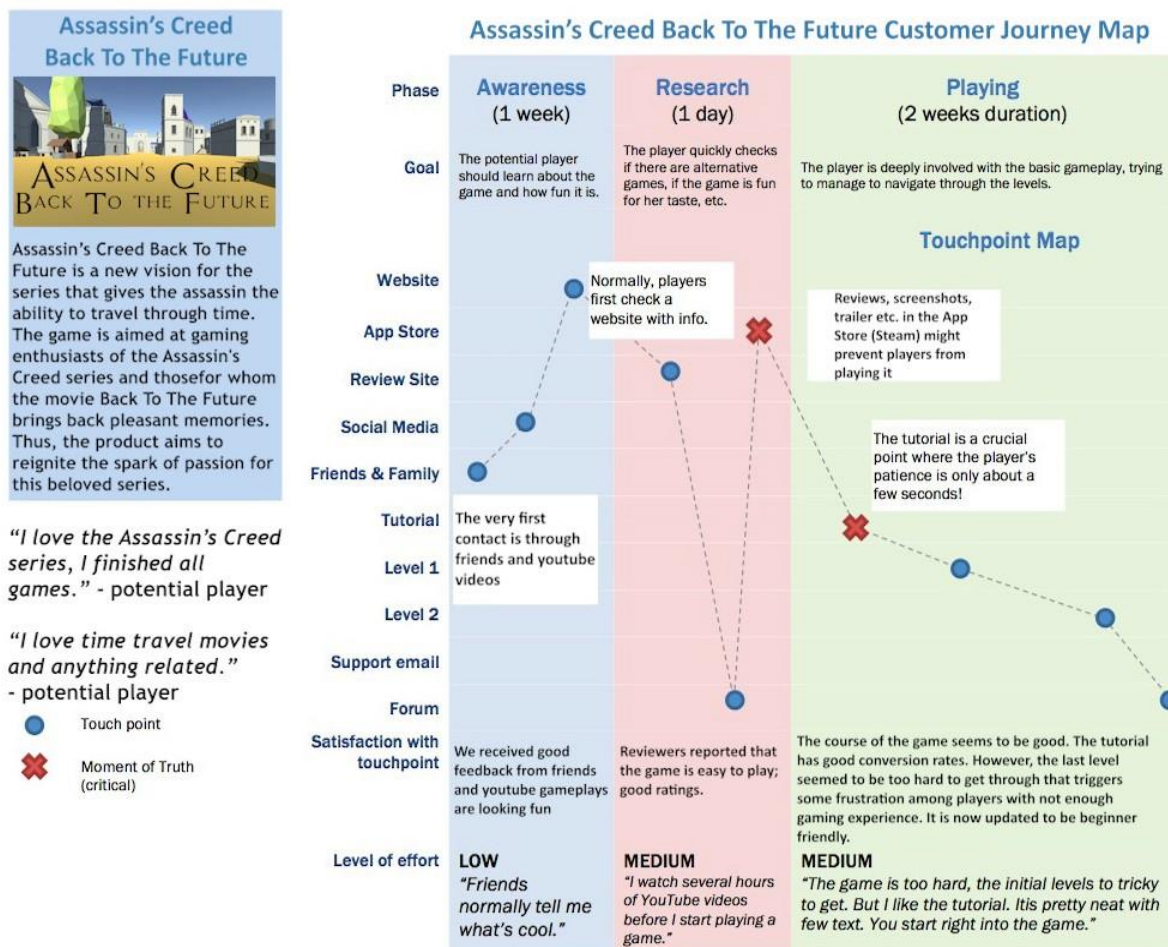
User input:

Inputurile de la utilizator în cadrul aplicației noastre sunt captate printr-o varietate de dispozitive, fiecare având rolul său specific în interacțiunea cu software-ul. Aceste dispozitive includ tastatura, mouse-ul și controlerul, fiecare contribuind la o experiență de utilizare completă și diversificată.

c) Configuration Files:

1. **Graphic Settings:** Fișiere care stochează preferințele utilizatorului pentru rezoluție, calitatea texturilor, umbre și alte setări grafice.
2. **Control Maps:** Fișiere care salvează configurările personalizate ale jucătorului pentru taste sau controlere.
3. **Gameplay Preferences:** Fișiere de configurare pentru nivelul de dificultate, personalizarea UI, nivelurile audio și alte preferințe personale.

4. A) User/data journeys



4. B) Most valuable output

Cel mai valoros rezultat al jocului nostru este crearea unei experiențe de joc imersive și profund captivante, care să ofere jucătorilor o combinație unică de poveste complexă, personaje memorabile, gameplay inovator și o lume vastă de explorat. Aceste elemente nu doar că mențin jucătorii angajați și interesați, dar le oferă și o experiență de joc profundă și semnificativă, care răsună dincolo de ecran.

În plus, punem un accent deosebit pe respectarea și valorizarea comunității noastre de jucători. Ne dorim să cultivăm un mediu în care jucătorii se simt ascultați, apreciați și parte integrantă a unei comunități mai largi. Încurajăm feedback-ul constructiv și dialogul deschis, având ca scop crearea unei comunități puternice și dedicate în jurul jocului nostru. Acest angajament față de jucători și dezvoltarea unei comunități active sunt esențiale pentru succesul pe termen lung al jocului. Prin aceasta, nu doar că ne extindem baza de utilizatori, dar și construim o bază de fani loiali, care contribuie la evoluția și popularitatea continuă a jocului nostru.

5. Deployment plan

În prezent, jocul nostru este în faza activă de dezvoltare, fiind implementat intern prin intermediul unui proces CI/CD utilizând GitHub Actions. Aceasta ne permite să automatizăm testarea și integrarea funcționalităților noi, asigurând o evoluție constantă și eficiență a proiectului.

Ne concentrăm, de asemenea, pe pregătirea pentru lansarea jocului pe Steam, un pas esențial pentru a ajunge la o audiență mai largă. Acest demers necesită o atenție deosebită la standardele de calitate și performanță impuse de platformă. Eforturile noastre sunt îndreptate spre optimizarea jocului, pentru a asigura o experiență fluidă și captivantă utilizatorilor la momentul lansării.

Această abordare dublă, de a menține un ciclu de dezvoltare agil intern și de a ne pregăti pentru publicarea pe o platformă recunoscută, ne poziționează jocul pentru succes și asigură că va satisface așteptările jucătorilor.

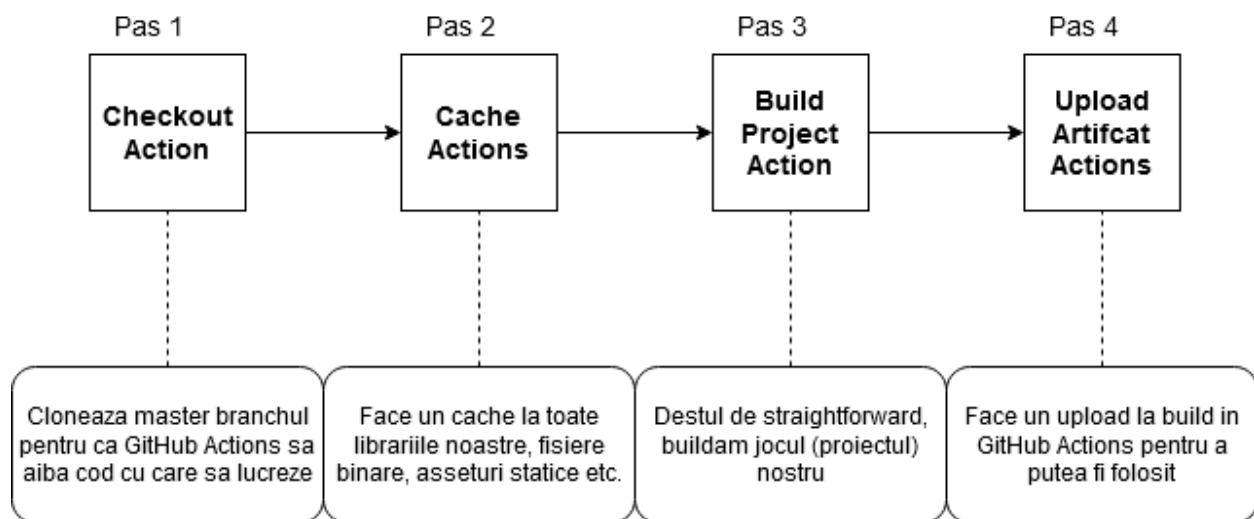
How the CI/CD pipeline works.

Este esențial să construim, să implementăm și să testăm jocul nostru cât mai repede posibil, pentru a putea primi rapid feedback de la clienți (sau fani) și, de asemenea, schimbările mici de cod sunt mult mai ușor de gestionat, ceea ce înseamnă că trebuie să petrecem mai puțin timp testând și făcându-ne griji cu privire la cazurile limită.

Cum lucreaza un Job in unity actions? (exemplul este pipeline-ului nostru)

Mai intai este rulat jobul de **Unit Tests** din GitHub Actions, care presupune **Checkout Action** -> **Rularea Unit Testelor** -> **Upload Artifacts**

Daca jobul de unit teste a fost rulat cu succes, se trece la jobul de **build** pentru **StandaloneWindows64** care are urmatoorii pasi:



6. Description of the QA process

- **PlayerGravity** - verifica daca valoarea gravitatiei este egala cu -9.8f in scriptul **PlayerMotor**.
- **SaveScriptKills** - verifică dacă numărul de morti este egal cu 0 la început cand facem LoadSaveObject, deoarece progresul jocului variaza în funcție de numărul de morti.
- **ClearTests, DeleteTests, ExistsTests, LoadTests, SaveTests** - verifica functionalitatea salvarii jocului.

7. External dependencies included in the project

- [GameCI](#) - o colectie de dockerizata de Unity Editors, pentru platforme diferite, care ajuta cu setupul CI/CD pipeline-urilor pentru GitHub Actions (si GitLab CI).
- [InputManager](#) - biblioteca interna din Unity, care mapeaza actiunile din Engine cu Inputurile userilor.
- [NavMesh](#) - API din Unity care face interogari spatiale, cum ar fi localizarea traseelor si testele pentru “walkability” a AI-urilor adaugate in joc.