A feladat egy játékmotor fejlesztése OpebGL grafikus könyvtárra alapozva. A játékmotor megvalósítás követően pedig egy demo játék elkészítése ami demonstrálja a játékmotor képességeit.

A feladat elsődleges része magának a játékmotornak a fejlesztése. De ahhoz, hogy valamilyen céllal fejlesszem választottam egy játékot, ami támpontot ad a szükséges funkciókhoz. Az általam választott játék egy labirintus játék.A választás oka, hogy a játékot az egészen egyszűrűtől a bonyolultig fokozatosan lehet fejleszteni. És előnye, hogy 2 és 3 dimenzioban is ertelmesen megvalosithato. A játék lényege, hogy kijussunk a labirintusból. A játékot úgy fejlesztem folyamatosan, hogy tükrözze a játékmotor aktuális képességeit.

A szükséges funkciókat két részre lehet bontani, az alapján hogy melyik részhez kell a funkció. A funkciók egyik része a játékmotor általános működéséért felelős. A másik része pedig a játék specifikus funkciókért. Áltaános funkciónak mondható a megjelenítés, a fizika és a kamera is. A fizika felelős azért, hogy a játék valósághű legyen és úgy tudjuk irányítani, ahogy azt a valóságban megszoktuk. A megjelenítés a legfontosabb része a motornak. Ez felel azért, hogy a játékos azt lássa a képernyőn amit szeretne. A Kamera, azért szükséges, hogy tudjuk kezelni éppen mint mutatunk a képzelet beli világból. A kamera lényegében a megjelenítés része, csak hozzáad vagy elvesz tőle még valamit. A játékspecifikus funkciók csoportjába tartozik a labirintus kezelése, mivel szeretnénk, hogy ez a motor szintjén legyen implementálva.

A követelmények megállapítását a specifikáció és a funkciók alapján végeztem, ezért is nagy az átfedés közöttük, de itt más szempontból vizsgálom őket. Az egyik legalapvetőbb követelmény, hogy meg tudjam jeleníteni a kívánt képet OpenGl segítségével, ehhez további követelmény még, hogy megfelelően gyors és zökkenőmentes legyen ahhoz, hogy a felhasználó szépnek lássa. Ahhoz, hogy irányítani is tudjuk mindenképp szükséges imputkezelés, az inputkezeléshez további elvárás még, hogy az testreszabható legyen, vagyis ki lehessen választani melyik input hatására mi történik. A játék élvezhetőségéhez és logikus működéséhez szükség van fizikár. A fizikától elvárt, hogy a valóságon alapuljon, de ugyanakkor lehessen parametrizálni. Ez alatt azt kell érteni, hogy például a gravitáció csökkentésével elérhetem azt a hatást mintha másik bolygón játsznánk. Ahhoz, hogy külső eszközzel gyártott modelleket használhassunk ezeket be kell tölteni és utána tudni kell kezelni. Ahhoz, hogy játszható legyen a játék kell valamilyen játéklogika, amit akár módosíthat is a fejlesztő. Ez azért fontos, hogy széles körben felhasználható legyen a motor.

Legelőször a játékfejlesztés során használatos tervezési minták tanulmányozásával foglalkoztam. Ezek közül szeretném kiemelni a legfontosabbakat. A Command minta többek között ahhoz használatos, hogy a programunk tudja kezelni a imputokat, főleg ha azok személyreszabhatóvá akarjuk tenni. Az Observer minta minden grafikus alkalmazás egyik legfontosabbja. A mint alapvető része a Model-View-Controller architectúrának. De ahhoz is hasznos ha például egy achievement rendszert szeretnénk kiépíteni. A Game loop és Update method használata együtt javasol. Az előbbi felel azért, hogy a játék folyamtosan fusson és itt történik az idő kezeleése is, ez lényegében egy sokáig futó ciklus. Az utóbbi a ciklus minden iterációjában meghívódó függvenyeket jelent amik frissítik az állapotokat. Ezek használata a játékfejlesztés során elengedhetetlen.

A tervezés során a könnyebb kezelhetőség és a modularitás érdekében két részre osztottam a projektet. A játékmotorra és a játékra. Ezek között olyan kapcsolatot építettem ki, hogy a motor által generált könyvtárat használja a játék, de mind a belépési pont mind a fő ciklus a motorban található. A fejlesztést c++ nyelven végeztem Visual Studio 2019 környezetben c++17 fordítóval. Verziókezelést használtam, aminel platformja a github. A labirintusok kezeléséhez alkottam egy fileformátumot, amivel leírom őket. Ezen fileok kezeléset a motor beépítetten támogatja. A tervezés során nagy hangsújt fektettem az objektum orintált elvek betartására és a tervezési minták használatára. Ezen kívül igaz, hogy a projektet openGl könyvtárr alapozva fejlesztem, de szerettem volna meghagyni a lehetőséget, hogy ez könnyen lecserélhető legyen. Ezért minden natív részt elrejtettem egy általános interface mögé ami miatt a specifikus részek könnyen lecserélhetőek.

Az egyszerűség kedvéért használtam külső könyvtárakat. A fizikai szimulációhoz a Bullet Physics SDK-t használtam. A megjelenítéshez az ImGui könyvtárat használtam ami hatékonyan képes renderelni, de használatához szükség van egyébb könyvtárakra amik például kezelik az inputokat. A hatékony fejlesztéshez készítettem egy log rendszert aminek alapjául az spdlog szolgált. A fejlesztés során felmerülő problémák miatt úgy döntöttem használok valamilyen projekt generátort. A választásom a Premakere esett.

Az implementálás során két projektet hoztam létre, az egyiket a motornak a másikat a játéknak. Megvalósítottam az ablakok és inputok kezelését. De csupán egy bemenetet volt képes egyszerre kezelni. A kiterjesztés során előjött problémák arra kényszerítettek, hogy visszatérjek a legutóbbi stabil verzióhoz, de elállítódtak a solution file include és linker beállításai valamint a projekt szerkezete. Ekkor döntöttem úgy, hogy projekt generátort használok és elkezdtem hozzá scripteket írni lua nyelven.

A tesztelés lényege, hogy segít abban, hogy tudjam jól működik-e a projekt egy-egy változtatás után. Ezekhez két féle eljárást használtam. A grafikus könyvtárak működéséhez manuális teszteket készítettem, ahol azt ellenőrzöm siekresen lettek-e integrálva a projektbe. A belső logika teszteléséhez Unit teszteket használok, de ezeknek nem sok dolguk van.

Összefoglalásként elmondható, hogy sikerült megtervezni a projektet és belekezdeni az impelmentálásába, ahol nem várt problémák késleltették a haladást. Azonban az eszközölt változtatások pozitívan hatnaj a projektre, főleg abból a szempontból, hogy több platformon lehessen fejleszteni.