Pre ukážku volím projekt starší projekt Raprots (2021) Ide o projekt ktorý bol vytvorený na základe Kenney game-jamu ktorého podmienkou že sa musí použiť asset od totho umelca. (zvečša lowpoly spritess) Samotný projekt Raptors má predstavovať niečo na ukážku a konkrétne volím Controll skript ktorý kontroluje celé jadro hry.

https://github.com/dretnoth/Raptors-KJ/blob/main/Assets/Scripts/Controll.cs

info: <https://dretnoth-portfolio.cms.webnode.sk/l/raptors-gamejam/>

https://dretnoth.itch.io/raptors-kenney-jam

Samozrejme hra má veľa priestoru na dopracovanie, hlavne pokiaľ by sa upustilo od sprite packu a dopracovali sa funkcie, ale ako sa povedalo je to projekt na ukážku.

V Awake() sa zadefinuje Game Controller aby čokoľvek mohlo nájsť tento skript i keby malo prehľadávať celú hru, pokiaľ rovno nemajú definovaný odkaz.

V Start() zabezpečím spustenie/ aktiváciu jednotlivých častí programu a taktiež spustenie hudby na pozadí. Start odkazuje na data.cs kam boli odsunuté premenné či skôr konštanty ako prefaby, narábanie s UI (menu a HUD bar) a podobne. S data.cs sa potom narába počas celého scriptu.

V Update() sa pracuje s prvkami ktoré nespadajú pod jednotlivé game komponenty ale musí nad nimi byť kontrola. Najdôležitejším prvkom je tu časovač/hodiny pre GameTikEvent()

V LevelUP() sa spracováva povýšenie úrovne hry, rozšírenie hracej plochy a priradenie upgradu pre hračovu loď.

EscapeToFromMenu() a Button...() robia čo majú popísané…

SpawnPlayer() robí čo sa má napísane. Nutno povedať, že v tomto projekte sa ešte stále využíva metóda Instatiate [gameObject] -> Destroy -> Insatiate [new] na novo. V novších projektoch je to Instatiate if dont exist -> namiesto Destroy povedz ze si umrel a Deactivate -> vyhľadaj deaktivovaný: ak áno, Reactivate and Reset. Else, Instatiate new

GameEventGameIsStarted() je spustenie samotného herného levelu, i keď to nie je skutočný Event type skript, vlastne žiadne z mojich event pomenovaných skriptov nie je podľa unity definície event type skript

GameEventGameIsEnding() sa stará o ukončenie levelu

EventBaseJumpOut() je pre spustenie finálnej kvázi animácie s odletom lodi. S low polly sa toho moc spraviť nedá.

(Potom je tu zopár buttonov pre UI)

GameCleanTheScene() má za úlohu vyprázdniť scénu, tak aby žiadny GameObject sa nezatúlal kam nemá, by Destroy()

EventCreateResource(…) cieľom hry je zbierať suroviny ktoré vypadnú z asteroidov, monštier a lodí. Aby neboli nalepené na jednom mieste je tam komponent bullet. Zas používa sa nákladný Instant – destroy -Instant new.

GameTikEvent() sa stará aby mapa mala gameObjecty

Spawn...() scripty sú pre každý typ hrozby samostatné. Obzvlášť keď SpawnOtherCraftHazard je vlastne konkurencia ktorá robí to isté ako player, zbiera suroviny s cieľom dosiahnuť potrebné množstvo na opustenie hracej plochy

EventAimChorsarGotData() GO pre efekt zacielenia pre torpedo

Asi by bolo vhodné povedať že odsunutie nepotrebných údajov a funkcií s controll.cs do data.cs prišlo z iného projektu (Visual novel založený skorej ktorý, nemá žiadne reálne funkčne GO (okrem kontajnerov pre kapitoly) a celý pracuje na UI úrovni) ktorého controll.cs sa natiahol na 2500+ riadkov kódu. S pridávaním nových a nových funkcií skrz využívanie child elementov ako kontajnery pre údaje (GO iba so skriptom kapitola, ktorý má v sebe mnoho child GO iba so skriptom part.cs), sa došlo k záveru že (používanie základného Inšpektoru a tohto typu zápisu inštrukcií pre jednotlivé kapitoly), je nepostačujúce a skonzumuje priveľa času.

Prvým bodom bolo vytvorenie vlastného Inšpektoru čo sa ukázalo ako trýznivé dobrodružstvo v ktorom sa mi podarilo implementovať len jeden krok. Výber použitého emoji podľa jeho spritu.

Očividným riešením však je prepracovať projekt do formy, kedy by som tvoril/ písal kapitolu v spustenej vysual novel za behu programu ktorý by zapísal dané údaje/inštrukcie do súboru/súborov. Avšak keďže je to rozsiahly krok; môj projekt o vysual novel je pozastavený.