Iterium Documentation



Revision: 2

Contents

[1. Game Overview 3](#_Toc126918979)

[1.1. Game Story 3](#_Toc126918980)

[1.2. Factions and ships 4](#_Toc126918981)

[1.3. Controls 4](#_Toc126918982)

[1.4 Scoring & Levels 4](#_Toc126918983)

[1.5. Upgrading 5](#_Toc126918984)

[2. Unity Project 5](#_Toc126918985)

[2.1. Project requirements 5](#_Toc126918986)

[2.2. Folder Structure 5](#_Toc126918987)

[2.3. Classes & Assemblies 6](#_Toc126918988)

[2.4. Input System 8](#_Toc126918989)

[2.5. Save System 8](#_Toc126918990)

[2.5.1. Save files and methods 8](#_Toc126918991)

[2.5.2. Path 8](#_Toc126918992)

[2.5.3. Player data 9](#_Toc126918993)

[2.5.4. Leaderboard data 9](#_Toc126918994)

[2.5.5. File save handler 9](#_Toc126918995)

[2.6. ScriptableObjects 9](#_Toc126918996)

[2.7. Managers 9](#_Toc126918997)

[2.8. Pooling 9](#_Toc126918998)

[2.9. UI 9](#_Toc126918999)

[2.10. Audio 9](#_Toc126919000)

[2.11. Builds 9](#_Toc126919001)

# Game Overview

|  |  |
| --- | --- |
| Genre | Top-down space shooter |
| Theme | 1980s Arcade |
| Perspective | 2.5D |
| Game engine | Unity (version 2021.3.15f1) |
| Scripting language | C# |
| Build targets | WebGL – Desktop browsers |
| License | Open source |
| Gameplay | The game follows a typical arcade genre where you try and survive as long as possible to gain the highest score and be entered on the leaderboard. Score can be achieved by shooting asteroids, collecting Iterium crystals, killing AI ships and Xoid flying Saucers.    At the start of the arena battle you will receive 3 ships (lives) and the battle will end when you lose all your lives. Asteroids do not do damage to your ship but will push your ship from course depending on the weight of the selected ship and size of the asteroid. Enemy fire can be destroyed with your own fire and damage can be reduced by deploying your shield.  When you or the AI are destroyed, a re-spawn delay will provide a short reprieve to the opponent to collect any floating crystals and rack up further points.    Iterium crystals are dropped randomly by destroying asteroids, crystals are used to upgrade the firepower, shield and speed of your spaceship. Picking up crystals before your opponent is crucial to prevent them from also upgrading, increasing your maneuverability, offensive and defensive capabilities will give you the edge in battle.    There are 3 faction ships to choose from, each having their own balance of firepower, shields and speed. Each of the ship properties can be upgraded twice, the USSR Hammer is the heaviest ship and will be less impacted by asteroids while the lightest Chinese Rapier, although the fastest ship will struggle against flying debris. Each of the ship’s statistics can be viewed on the faction selection screen, including the status after upgrading. |

## Game Story

In the 2055 the moon being struck by a large asteroid, left considerable debris containing a rare crystal called lterium.

You own a mining organization that is tasked to recover as much of the precious resource before other hostile organizations. Shoot the asteroid debris to expose the Iterium crystals and increase your wealth, keep a close eye on other miners who will stop at nothing to destroy you. To make the challenge even more difficult, an alien race called Xoids, use Iterium in their warp drives and have also entered the fight for extraction.

## Factions and ships

The Sword belongs to the USA faction and is the most balanced of the 3 ships. The ships most impressive feature is its firepower, medium thrust capabilities with its shields being its weakest feature. Good balance of attack and maneuverability when fully upgraded. Weight is in the medium range and impacts how you are affected by the weight of an opposing asteroid.

The Hammer is the pride of the USSR faction and is built like a tank, very slow in maneuverability but makes up for it with its superb shields and medium fire power capabilities. A good ship for attack and defense during heated battles when fully upgraded. Heaviest weight of all the ships, impacts how you are affected by the weight of an opposing asteroid.  
  
The Rapier belongs to the Chinese faction, is sleek in design with blazingly fast thrusters. Firepower is the weakest of all ships with a medium shield capability. A good ship to avoid attack and quick Iterium collection. The lightest ship which impacts how you are affected by the weight of an opposing asteroid.

## Controls



## 1.4 Scoring & Levels

Destroying NPC

Score + 1500

XP + 100

Destroying Player

Score + 500

XP + 50

Destroying Asteroid

Score + 50

XP + 10

Collecting Iterium

Score + 250

XP + 25

Points Bonus (Game Over scene)

At the end of each battle you will receive a points bonus calculated as (Iterium collected x 100) x level.

Leveling

Every 20000 XP gained.

## Upgrading

Upgrades are split into 3 different categories that can be upgraded twice each:

* Thrust
* Shields
* Firepower

You will need the required amount of Iterium indicated by each category slider on the upgrade screen. Iterium is dropped by destroying asteroids, there is a 20:1 chance of a drop.

# Unity Project

### Project requirements

* Unity version 2021.3.15f1
* Render Pipeline: URP 12.1.8
* Input System (new)
* UI Toolkit
* Timeline

### Folder Structure

|  |  |
| --- | --- |
| **Assets path sub folders** | **Description** |
| 3DModels | Ship and asteroid models |
| 3DModels/Bullets | Bullet models |
| 3DModels/Crystal | Iterium crystal model |
| 3DModels/Environment | Planet model |
| 3DModels/Materials | All model materials |
| 3DModels/Textures | All model textures |
| Animations | Animator/timeline files |
| Fonts | Font files |
| Images | Sprites for UI |
| Plugins | .wslib JavaScript plugin |
| Prefabs | All prefabs |
| RenderTextures | RenderTextures used to display models and shader graph progress bars |
| Scenes | All Unity scenes |
| ScriptableObjects | Created assets that hold the data |
| ScriptableObjects/Bullets | 3 levels of bullets per faction, Xoid bullet |
| ScriptableObjects/Factions | Factions – China, USA, USSR and Xoid |
| ScriptableObjects/Common | Common data and lists |
| ScriptableObjects/Common/Faction Types | Faction and NPC types |
| ScriptableObjects/Common/Lists | Lists of GameObjects and factions |
| ScriptableObjects/Common/Sound | Lists of sound effects and music |
| ScriptableObjects/Players | Player and AI data, holds the selected faction character SO |
| ScriptableObjects/Ships | Ship data, hold the ship bullet SO. |
| Scripts/ | All C# scripts |
| Scripts/Characters | Player controllers, leaderboard, bullets, iterium |
| Scripts/General | Utility script like rotating an object |
| Scripts/Managers | Game and sound managers, object pooling |
| Scripts/Save System | Save/load system |
| Scripts/ScriptableObjects | ScriptableObject scripts |
| Scripts/Spawning | Player/AI/NPC spawners and de-spawners |
| Scripts/UI | UI scene scripts (UI Toolkit) |
| Settings | URP render pipeline assets |
| Settings/Import | Assets import templates |
| Settings/Input System | Input system asset (New Input System) |
| Settings/Renderer | URP renderers |
| Settings/Sound | Sound/music mixer |
| ShaderGraphs | Effects for ship thrust, progress bars |
| Sounds | All audio files |
| Sounds/Effects | Sound effect files |
| Sounds/Music | Music files |
| UI | UI Toolkit files |
| UI/Documents | UI Toolkit Documents (UXML) |
| Styles | UI Toolkit styles (uss) |
| UI Toolkit | UI Toolkit themes |
| WebGL Templates | WebGL custom loader (HTML) |

### Classes & Assemblies

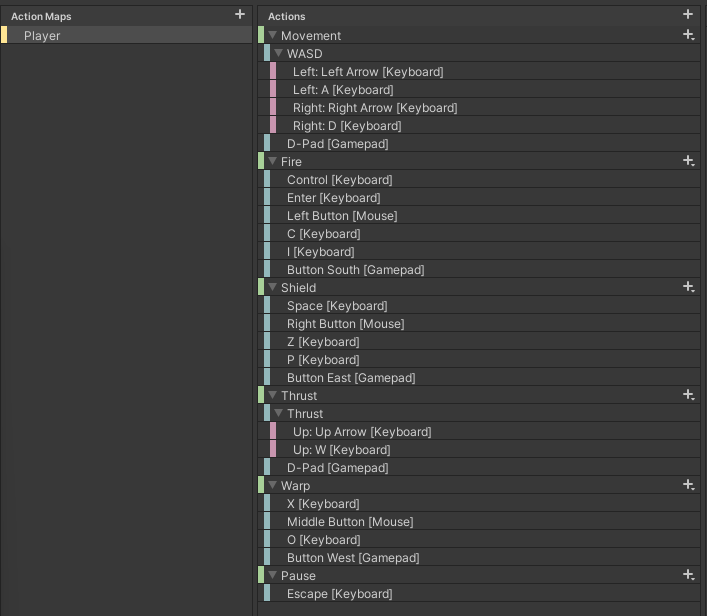
|  |  |  |  |
| --- | --- | --- | --- |
| **Class Name** | **Description** | **Derived From** | **Assembly** |
| AIController | Controls AI movement, shielding, firing etc. | MonoBehaviour | Assembly-CSharp |
| PlayerController | Controls player movement, shielding, firing etc. | MonoBehaviour | Assembly-CSharp |
| NPCController | Controls movement and firing for Xoid UFO | MonoBehaviour | Assembly-CSharp |
| BulletBase | Bullet base class, asteroid destruction, bullet explosions, iterium drop etc. | MonoBehaviour | Assembly-CSharp |
| Bullet | Player bullet, collision detection | BulletBase | Assembly-CSharp |
| BulletAI | AI bullet, collision detection | BulletBase | Assembly-CSharp |
| BulletNpc | Xoid bullet, collision detection | BulletBase | Assembly-CSharp |
| Iterium | Collision detection, score & xp | MonoBehaviour | Assembly-CSharp |
| SO\_Leaderboard | List<> of LeaderboardItem | ScriptableObject | Assembly-CSharp |
| LeaderboardItem | Leaderboard item fields |  | Assembly-CSharp |
| AsteroidPooling | Asteroid spawn/de-spawn object pooling | MonoBehaviour | Assembly-CSharp |
| BulletPooling | Bullet spawn/de-spawn object pooling | MonoBehaviour | Assembly-CSharp |
| ExplosionPooling | Explosion spawn/de-spawn object pooling | MonoBehaviour | Assembly-CSharp |
| Singleton | Singleton<T> base class | MonoBehaviour | Assembly-CSharp |
| GameManager | Player/AI objects, save/load, xp/level calculation, game settings, leaderboard, etc | Singleton<T> | Assembly-CSharp |
| SoundManager | Playing of sounds and music | Singleton<T> | Assembly-CSharp |
| InputManager | Device input for fire, thrust, rotate, shield etc. | MonoBehaviour | Assembly-CSharp |
| PlayMusic | Play sound effect by index from the SoundManager | MonoBehaviour | Assembly-CSharp |
| FileSaveHandler | Load/save serialized JSON data to a file, 3 files for player/AI/Leaderboard |  | Assembly-CSharp |
| SaveData | Player/AI save data fields |  | Assembly-CSharp |
| AISpawner | Spawn the AI prefab after a set time | MonoBehaviour | Assembly-CSharp |
| AsteroidSpawner | Spawn asteroid prefabs at set intervals | MonoBehaviour | Assembly-CSharp |
| DeSpawnAsteroid | Release asteroid to pool when leaving the screen | MonoBehaviour | Assembly-CSharp |
| DeSpawnExplosion | Release explosion after a set time | MonoBehaviour | Assembly-CSharp |
| GameOverSpawner | Spawn selected ship on the game over scene | MonoBehaviour | Assembly-CSharp |
| NPCSpawner | Spawn xoid UFO at set intervals | MonoBehaviour | Assembly-CSharp |
| PlayerSpawner | Spawn player prefab after a set time | MonoBehaviour | Assembly-CSharp |
| UI\_Button | Generic button script to load a scene | MonoBehaviour | Assembly-CSharp |
| UI\_FactionSelection | UI for the faction selection scene | MonoBehaviour | Assembly-CSharp |
| UI\_Game | UI for gameplay scene | MonoBehaviour | Assembly-CSharp |
| UI\_GameOver | UI for game over scene | MonoBehaviour | Assembly-CSharp |
| UI\_Leaderboard | UI for the leaderboard scene | MonoBehaviour | Assembly-CSharp |
| UI\_LoadUpgradeScene | Button script to load the upgrade scene | MonoBehaviour | Assembly-CSharp |
| UI\_Profile | UI for the player profile scene | MonoBehaviour | Assembly-CSharp |
| UI\_QuitGame | Button script to quit the application | MonoBehaviour | Assembly-CSharp |
| UI\_Settings | UI for the settings scene | MonoBehaviour | Assembly-CSharp |
| UI\_Tutorial | UI for the tutorial scene | MonoBehaviour | Assembly-CSharp |
| UI\_Upgrade | UI for the upgrade scenes | MonoBehaviour | Assembly-CSharp |
| CameraZoom | Zoom camera when starting the play scene | MonoBehaviour | General |
| PlaySFXDelay | Play a AudioSource with a delay | MonoBehaviour | General |
| Rotate | Rotate any GameObject, example asteroids | MonoBehaviour | General |
| SO\_Bullet | Bullet data, speed, firepower, prefab | ScriptableObject | SO |
| SO\_Faction | Faction data, SO\_Ship, factionId | ScriptableObject | SO |
| SO\_FactionList | List<SO\_Faction> of factions | ScriptableObject | SO |
| SO\_GameObjects | List<GameObject> asteroids, explosions, iterium | ScriptableObject | SO |
| SO\_NPC | NPC data, Xoid ship | ScriptableObject | SO |
| SO\_Player | Player/AI data, score, xp, health, lives, upgrades, SO\_Character | ScriptableObject | SO |
| SO\_SFX | List<AudioClip> sound effects | ScriptableObject | SO |
| SO\_Ship | Ship data, SO\_Bullet, speed, shield power, prefab, SO\_Types | ScriptableObject | SO |
| SO\_Types | Generic types or categories | ScriptableObject | SO |

### Input System

**InputManager class is the main class to use the InputSystem for all actions except the Pause action, this is actioned from the UI\_Game class.**

**Input System (Input Actions)**Single action map called “Player” with the following actions.

* Movement
* Fire
* Shield
* Thrust
* Warp
* Pause



### Save System

### Save files and methods

The system saves all data in 3 separate JSON files, namely Player.save, AI.save and Leaderboard.save. The file names can be changed on the GameManager inspector. The GameManager methods for saving and loading are…

* SaveGame()
* LoadGame()
* SaveLeaderboard()
* LoadLeaderboard()

### Path

The save path is set as Application.persistentDataPath by passing the value to the FileSaveHandler constructor when the new object is created inside the GameManager Start() method.

### Player data

The SaveData class, contains all player data fields that are saved. Two objects are created from this class inside the GameManager and passed to the FileSaveHandler for saving…

* saveData (Player)
* SaveDataAI (AI)

### Leaderboard data

The LeaderboardItem class contains the save data fields for each row in the leaderboard. ScriptableObject SO\_Leaderboard contains a List<LeaderboardItem> that stores the leaderboard rows, this list is passed to the FileSaveHandler for saving. The name of ScriptableObject and its List<> inside the GameManager is…

* Leaderboard (ScriptableObject)
* leaderboard.Leaderboard (List)

### File save handler

The FileSaveHandler class writes the data to text files in JSON format. The methods are as follows…

* Load(string fileName)
* Save(SaveData saveData, string fileName)
* LoadLeaderboard<T>(string fileName)
* SaveLeaderboard<T>(List<T> saveData, string fileName)

The Unity JsonUtility does not serialize a List<> to JSON if it is the top node, the FileSaveHandler contains a JsonHelper class as a wrapper to wrap the leaderboard List<> inside a “Items”:[] node. The List<> is cast to an array for serialization and cast back to a List<> before being returned by the FileSaveHandler. The class contains 3 methods to be used instead of the JsonUtility methods…

* FromJson<T>(string json)
* ToJson<T>(T[] array)
* ToJson<T>(T[] array, bool prettyPrint)

## ScriptableObjects

To reduce memory, aid testing and provide a designer friendly approach, data is stored inside ScriptableObject assets, this includes runtime data. Runtime data is loaded from the game save when the game starts, and updated data is saved at certain points back to the game save.

As summery of the main ScriptableObject game structure, the player asset holds the players faction, faction holds the ship, ship hold the bullet and bullet holds the bullet types.

### SO\_Leaderboard

The leaderboard asset contains a List<LeaderboardItem> that stores the high score rows. This list is passed to the FileSaveHandler during a game save.

### SO\_Player

The player asset holds all the data for the player/AI/Xoid – example health, level, upgrades and faction. The game can retrieve the players faction data (SO\_Faction faction).

### SO\_FactionList

Contains a list of factions (List<SO\_Faction> factions).

### SO\_Faction

Contains data about the faction ID and the game can retrieve the ship data (SO\_Ship).

### SO\_Ship

Contains data about thrust, turn speed, shield, prefab and the game can retrieve the ships bullet data (SO\_Bullet).

### SO\_Bullet

Contains bullet data like speed, firepower and list of 3 bullet prefabs (List<GameObject> bullet)

### SO\_SFX

Contains a list of sound effects or music to play (List<AudioClip> clips), has a method that returns a random clip (SelectRandomSound()).



## Managers

### Singleton

Base class for all singleton managers, namely GameManager and SoundManager classes. These 2 manager scripts are attached to the Managers prefab object and get instantiated from the main menu scene and live for the lifetime of the game.

### GameManager

The main class from where most game functions are managed. Exposed inspector properties are as follows…

|  |  |
| --- | --- |
| Property | Function |
| SaveFile | Name of the player game save file. |
| SaveFileAi | Name of the AI game save file. |
| SaveFileLeaderboard | Name of the leaderboard game save file. |
| Leaderboardsize | The number of rows the leaderboard can have. |
| player | Player Scriptable Object for the game to access all player and faction data. |
| factions | List of all factions to assign a faction to a player when he selects a ship. |
| deathRespawnRate | Delay in seconds before a player or AI is re-spawned after death. |
| xpLevelSteps | XP to gain for the player to gain a new level. |
| maxLevel | The maximum number of achievable levels. |
| freeShip | Score needed to gain a free live, this only happens once per battle. |
| aiPlayer | The AI ScriptableObject for the game to access all AI data and faction data. |
| aiPermadeath | Does the AI have lives or continuously re-spawns which is the default. |
| npcPlayer | Xoid ScriptableObject for the game to access all NPC data. |
| iterium | Iterium ScriptableObject contains the iterium prefab |
| iteriumChance | Chance for a splitting asteroid to drop a iterium crystal. Default = 20:1 |
| speedLevel1 | Number of iterium need to upgrade speed to level 2. |
| speedLevel2 | Number of iterium need to upgrade speed to level 3. |
| shieldLevel1 | Number of iterium need to upgrade shield to level 2. |
| shieldLevel2 | Number of iterium need to upgrade shield to level 3. |
| firepowerLevel1 | Number of iterium need to upgrade firepower to level 2. |
| firepowerLevel2 | Number of iterium need to upgrade firepower to level 3. |
| leaderboard | Leaderboard ScriptableObject containing a list of leaderboard rows. |

### SoundManager

All the sound effects and music is managed from this class. Exposed inspector properties are as follows…

|  |  |
| --- | --- |
| Property | Function |
| asteroidExplosion | ScriptableObject list of sound clips for asteroid explosions |
| shipExplosion | ScriptableObject list of sound clips for ship explosions |
| effects | ScriptableObject list of sound clips for sound effects |
| music | ScriptableObject list of sound clips for music |
| audioSourceNumber | Number of AudioSources to create dynamically for use by sound effects. Music has its own single dedicated AudioSource and is not included in this. |
| mixerMaster | Audio mixer group for the master |
| mixerMusic | Audio mixer group for music |
| mixerSfx | Audio mixer group for sound effects |