VOYAGE

Game Design Document

May 2019

Contents

Game Overview

Game Developers:

Irina Brigido Chan

Armando Perez

Anthony Reese

Dwayne Wilkes

Game Title: Voyage

Game Platform: Mac OS10.11+ and Windows 7 SP1+

Game Style: Third-Person Action Exploration

Rijeka Repository: CS596S19_3D_Game_Team04

URL: https://tinyurl.com/y482yhe3

High Concept

Voyage is a third-person Action Exploration game where the player is a space traveller on a mission to planet P-596 who crashed into this unknown world. Now, the player needs to rebuild his spaceship by exploring the environment to find the missing pieces and avoiding the different types of enemies encountered along the way!

Game Objective

The main objective of the game is to recover all lost spaceship pieces (three total) and successfully deliver them back to the main ship. There are different types of enemies spread around the environment, as well as pickup items that will replenish a small portion of the player's health. Once the player delivers all three pieces, the Credits scene is displayed. If the enemies deal enough damage to deplete the player's health completely, the Game Over screen is displayed.

Gameplay

Game Camera: The main camera is positioned behind the player and view angle can be adjusted by moving the mouse .

Game Controls:

- (WASD) and Space bar - Standard character movement and jump.
- (E) - Item pickup

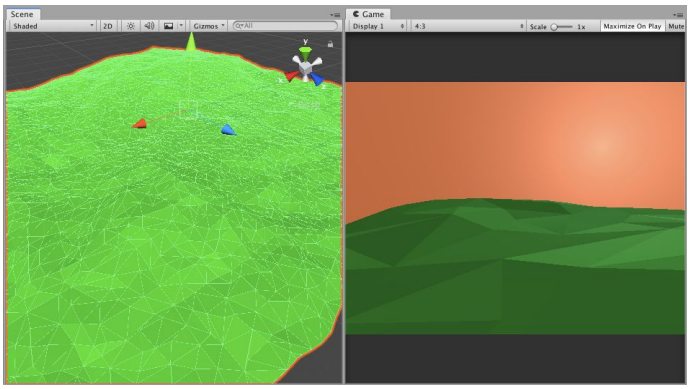
Level Design

Art Style

The art style of the game is “low poly” in order to deliver a distinct environment feel, provide easier game asset creation and modification, and lower computational demand for both development and gameplay.

Environment

The main level is the planet’s surface which consists of a large mesh with slight terrain variations. The player is set at the center of the terrain, along with the spaceship.



The environment is populated by different enriching assets, like rocks, grass, plants, trees and sticks.

Ship Mechanics

The ship pieces are instantiated at the beginning of the game across a set of available waypoints. The pieces are distributed randomly among the waypoints, and no more than 1 piece can be instantiated at the same location.

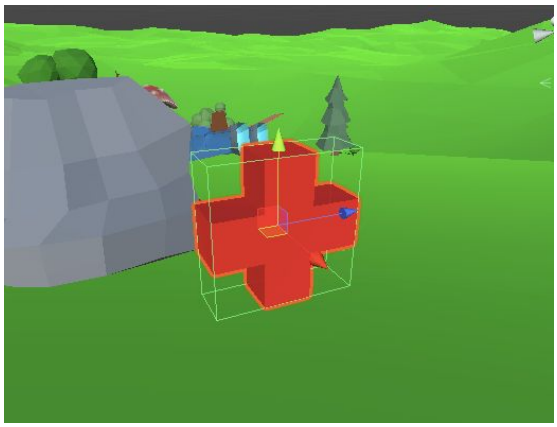
As the player successfully delivers a missing piece, the ship's model is updated to reflect its level of completion.

Ship with collected parts, Ship status bar (empty) and health bar

Spawn points and ship location

Pickup Items

There are health packs around the environment that replenish a small amount of the player's health. The value of each health pack is random (within a set range) and generated at runtime.



Note: The player's current health is never higher than the maximum health set in Script. Picking up a health pack while having full health has no effect on the player's health. Similarly, picking up a health pack that has a randomized value that would supersede the player's maximum health will only fill the current health value to the maximum allowed by Script. Refer to the Technical Design Document for more details on this mechanic.

AI Mechanics and Behavior

Here is a general description of each type of AI behavior, with further details included on the Technical Design Document:

1. *Ranged, static*: Enemy that has low or no mobility and shoots projectiles at the player's direction when detecting the player within a radius;
2. *Ranged, dynamic*: When detecting the player within a radius, the enemy will start following and shooting projectiles at the player's direction. Following behavior continues as long as the player stays within range;
3. *Melee, dynamic*: Follows the player when within range, and will perform a melee attack when reaching the player's current position. Following behavior continues as long as the player stays within range;

Game Assets and Script Bibliography

The following resources were used as reference for asset creation and game development during the prototype phase under Creative Commons[1] (CC) licenses.

3D Assets

- Spaceship: The spaceship and its pieces were modeled on Blender and on Unity Editor. The spaceship texture was painted using Adobe Photoshop CS5.1. The following tutorials were used as reference for modeling and UV wrapping:
    - https://tinyurl.com/y56zmbzy
    - https://tinyurl.com/y65w3bz6
- Skeleton enemy: https://tinyurl.com/y2l4jsng
- Environment elements: https://jaks.itch.io/lowpolyforestpack
- Mushroom enemy: https://tinyurl.com/y2cns943
- Mage enemy: https://tinyurl.com/y5s53dde

UI

- Font: *Cinzel*, by Natanael Gama: http://www.ndiscovered.com/
- Menu background image: screenshot obtained from the free available Asset Store Pack at https://tinyurl.com/y266m355
- Health bar and ship completion bar: https://tinyurl.com/y4rp6a54
- Crosshair from Kenney: https://www.kenney.nl/assets/crosshair-pack

Sounds

- Sound effects: https://tinyurl.com/ya9ejpmo

[1] https://creativecommons.org/ Please refer to individual asset for specific version.

Script

The Unity Scripting Reference[2] was used extensively, along with the following tutorials:

- Instantiating Projectiles:

https://learn.unity.com/tutorial/instantiate?projectId=5c8920b4edbc2a113b6bc26a#

- Item drop and pickup mechanics:

https://www.youtube.com/watch?v=IEV64CLZra8

- AI Behavior:

https://www.youtube.com/watch?v=gXpi1czz5NA&t=491s
https://www.youtube.com/watch?v=xppompv1DBg&

[2] https://docs.unity3d.com/ScriptReference/ version 2018.2