

A Price to Pay

Game Design Document

MJ's JRPGs

Micah Grande, Gerald Lappay, Jack Chen, Richard Estes, James
Zheng, Patrick Noosaeng

Game Development Team

Micah Grande: Enemy AI: Enemy Spawning, Implemented Credits scene, Created Prefabs, Quality-of-Life Improvements

Gerald Lappay: Audio Design: Voice Acting, Git Management

Jack Chen: Enemy AI: Flocking Behavior and Tracking, Implementing Spawn Point and Treasure Art Assets

Richard Estes: Audio Design: Music Composition

James Zheng: Procedural Maze Generation, Implementing Boid Art Assets

Patrick Noosaeng: Documentation: Game Design Document and Technical Design Document

Game Overview

Title: A Price to Pay

Platform: Windows, Mac OS, Linux

Genre: 1st Person Point-of-view, Procedurally-generated Maze, Survival Horror, Stealth

Target Audience: (6+) ESRB

Release Date: May 7, 2019

Publisher: MJ's JRPGs

Game Flow Summary: “A Price to Pay” is a 1st person, procedurally generated maze game where the player attempts to find treasure and bring it back to a spawn point, all while avoiding NPC enemies who rush to the player upon seeing them, and, upon contact, send the player all the way back to the beginning of the maze. When the player either completes the game or gets hit by an enemy, a new, different maze will be generated for the player to traverse.

Look and Feel: The game is meant to have a horror look and feel. This is expressed in the dark lighting, the spooky music, the haunted maze-like setting, and the floating, ghost-like enemies.

Gameplay and Mechanics

Game Progression: The player progresses the game by traversing the maze and avoiding the enemies on their way to the treasure and back.

Mission/Challenge Structure: There is technically only one level of the game to complete, though the maze is procedurally generated each time the player respawns, which occurs every time the player either gets hit by an enemy or brings the treasure to the spawn point. Other than this, there are no other distinct missions or challenges.

Puzzle Structure: There are no puzzles in the game.

Objectives: The player's sole objective is to obtain the treasure and bring it back to spawn without being hit by an enemy on the way.

Play Flow: The player traverses the maze at their own pace while trying to avoid the enemies on their way to the treasure and back.

Mechanics

Physics: There is no distinctly defined gravity in the game. The player and the enemies are unable to pass through walls, and nothing about the world itself distinctly impedes the character's overall movement. The enemies float through the maze as they wait for the player to fall within their line of sight.

Movement: The player moves using directional keys. 360 degree movement is possible by combining directional key inputs.

Objects: N/A

Actions: Besides movement, none.

Combat: N/A

Economy: N/A

Screen Flow:

(Open Game) -> Splash Screen -> Title Screen -> (Start Game [button]) -> Main Game Scene / Title Screen -> (Credits [button]) -> Credits Scene

Game Options: The player can toggle the instructions on the HUD on or off using the Space bar

Replaying: The game can be restarted upon either getting the treasure and returning to the spawn, or hitting an enemy

Saving: N/A

Cheats: N/A

Easter Eggs: 30 seconds after the credits are completed, a picture of Professor Price (the one used on the boids) appears

Story, Setting, and Character

Story and Narrative: The player is trapped in a haunted maze-like version of EBA 410 which is guarded by the Price ghosts who seek to ensure that the player remains trapped forever. There is, however, one hope for the player: finding the passing grade somewhere in the classroom and bringing it back to the spawn point. The player must

traverse the maze, find the passing grade, and bring it back before they are caught by the Price ghosts.

Plot Elements: None besides the above described

Cut Scenes: N/A

Game World: The world is designed to be horror-themed, based around the procedurally-generated maze (to give the player a sense of uncertainty traversing the maze) and the ghost enemies that chase the player (which will give the player a sense of urgency as they attempt to find the treasure and bring it back to spawn).

Areas: There is one distinct area: the maze that the player traverses. While the maze is procedurally generated each time, the physical characteristics (the color and overall appearance of the maze) do not change with each maze generation.

Characters:

- **Player:** Has no real appearance, personality, animations, abilities (besides moving), back story (none as it directly relates to the gameplay), or relevance to the story (none as it directly relates to the gameplay). The relationship between the ghosts and the player is that the ghosts wish to

prevent the player from finding the treasure and bringing it back to the spawn point

- **Player Metrics:**
 - **Speed:** None specified
 - **Max Health:** One contact with the enemy
 - **Spawn Point:** Always spawns in the bottom left of the map
- **Price Ghosts:** Have no personality (as it relates to the gameplay), no animations (besides floating and moving), no back story (none as it directly relates to the gameplay), and no relevance to the story (none as it directly relates to the gameplay). The ghosts will move directly towards the player upon seeing them. In addition, they will choose a random line out of a potential six to play upon seeing the player. These enemies will respawn the player at the initial start point upon touching them.



- **Enemy Metrics:**
 - **Speed:** Slower than player movement until player is spotted, then becomes the same speed as player
 - **Spawning:** Spawns in random places in the maze

Levels

Levels: There is one (procedurally-generated) maze level, and the objective is to find the treasure located somewhere in the maze and bring it back while avoiding the enemies along the way.

As far as the maze itself is concerned, the maze itself has low lighting, brown, wooden-like walls, and gray, concrete-like floors and ceilings. The critical path to the goal of the level changes with each procedural generation, as the shortest path to the treasure will change as each new map is generated.

The important encounters involve the enemies the player is trying to avoid.

Training Level: N/A

Interface

Visual System: The HUD will have instructions on how to play the game, the player's objective, the mini-map, and how to toggle the instructions and mini-map on and off with hot-keys.



Camera Model: First Person View. The player will be given a “flashlight” that will brighten a circular area in front of the camera

Control System: The player controls their character using the directional keys or the W, A, S, D keys, and maneuvers the camera using the mouse.

Audio, Music, Sound Effects: Audio will play when the reapers spot the player, and the music will be

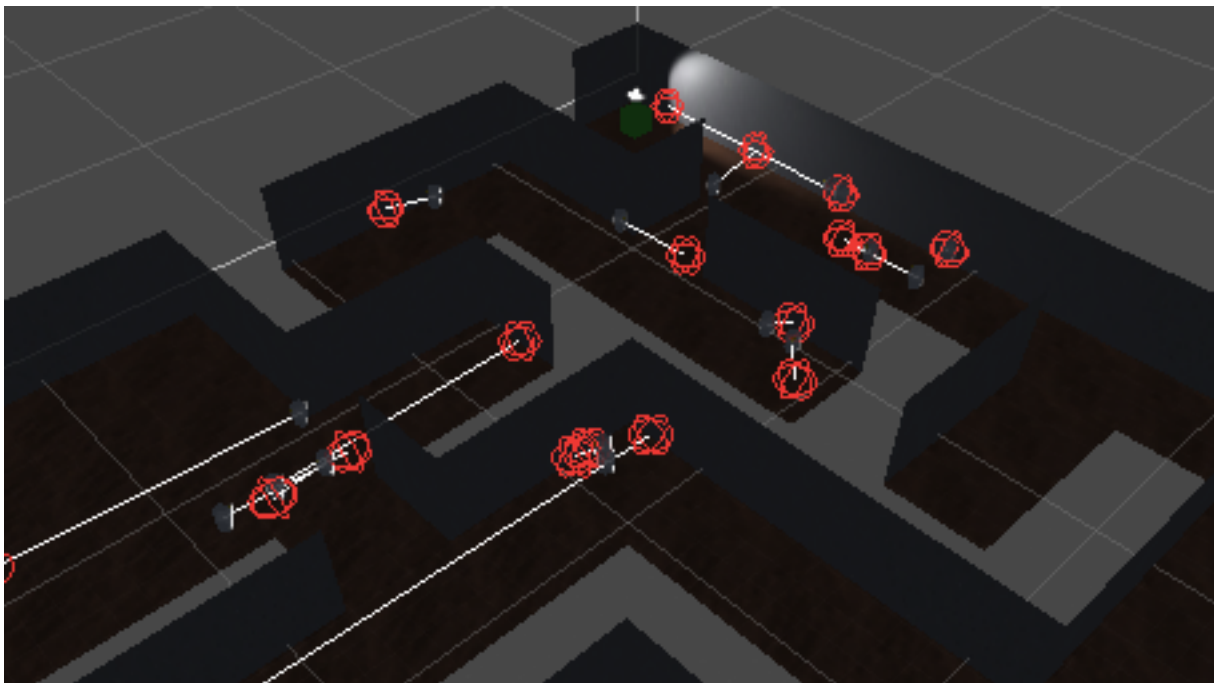
horror-themed and will be played in the Main Menu, the main game, and the credits.

Help System: A mini-map on the top left of the screen and instructions on how to play the game in the top middle are on by default unless the player toggles off this help system.



Artificial Intelligence

Opponent and Enemy AI: If the ghosts come into range or spot the player using the **ray-cast features** in Unity, they will adjust their behavior to move towards the player using the shortest path to the player.



Non-combat and Friendly Characters: N/A

Support AI:

- Flocking AI will be used to make the ghosts continuously follow the player upon seeing them
- The ghosts actively avoid colliding with any walls due to collision detection and avoidance AI

Unique Selling Points

- Procedurally generated content
- Flocking AI enemy behavior
- For all ages
- Multi Platform
- Unique Premise

Platform Minimum Requirements

- Runs on Windows, MacOS, and Linux platforms
- 2 GB RAM minimum
- Graphic Card: DX9 (shader model 2.0) capabilities; generally anything made since 2004 should work

MVP (Minimum Viable Product)

- One playable character
- One scene / level to play (Procedurally generated)

- Built for Windows, MacOS, and Linux

Music and Audio files

Audio Files

- Price1.wav - One of the randomly selected voice lines that plays when the player is spotted by the boid
- Price2.wav - One of the randomly selected voice lines that plays when the player is spotted by the boid
- Price3.wav - One of the randomly selected voice lines that plays when the player is spotted by the boid
- Price4.wav - One of the randomly selected voice lines that plays when the player is spotted by the boid
- Price5.wav - One of the randomly selected voice lines that plays when the player is spotted by the boid
- Price6.wav - One of the randomly selected voice lines that plays when the player is spotted by the boid

- Price7.wav - One of the randomly selected voice lines that plays when the player is spotted by the boid
- PriceOut.wav - Was supposed to play when the player was defeated, but ended up being too long and was unused in the final product.

Music Files

- ambience.mp3 - Horror-themed soundtrack. Plays in the main menu.
- big_spook.mp3 - Horror-themed soundtrack. Plays in the main game
- end_credits.mp3 - Plays in the end credits scene.

Wishlist

- Implementing higher quality 3D models for the enemies
- Improving overall quality of life
- Making the “escape” sequence more difficult by implementing a large boss-like enemy that the other ghosts would flock around as it chased the player

Bibliography

Assets Used from:

- 3D boids and maze appearance were custom designed
- Music was custom made
- Voice Acting was custom made
- Picture of Professor Stephen Price: <http://www.cs.sdsu.edu/faculty-and-staff/>
- Floor Segment - From Unity Asset Store: <https://assetstore.unity.com/packages/3d/environments/dungeons/floor-segment-20330>
- A Piece of Nature - Treasure Chest - From Unity Asset Store: <https://assetstore.unity.com/packages/3d/environments/fantasy/a-piece-of-nature-40538>

Scripts:

- CloseGame.cs - Written by our team
- Credits.cs - Written by our team
- Exit.cs - Unused

- FpsMovement.cs - Sourced from MIT open source, written by Joseph Hocking (2017)
- GameController.cs - <https://www.raywenderlich.com/82-procedural-generation-of-mazes-with-unity> and edited by our team to fit the needs of the game
- InstructionsToggle.cs - Written by our team
- LevelController.cs - Written by our team
- MazeConstructor.cs - <https://www.raywenderlich.com/82-procedural-generation-of-mazes-with-unity>
- MazeDataGenerator.cs - <https://www.raywenderlich.com/82-procedural-generation-of-mazes-with-unity>
- MazeMeshGenerator.cs - <https://www.raywenderlich.com/82-procedural-generation-of-mazes-with-unity>
- SpawnController.cs - Written by our team
- StalkerMovement.cs - Written by our team and used various online sources to help write the script

- StalkerSpawn.cs - Written by our team
- ToMenu.cs - Written by our team
- TriggerEventRouter.cs - Sourced from MIT open source, written by Joseph Hocking (2017)