Retro Game Jam GDD Template

Game Title:

Pac-Man: Beyond the Maze

Team Name:

Baomeide

Team Members:

List all members and their student IDs:

- Shuchen Yuan (82489881)
- Jinxi Hu (48528608)
- Lucas Xu (60844602)
- Ming Xu (84642180)

---

1. Core Concept

Briefly describe the game.

- What classic retro game are you taking inspiration from? (**You can not do Breakout since it will be a topic for a future studio assignment!)**
- What unique twist are you adding to make it different? (e.g., new mechanics, modernized visuals, additional features)

Example: *Our game is inspired by Breakout, but instead of a paddle at the bottom, the player controls a magnet that can attract or repel the ball, adding a physics-based challenge.*

We gain inspiration from the ***Pac-Man*** - a classic arcade game released in 1980 by *Namco.* The original game takes place in a flat and 2D maze, where the player controls Pac-Man to eat pellets, and at the same time escaping from the hunting of the ghost guardian. The goal is to clear all the pellets and try to survive.

One of the twists on this retro game is to bring Pac-Man into a fully 3D environment. Instead of moving through a simple, top-down maze, players will attempt to survive in a multi-level labyrinth with stairs, ramps, and tunnels. In addition, we introduced new skill mechanisms to complement the 3D space. These include effects that can make walls temporarily transparent, space move instantaneously, and become larger and smaller.

Key differences from the original game include:

- ***3D Maze Structure:*** Players move through a multi-level maze, where paths may lead up, down, and around.
- ***New Power-Ups & Effects:*** Pac-Man can gain different effects throughout the maze, altering gameplay in unique ways. Some of the might-be-included:
  - **Speed Burst** – It grants a temporary speed boost so that Pac-Man could escape danger or collect pellets faster.
  - **X-Ray Vision -** This effect allows Pac-Man to see through walls to discover potential hidden paths, caches, or ghosts close to them.
  - **Size up effect -** The character can be made larger or smaller to pass through some special area
  - **Teleportation -** When encountering coins that are not easily found, you can use this buff to teleport the player near random coins

2. Core Gameplay

Game Loop:

Describe the basic actions the player takes and how they interact with the game.

- What is the player doing repeatedly? (e.g., jumping, shooting, dodging, collecting items)
- How does the game provide feedback? (e.g., visual effects, sound effects, UI updates)

Example:
The player moves a spaceship left and right to avoid asteroids and shoot enemies. Destroying enemies grants points and occasional power-ups, making the ship stronger.

Movement: The player can control Pac-Man to move forward, backward, left, and right, while jumping some obstacles

Angle of view: You can switch between the first time and the third Angle of view, hold down the right button and drag to look around and see the surrounding environment

Item Activation: Pac-Man can activate different items after collecting them, with special abilities (such as acceleration, size change, teleportation, or perspective) for a short period of time.

Evasion and pursuit: The player must constantly evade the pursuit of the ghost guards to avoid being captured. Players can fight back against Ghost guards when they get invincible buffs and can chase them for short periods of time

Sound feedback: Pac-Man has a certain sound when collecting beans or activating items, and a corresponding background music is added when it is captured by a ghost guard or when it fights back

UI update: The interface displays the progress of bean collection and the effect of currently active items. A small map is placed in the upper right corner to give the player a clearer understanding of their current location. (Gives hints when Ghost guards are approaching)

3. Player Controls - Raven

List the controls clearly and simply.

Example:

- Keyboard (or Controller) Inputs:
  - o Move: Arrow Keys / A & D
  - o Jump: Spacebar
  - o Attack/Shoot: Left Mouse Button / Ctrl
  - o Special Ability: Shift / Right Mouse Button

Keyboard Inputs:

o Move: W move forward, S to turn back

o Rotate Move direction: A to left/D to right

o Dash: Direction key + E

o Jump: space

o Open menu: P

o Show/hide the mouse: ESC

4. Level & Progression

Game Structure:

- How many levels are in the game? (e.g., 5 levels, 10 waves, endless mode)
- Are levels handcrafted or procedurally generated?

Progression System:

- How does difficulty increase? (e.g., faster enemies, more obstacles, limited time)
- Are there power-ups or upgrades? (e.g., speed boost, double jump, new weapons)

Example:
Each level introduces new enemy types and increases their speed. The player can collect shield power-ups for temporary invincibility. There will be 3 levels in total and the player has to win in all 3 to finish the game

---

A map is used in the game, which consists of three stages (including two mazes and an open scene). Each level is handmade. There are 30 coins distributed on the map, and players only need to collect 20 of them to win

The difficulty of the three stages is different.

The simplest maze is made of low walls that the player can easily climb over. This map includes freeze buffs and sprint buffs.

The second maze is made of high walls that the player cannot climb. But will unlock new transparent wall buffs, and the player can better plan the path through this buff.

The open world is made up of a 5 floor house and 3 huts with a complex interior structure and is guarded by many ghosts. This part will provide a teleportation buff, which can be used to quickly find additional coins when players cannot find it, or easily escape when they are blocked by a tracking ghost.

5. Scoring & Win/Loss Conditions - Annie

Winning:

- How does a player complete the game? (e.g., reach the final level, defeat a boss, survive a time limit)

Losing:

- What causes a game over? (e.g., losing all lives, running out of time, missing too many targets)

Score System:

- How is progress tracked? (e.g., points, time, collected items)
- Are there multipliers, streaks, or bonuses?

Winning:

• The player controls Pac-Man as he moves around the map and earns points through collecting coins. Win by collecting 20 out of 30 coins before the three life points are exhausted.

Losing:

• If the player does not run out of lives before collecting 20 coins (one life per capture by a ghost), it is considered a failure

Score System:

• Players gain points by controlling Pac-Man to collect coins, each coin being worth one point

---

6. Timeline & Milestones

Week 1: Core Mechanics & Gameplay Elements

Example:

- Set up a basic Unity project <Name (of student assigned to the task)>
- Implement player input and core mechanics (moving, jumping, shooting, etc.) <Name>
- Implement scoring system <Name>
- Create a rough UI (score display, health, etc.) <Name>

====

Week 2: Polish & Finalization

Example:

- Design and implement levels/waves <Name>
- Add particle effects, sound effects, and other polish <Name>
- Improve UI and menus <Name>
- Package the final build <Name>

Week 1: Core Mechanics & Gameplay Elements

• Implement ghost movement and pursuit logic as well as rough UI maps (Lucas Xu)

- Implement rough UI interface (including start screen, game introduction screen, menu screen) (Lucas Xu)
- Implement partial buffs (including transparent walls, teleportation, freezing effects) (Lucas Xu)
- Implement partial buff (increase size, decrease size, dash) and the duration of the buff (Ming Xu)
- Implement coins scoring logic and create a scoreboard UI (Ming Xu)

- Implement rough **UI interface**s (including start screen, game introduction screen, menu screen) (Iverson Yuan) - COMPLETED
- Implement player health system with **visual feedback** (Iverson Yuan) - COMPLETED
- Implement **UI animation system** with unscaled time support (Iverson Yuan) - COMPLETED
- Implement **pause and game-completion-related menus** functionality with multiple options (Iverson Yuan) – COMPLETED
- Finished building and designing the scene (Iverson Yuan) – COMPLETED
- Adjust the inheritance order of scene models (Jinxi Hu) - COMPLETED
- Implement a basic min map and its UI (Jinxi Hu) - COMPLETED
- Implement a switch between different minimaps based on the player's position in the map (Jinxi Hu) - COMPLETED

Week 2: Polishing & Optimizing

- Final scene layout (Lucas Xu)
- Optimize ghost and coin UI (Lucas Xu)
- Optimize bake path for navmesh surface (Lucas Xu)
- Choose a visual image for the player and combine it with animation effects (Ming Xu)
- Export and upload the game version for user testing (Ming Xu)

- Optimize UI transitions between scenes (Iverson Yuan) - COMPLETED
- Implement Game Over screen (Iverson Yuan) - COMPLETED
- Create and integrate audio management system for different game states (Iverson Yuan) - COMPLETED

- Polish UI visual effects and animations (Iverson Yuan) - COMPLETED
- Fix collision detection issues between player and ghosts (Iverson) - COMPLETED
- Implement the enemy paused after enemy touch the player (Jinxi Hu) - COMPLETED
- Advance the min map UI to make it looks better and provide more information (Jinxi Hu) - COMPLETED

---

7. Assets

Models & Art:

- **Gold UI optimization:** Area of Effect Spell FREE | Spells | Unity Asset Store
- **Ghost UI:** Demon Tree | 3D Creatures | Unity Asset Store Little Ghost (Free) | 3D Characters | Unity Asset Store
- **Main Scene:** POLYGON HORROR MANSION
- **HP System UI:** Heart Icon
- **buff UI:** LOW POLY - Magical Potions Pack | 3D Props | Unity Asset Store
- **buff effect UI:** Magic Effects FREE | Spells | Unity Asset Store
- **buff light band in Invisible wall:** Volumetric Lines | Particles/Effects | Unity Asset Store
- **the player character:** Lowpoly Cowboy RIO V1.1
- **the player animation:** Basic Motions FREE

Most of the **Models and Art** in our game are **open-sourced** and will come from ***Unity Asset Store***

- **Unity Asset Store:** https://assetstore.unity.com/zh-CN?srsltid=AfmBOoo5FdYslJKp166v4Mxd-Uq20Tjk8p7vi1xDMYrpYVKWJFcX-iWZ

Sound & Music:

The source of the audio material is similar to the models and art, most from Unity Asset Store and some other platforms like Freesound.

- **Unity Asset Store:** https://assetstore.unity.com/zh-CN?srsltid=AfmBOoo5FdYslJKp166v4Mxd-Uq20Tjk8p7vi1xDMYrpYVKWJFcX-iWZ
- Pre Game Music: 高梨康治 - Dance of Death

- In Game Music: Ludvig Forssell - Once, There Was an Explosion
- **Freesound:** https://freesound.org/

UI:

Similarly, UI components are also open-sourced internet material. Most of them comes from the Unity Asset Store and OpenGameArt.

- **Unity Asset Store:** https://assetstore.unity.com/zh-CN?srsltid=AfmBOoo5FdYsIJKp166v4Mxd-Uq20Tjk8p7vi1xDMYrpYVKWJFcX-iWZ
- **OpenGameArt:** https://opengameart.org/