Cal Poly Pomona
CS470 Game Development
Spring Quarter
2017

****

Documentation type: Architecture Specification

Project: The Farting Panda
Team: BaDkINgZ
Members: Mahmudur Khan,
Nathaniel Krueper,
Benjamin Krueper,
Chu Yiu Cheung,
Ian Stodart
Professor: Tony Diaz

****

BaDkINgZ
STUDIO

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of *The Farting Panda* 2D platformer game, using a number of different views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the game.

1.2 Scope

This Software Architecture Document provides an architectural overview of *The Farting Panda* 2D platformer game. *The Farting panda* is being developed by Team BaDKINgZ for CS470 Quarter Project.
This Document is written on Google document via Google Drive and the view models are produced on Draw.io.

1.3 Definitions, Acronyms, Abbreviations

TFP - *The Farting Panda*, game title

2. Architectural Representation

This document presents the architecture as a series of views under the Unified Modelling Language (UML) standard: use case view, and object model view.

3. Architectural Goals and Constraints

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

1. TFP is developed solely on the Unity 2D engine, therefore the functionalities and features completely depend on the capabilities of Unity.

2. The goal is to create a architectural design that makes a 2D platformer game possible.

4. Use-Case View

A description of the Use--Case View of TFP. The Use-Case View is important input to the selection of the set of scenarios and/or use cases that are the focus of an iteration. It describes the set of scenarios and/or use cases that represent the central functionality of the game.

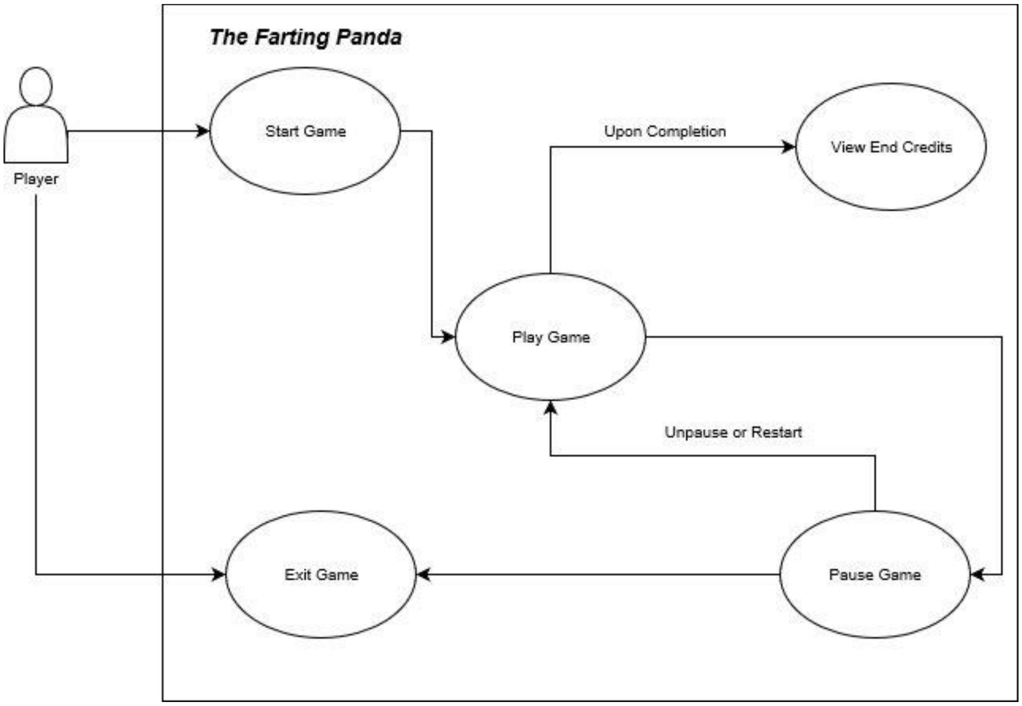
The TFP use cases are:

-

  Start Game
  Exit Game
  Play Game
  Pause Game
  View End Credits

These use cases are initiated by the player, except view end credits which is only available upon game completion.

4.1 Architecturally- Significant Use Cases



4.1.1 Start Game

Brief Description:This use case allows a player to start a new game on the start screen.

4.1.2 Exit Game

Brief Description: This use case allows a player to exit the game. Exit Game is available on start screen and pause menu in gameplay.

4.1.3 Play Game

Brief Description: This use case allows a player to play the game. A player can start/continue playing the game either by clicking start game on start screen, unpause after pausing the game, or choosing restart level on pause menu.

4.1.4 Pause Game

Brief Description: This use case allows a player to pause the game during gameplay. As a result, a pause menu is displayed.

4.1.5 View End Credits

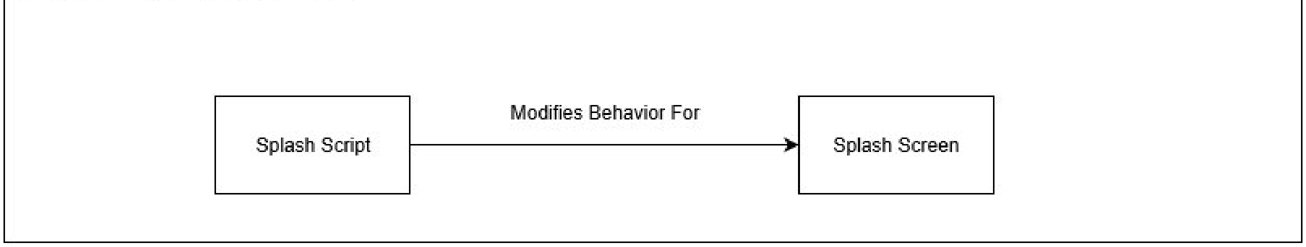
Brief Description: This use case allows a player to view all credits related to the game production. End Credits screen cannot be initiated by the player, it is displayed only when the player finishes the game (reaching the end goal).

5. Object Model View

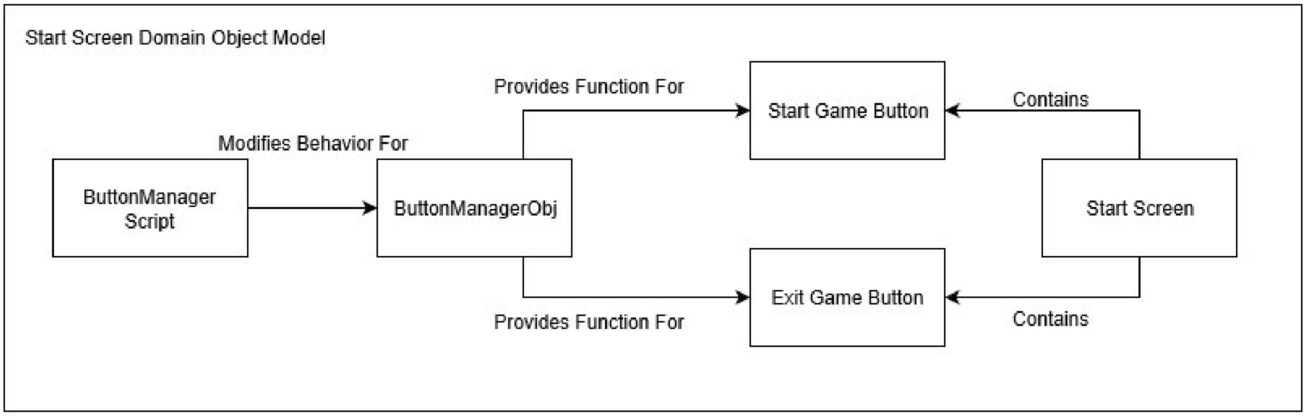
A description of the Object Model View of TFP. The Object Model View is important mapping of relationships between GameObjects and C# scripts. It describes how the game objects interact with their C# scripts and with each other.

5.1 Object Model Diagram

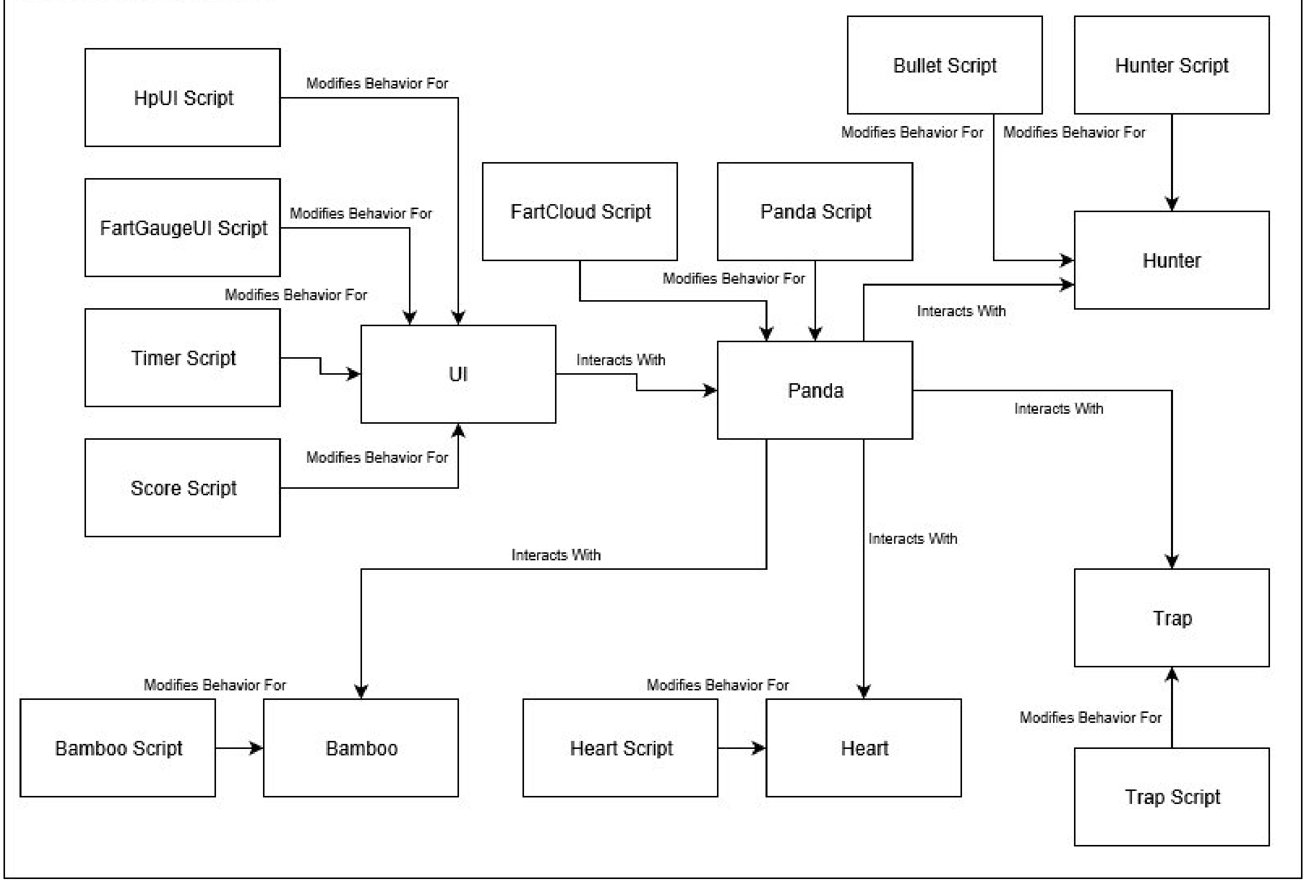
Splash Screen Domain Object Model



Start Screen Domain Object Model



Level1 Domain Object Model



5.1.1 Splash Screen Domain Object Model

Splash Script - Defines the time the splash screen is displayed for, then proceed the game state to start screen.
Splash Screen - Contains our team logo centered on a black background.

5.1.2 Start Screen Domain Object Model

ButtonManager Script - Defines two methods (StartGame and ExitGame).
ButtonManagerObj - An empty GameObject for ButtonManager Script to attach to.
Start Game Button - Button to start game, activates StartGame Methods on click.
Exit Game Button - Button to exit game, activates ExitGame Methods on click.

5.1.3 Level1 Domain Object Model

HpUI Script - Defines how hp bar is updated upon hp increase or decrease.
FartGaugeUI Script - Defines how fart gauge is updated upon fart capacity increase or decrease.
Timer Script - Defines how the game keep tracks of time.
Score Script - Defines how the Score UI is updated.
Heart Script - Defines the heart’s animation.
Heart - GameObject that displays a heart.
Bamboo Script - Defines the bamboo’s animation.
Bamboo - GameObject that displays a bamboo.
UI - GameObject that displays hp bar, fart gauge, timer, and score.
FartCloud Script - Defines the animation when Panda farts.
Panda Script - Defines all attributes of the panda and their behaviors.
Panda - GameObject that displays the panda.
Hunter Script - Defines all attributes of hunters and their behaviors.
Hunter - GameObject that displays a hunter.
Trap Script - Defines all attributes of traps and their behaviors.
Trap - GameObject that displays a trap.

6. Size and Performance

TFP is developed under the Unity 2D engine for desktop environment, therefore it automatically conforms to the system requirement given by Unity:

OS: Windows XP SP2+, Mac OS X 10.8+, Ubuntu 12.04+, SteamOS+.

Graphics card: DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities.

CPU: SSE2 instruction set support.