Jean Cacciattolo

Game Programming Synoptic

Documentation (Concept Doc & Scope Check, GDD, CRC and UML)

Concept Document

Game Concept

- Core Idea: "Oh No! My Figolla" is a casual interactive game where players follow a series of dynamic instructions to perform tasks and achieve the highest score.

Gameplay Mechanics

- Instructions: Players receive on-screen instructions to perform actions such as clicking buttons, rotating dials, and toggling switches.
- Scoring: Points are awarded based on the accuracy and speed of performing tasks. The game tracks and displays high scores.
- Difficulty: The difficulty increases as the game progresses, with tasks becoming more challenging.

Target Audience

Demographics: Casual gamers, puzzle enthusiasts, and players looking for quick, engaging gameplay.

Platforms: Primarily PC, with potential expansions to mobile platforms.

Scope Check

Features

Core Features:

- On-screen instructions for tasks.
- Interactive elements like buttons, dials, and switches.
- Scoring system with high score tracking.
- Sound effects for feedback.

Out-of-Scope Features

- Advanced AI: Complex AI behavior beyond simple interactions.
- Multiplayer: Online or local multiplayer features.
- Extensive Storyline: Deep narrative elements beyond basic instructions.

Timeline

- Development Phases:
    - Prototype: Basic gameplay mechanics and UI elements.
    - Alpha: Core features implemented, basic testing.
    - Beta: Feature-complete, extensive testing and bug fixing.
    - Release: Final version, polish, and optimization.

Game Design Document (GDD)

1. Game Overview

- Game Title: Oh No! My Figolla
- Genre: Interactive Puzzle Game
- Platform: PC (Unity Engine)
- Target Audience: Casual gamers

2. Concept

- Game Concept: "Oh No! My Figolla" is a game where players follow on-screen instructions to perform various tasks like clicking buttons, rotating dials, toggling switches, and more to achieve the highest score. The game is designed to be engaging and test the player's reaction time and accuracy.

3. Story and Characters

- Story: The game doesn't follow a traditional narrative but revolves around the player completing a series of tasks to keep the Oven, satisfied.
- Characters:
    - Oven: The main character whose satisfaction depends on the player's actions.

4. Gameplay Mechanics

- Tasks: Players must follow on-screen instructions and perform tasks like clicking, rotating, and toggling.
- Scoring: Points are awarded for correctly performing tasks. The game tracks high scores.
- Challenges: The tasks become more difficult as the player progresses.

5. User Interface

- Main Menu: Start game, view high scores.
- In-game UI: Score display, timer, action buttons.

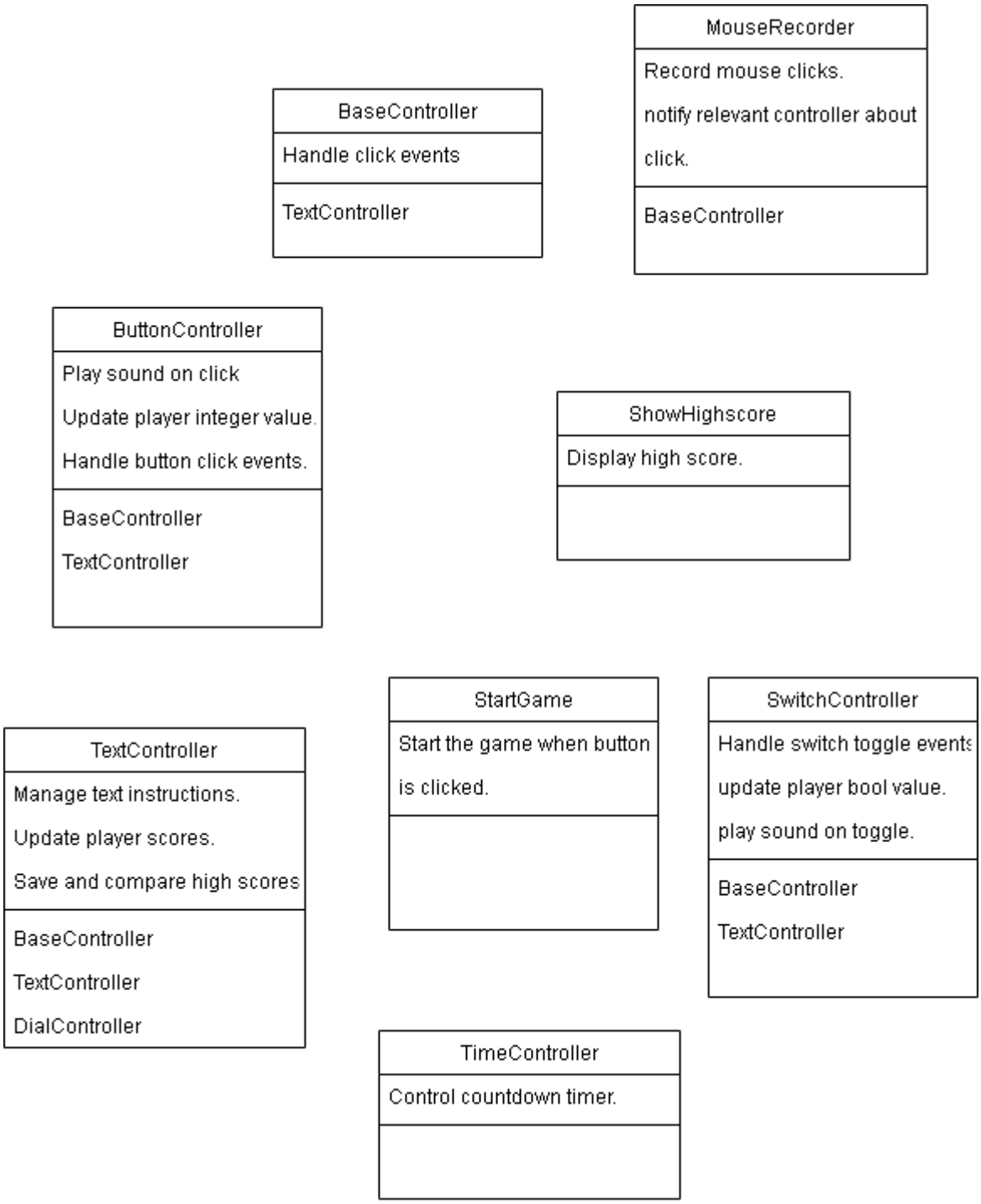
6. Art and Sound

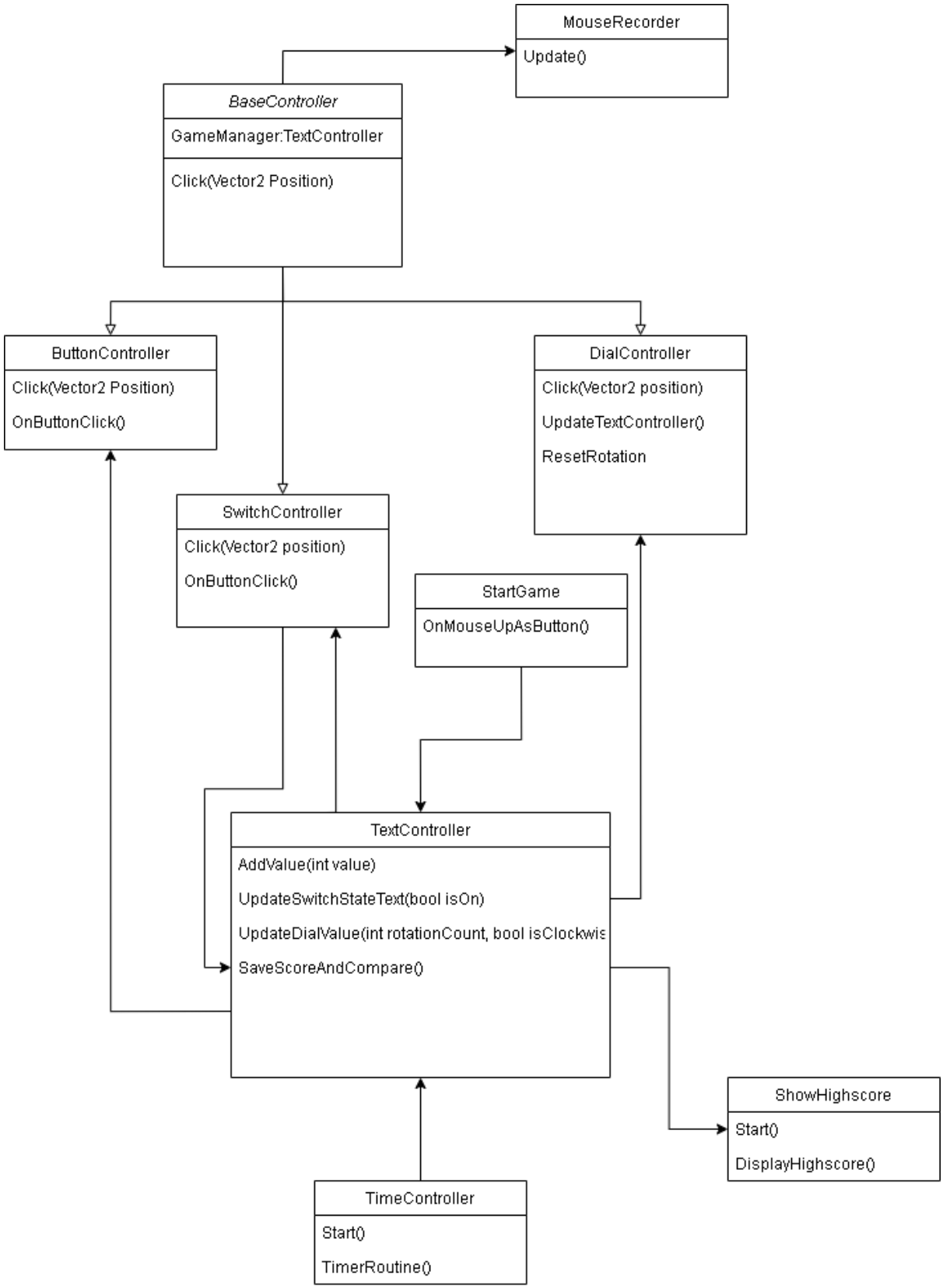
- Art Style: Cartoonish and colorful to keep the game visually engaging.
- Sound Effects: Feedback sounds for actions like clicking and toggling. Background music to enhance the gaming experience.

7. Technical Requirements

- Engine: Unity 2020.3 or later
- Libraries: TextMeshPro for UI text elements
- Assets: Custom sprites for buttons, dials, and switches.

UML and CRC




MouseRecorder
Update()
BaseController
GameManager:TextController
Click(Vector2 Position)
ButtonController
Click(Vector2 Position)
OnButtonClick()
DialController
Click(Vector2 position)
UpdateTextController()
ResetRotation
SwitchController
Click(Vector2 position)
OnButtonClick()
StartGame
OnMouseUpAsButton()
TextController
AddValue(int value)
UpdateSwitchStateText(bool isOn)
UpdateDialValue(int rotationCount, bool isClockwis
SaveScoreAndCompare()
ShowHighscore
Start()
DisplayHighscore()
TimeController
Start()
TimerRoutine()