**Game Development Club Minigames Collection Documentation**

By Jose Pacio

Introduction

The GDC Minigames Collection is a compilation of minigames created by members of the UW Game Development Club. This project is intended to act as a way for newcomers to Game Development to practice Unity and as something the whole club can easily do together.

Please note that the infrastructure is being constantly worked on. So if you have any suggestions on how to make it better, or if you run into any issues, please let us know. Also please let us know if anything is confusing or doesn’t make sense in these instructions.

Overall Code Infrastructure

This may be helpful to understand the way scripts are laid out. If you are just making minigames, then the only thing you really need to worry about is the “minigame” script.

Game Manager

* Loads minigames in
* Manages overall state of the game
  + Tracking current round and lives

Game Manager GUI

* The visual display of the Game Manager
* Handles animation transitions between and into minigames

Lives GUI

* The visual display of how many lives the player has

Minigame

* The script that handles minigame logic

Timer GUI

* The visual display for the minigame countdown timer

Playing The Game

1. Go to Assets > MinigameRunner
2. Make sure in the Game Manager Script in the DormGuiManager debugging level is false
3. Hit Start

Creating Minigames

Initial Setup

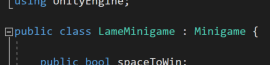
1. Create a new folder in Assets>Minigames. This is where you will store all your assets for your minigame
2. Delete the Current Main Camera
3. Save your scene
4. Add your minigame to build
   1. File>Build Settings… then hitting “Add Open Scenes”
5. Add the DormGUIGameManager prefab to your scene, which can be found in Assets>Game nfrastructure
6. Add the MinigameManager prefab to your scene, which can be found in Assets>GameInfrastructure

-or-

5. Copy the gameObjects from one of the example minigames

Creating new Minigame

1. Create a new script for your minigame, and add it to the MinigameManager Object
2. Remove the current minigame script (currently “Lame Minigame”)
3. Open up your new script
4. Have your minigame script extend Minigame



Minigame API

Here are some important functions and variables you get when you extend the Minigame class.

Important Functions

public override void gameStart(difficulty difficulty) – sets up specific logic and variables for this minigame. Runs once when the minigame starts.

public override void gameUpdate() – logic that runs once on every game update call.

public override void lose() – ends the minigame with a loss

public override void win() – ends the minigame with a win

Variables

public float time – the current time till the minigame ends (also doubles as the starting time if the minigame is at easy difficulty)

public float mediumTime – starting amount of time when it is medium difficulty

public float hardTime – starting amount of time when it is medium difficulty

public float loseWhenNoTime – the win/loss condition if the player runs the timer to zero

public string actionDescription- the descrpition in the GUI when the player loads this minigame

public GameObject minigameContainer- a container that holds all of the gameobjects of the minigame

other settings

public enum difficulty { EASY, MEDIUM, HARD} – this is a variable that dictates the difficulty of a minigame. You can choose to make minigames behave differently/get harder depending on the setting, such as changing the amount of time the player has.

**Notes**

**!!!Make sure to parent all of your active GameObjects into a GameObject that will be the designated minigameContainer. If you don’t, a minigame may start before the GUI has actually loaded it in!!!**

Creating GameManagerGUIs

This section is optional for those who want to create their own

custom Game Transition GUI stuff. There is currently a default GUI, so you don’t really need to worry about it, but if you want to it’ll be here… at some point

-under construction-