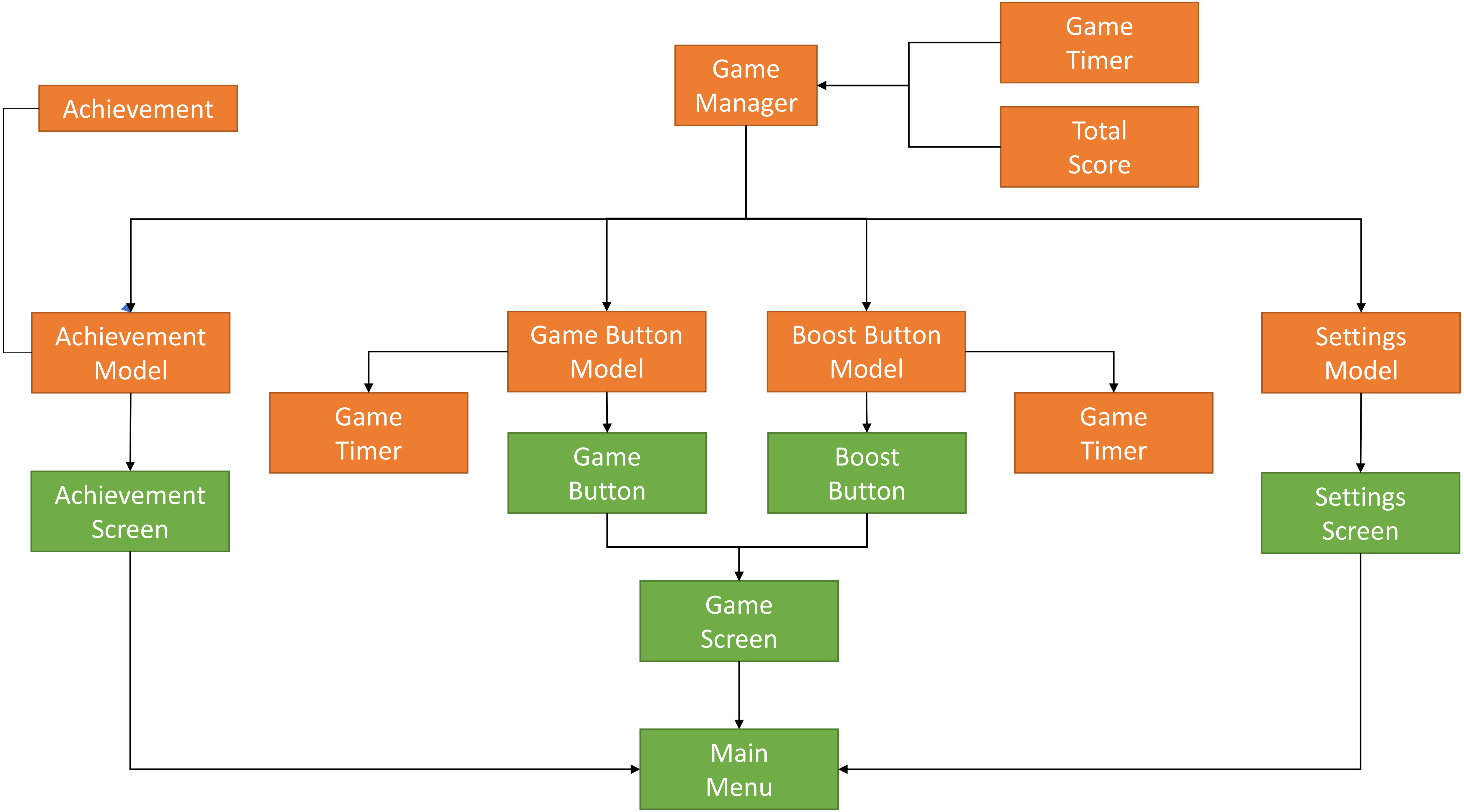
Game Timer
Game Manager
Achievement
Total Score
Achievement Model
Game Button Model
Boost Button Model
Settings Model
Game Timer
Game Button
Boost Button
Game Timer
Achievement Screen
Settings Screen
Game Screen
Main Menu

Game Manager

Purpose: Maintains state of the game, loads and saves files

Data Members: Game State (Enum)

Responsibility:

Load file

Save file

Save settings

New file

Game Manager - Parameters

Name	Type	Definition
GameState	Enum	Stores the current state of the game e.g. Menu, Initialising, Restart

Game Manager - Functions

Action	Result
Load File	• User clicks on “Start Game” • Locates the last save of the game • Provides each game object with their state
Save File	• User clicks on the back button while in the game or closes the application • Collects each game object’s state
Save Settings	• User changes settings in settings state • Saves setting state
New File	• User clicks restart game • Reinitialises game objects • Overwrites saved file

Game Timer

Purpose: Constructs a timer providing a countdown or a stopwatch like function

Data Members: Running (Boolean)

Responsibility:

Countdown

Elapsed Time

Start

Stop

Game Timer - Parameters

Name	Type	Definition
Running	Boolean	Shows whether the timer is running
Timer	Timer	The timer object

Game Timer - Functions

Action	Result
Countdown	• Timer starts • Timer will count down from the given time • Stops when the given time is reached
Elapsed Time	• Starts counting from current time • When stopped, gives the elapsed time
Start	• Constructs a timer and sets it to running
Stop	• Stops the running timer and emits a finished signal

Total Score

Purpose: Keeps track of the total score in the game

Parameters: Total Score (Integer)

Responsibility:

Provides the total score of the game

Deducts the score when game buttons are bought

Increases score when the game button is clicked

Deducts from the score when the game button is automated

Total Score- Parameters

Name	Type	Definition
Score	Integer	Provides the total score of the game

Total Score- Functions

Action	Result
Show Score	• Returns the score
Add to Score	• When a game button is clicked, the total score increases • When a game button is automated, the automated click will increase the total score
Deduct from Score	• When a game button is bought • When a game button is automated

Game Button Model

Purpose: Contains all data required for the game buttons and the functionality that they need

Data Members: Cost (Integer), Initial Score (Integer), Score (Integer), Timer (Game Timer), Amount (Integer), Cooldown (Integer), Automate Cost (Integer), Automated (Boolean), Boosted (Boolean)

Responsibility:

Button can be clicked

More buttons can be bought

Button purchases automation

Button automation routine

Button boosted

Game Button Model - Parameters

Name	Type	Definition
Score	Integer	Provides the number of points the button will add to the score
Timer	Game Timer	Used to track how long the button has left of its cooldown
Cooldown	Integer	Shows the amount of time the button has to wait before being clicked again
Cost	Integer	How much the button will deduct from the total game score when bought
Amount	Integer	The amount of that button
Automate Cost	Integer	Cost to automate the button
Automated	Boolean	Shows whether the button is automated or not
Boosted	Boolean	If this is true, the score of the button will be multiplied by 10

Game Button Model - Functions

Action	Result
Button is Clicked	• The button’s score is added to the total score • A timer starts that deactivates the mouse area so the user cannot click the button again during this time • A gradient fill will look at time elapsed and fill relative to this • A timer then ends and the mouse area is activated
Button is Bought	• Each button has a cost and the total game score will divide itself by this cost and the button will show the amount that can be bought of that button • When the amount is bought, the amount of the button increases by the number bought • When the amount is bought, the cost of the amount being bought is deducted from the total game score • When the amount is bought, the score of the button will increase relatively • When the amount is bought, the cost of the button will increase relatively

Game Button Model - Functions

Action	Result
Button Purchases Automation	• The button will show how much it costs to automate the button • When the buy automation is clicked, this cost is deducted from the total game score • The button will then show automated
Button Automation	• While automated, the button will automatically click itself whenever it is off cooldown
Button Boost	• When the boost button is clicked, the button’s score will be multiplied by 10

Boost Button Model

Purpose: Contains all data required for the boost button activity

Data Members: Active Boost Time (Integer), Game Timer (Game Timer), Cooldown Timer (Game Timer), Boosting (Boolean)

Responsibility:

Button is clicked

Boost Button- Parameters

Name	Type	Definition
Active Boost Time	Integer	The amount of time the boost runs for
Game Timer	Game Timer	Used to track how long the button has left on the active boost
Cooldown Timer	Game Timer	Used to track 4 hours of gameplay
Boosting	Boolean	Shows whether the boost button is currently active or not

Boost Button- Functions

Action	Result
Button is Clicked	• The other game button's score gets boosted • A timer begins that counts down for however long the boost lasts • While the boost is active the mouse area of the boost button is not enabled • After the boost is ended a timer is used to count 4 hours of gameplay • After 4 hours of gameplay, the boost button's mouse area becomes active

Achievement Model

Purpose: Provides a collection of achievements and their states

Data Members: Achievements (List<Achievements>)

Responsibility:

Return a list of all the achievements

Return whether an achievement is complete or not

Achievement Model - Parameters

Name	Type	Definition
Achievements	List<Achievement>	A list of all the achievements available in the game

Achievement Model - Functions

Action	Result
Return Achievements	• Return the list of achievements
Return Achievement	• Search the list of achievements • Return the achievement

Achievement

Purpose: Creates the different achievements within the game. Stores all data needed for this

Data Members: Name (String), Image (Image), Description (String), State (Enum)

Responsibility:

Should show the title of the achievement and the accompanying image.

A more detailed explanation is then needed which will be used when the achievement is clicked on

Stores the state of the achievement

Achievement - Parameters

Name	Type	Definition
Name	String	A unique name for the achievement
Image	Image	Identifiable image for the achievement
Description	String	Gives a description on how to complete the achievement
State	Enum	The state defines whether the achievement has been completed or not

Achievement - Functions

Action	Result
Return State	• Returns whether the achievement has been completed or not

Settings Model

Purpose: Contains all data for the settings – stores the properties

Data Members: Sound Volume (Integer), Resolution (Pair(Int, Int), FullScreenState (Boolean)

Responsibility:

Change the volume of the application

Change the resolution of the application

Change the full screen mode of the application

Settings Model - Parameters

Name	Type	Definition
Sound Volume	Integer	Store the sound volume of the application
Resolution	Pair<Int, Int>	Store the resolution of the application
Full Screen Mode	Boolean	Stores whether the application is in full screen or windowed mode

Settings Model - Functions

Action	Result
Change Volume	• User moves the slider up or down • Turns the volume up or down of the application • Change is saved
Change Resolution	• User selects a resolution from the drop down box • Application changes resolution • Change is saved
Change Full Screen Mode	• User unchecks or checks the box for full screen mode • Application changes window state • Change is saved

Game Button

Purpose: View for the game button model

Responsibility:

Must look at the Game Button Model and maintain the properties

Display: Show the UI needed for the game button

Buy Button Pressed: When the user clicks to buy the game button emit onBuyButtonPressed

Game Button Pressed: When the user clicks on the game button, emit the onGameButtonPressed

Automate Button Pressed: When the user clicks to automate the game button, emit the onAutomatePressed

Boost Button

Purpose: View for the boost button model

Responsibility:

Must look at the Boost Button Model and maintain the properties

Display: Show the UI needed for the boost button

Boost Clicked: When the user clicks on the boost, emit onBoostClicked

Game Screen

Purpose: A container for the various game items - contains all items needed to play the game

Responsibility:

Display: Show the six game buttons and the one boost button, the total score and a back button

Game Button Pressed: emit onGameButtonPressed

Boost Button Pressed : emit onBoostButtonPressed

Back Pressed: emit onBackPressed

Settings Screen

Purpose: View for the settings model – displays sliders/checkboxes etc. for changing various settings

Responsibility:

Display: Using properties provided, display the various UI needed

Sound Changed: When the user moves the progress bar for the sound, onSoundChanged is emitted

Resolution Changed: When the user selects a new resolution, onResolutionChanged is emitted

Full Screen Changed: When the user checks/unchecks full screen mode, the onFullScreenChanged is emitted

Back Pressed: When the user selects the back button, the onBackPressed is emitted

Achievement Screen

Purpose: View for the achievement model – displays the achievements in a grid like view

Responsibility:

Display: Display the Achievement model with various UI needed

Back Pressed: When the back button is pressed, the onBackPressed is emitted

Achievement Selected: Create an overlay displaying properties of the achievement selected

Main Menu

Purpose: Provides the user with the options to go between the various screens

Responsibility:

Maintain the state of the game – when changing states, the screen visible needs to be updated.

Display: When shown, display the various UI required

Start Clicked: Game screen needs to be shown

Restart Clicked: Game screen needs to be shown

Achievements Clicked: Achievement screen needs to be shown

Settings Clicked: Settings screen needs to be shown

Quit Clicked: The application will close