

3.1 User Interface Subsystem

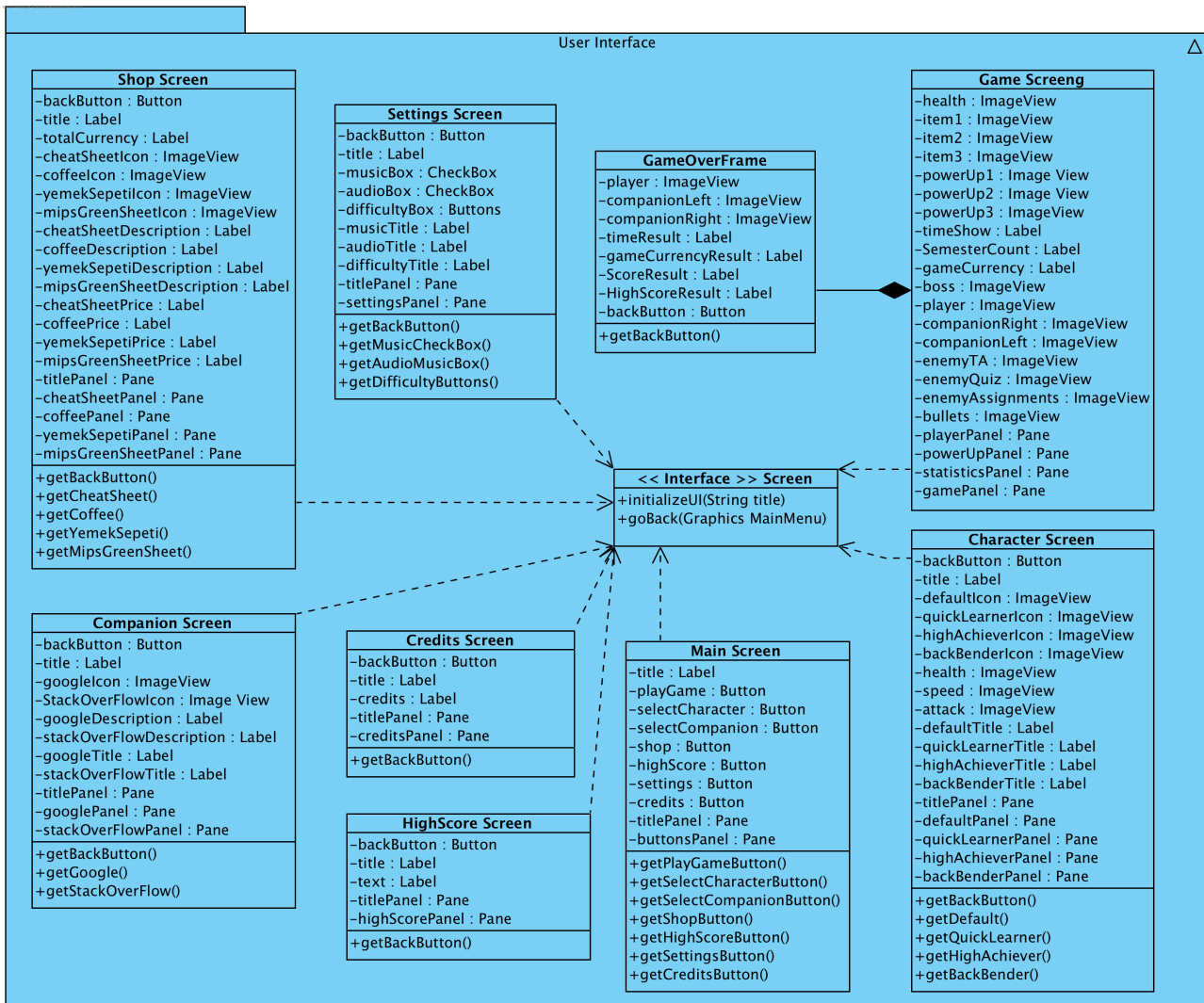


Figure X. Detailed User Interface Subsystem

User Interface Subsystem of «A day in Bilkent» is composed of 8 different classes, each for one screen and a Frame that is going to be accessed in one of those 8 classes. All the classes extend the Screen Interface which consists of two methods: initializing the GUI and going back to the Main Menu. Main Menu can be called the main screen for the game, because all the other screens are accessed through it. Each of the 8 screens have their own controller classes, which will implement all the needed functionality. However all GUI classes have get methods in order to give the controller classes the ability to access needed buttons. GameOverFrame is inside the Game Screen Class and will be called only when the player dies or completes the game, that is when the game ends. Overall, the diagram of the User Interface Subsystem above shows how the UI will be implemented and how it functions.

Key points:

1. All Screen Classes are the done using MVC logic and are strictly GUI.
2. All the Screen Classes are implementing the same Screen Interface.
3. The Main Screen is a menu screen which gives access to all the other screens through predefined buttons.
4. Almost all the Screen Classes, except Game Screen, have get methods as a link to their Controller Classes.
5. Game Screen has the GameOverFrame which will be used once the game ends.

3.1.1 <<Interface>> Screen

<< Interface >> Screen
+initializeUI(String title) +goBack(Graphics MainMenu)

Figure X. Screen Interface

- public void initializeUI(String title) - initializes the needed UI due to the specified parameter.
For example: title is Settings -> Settings UI is initialized.
- public void goBack(Graphics MainMenu) - When user returns to Main Menu, loads it.

3.1.2 Main Screen

Main Screen
-title : Label -playGame : Button -selectCharacter : Button -selectCompanion : Button -shop : Button -highScore : Button -settings : Button -credits : Button -titlePanel : Pane -buttonsPanel : Pane
+getPlayGameButton() +getSelectCharacterButton() +getSelectCompanionButton() +getShopButton() +getHighScoreButton() +getSettingsButton() +getCreditsButton()

Figure X. Main Screen

Attributes

- private Label title – the title of the Screen Class that is currently executing
- private Button playGame – accesses the Game Screen
- private Button selectCharacter – accesses the Select Character Screen
- private Button selectCompanion – accesses the Select Companion Screen
- private Button shop – accesses the Shop Screen
- private Button highScore – accesses the HighScore Screen
- private Button settings – accesses the Settings Screen
- private Button credits – accesses the Credits Screen
- private Panel titlePanel – panel for the title
- private Panel buttonsPanel – panel for 7 predefined buttons

Methods:

- public Button getPlayGameButton() – returns the Play Game button to the controller
- public Button getSelectCharacterButton() – returns the Select Character button to the controller
- public Button getSelectCompanionButton() – returns the Select Companion button to the controller
- public Button getShopButton() – returns the Shop button to the controller
- public Button getHighScoreButton() – returns the HighScore button to the controller
- public Button getSettingsButton() – returns the Settings button to the controller
- public Button getCreditsButton() – returns the Credits button to the controller

3.1.3 Game Screen

Game Screeng
-health : ImageView
-item1 : ImageView
-item2 : ImageView
-item3 : ImageView
-powerUp1 : Image View
-powerUp2 : Image View
-powerUp3 : ImageView
-timeShow : Label
-SemesterCount : Label
-gameCurrency : Label
-boss : ImageView
-player : ImageView
-companionRight : ImageView
-companionLeft : ImageView
-enemyTA : ImageView
-enemyQuiz : ImageView
-enemyAssignments : ImageView
-bullets : ImageView
-playerPanel : Pane
-powerUpPanel : Pane
-statisticsPanel : Pane
-gamePanel : Pane

Figure X. Game Screen

Attributes

- private ImageView health – the current health of the Player
- private ImageView item1 – if equipped, one of the items of the Player
- private ImageView item2 – if equipped, one of the items of the Player
- private ImageView item3 – if equipped, one of the items of the Player
- private ImageView powerUp1 – if equipped, one of the drop-powerUps of the Player
- private ImageView powerUp2 – if equipped, one of the drop-powerUps of the Player
- private ImageView powerUp3 – if equipped, one of the drop-powerUps of the Player
- private ImageView player – the Player model
- private ImageView companionRight – the right Companion model
- private ImageView companionLeft – the left Companion model
- private ImageView boss – the main boss model
- private ImageView enemyTA – model of one of the enemies
- private ImageView enemyQuiz – model of one of the enemies
- private ImageView enemyAssignments – model of one of the enemies
- private ImageView bullets – model of bullets that are shot at player/by player
- private Label timeShow – the time elapsed during the game
- private Label semesterCount – the waves of enemies cleared
- private Label gameCurrency – the game currency gained during the game
- private Pane playerPanel – panel for the player's health and items
- private Pane powerUpPanel – panel for the player's equipped power ups
- private Pane statisticsPanel – panel for the statistics of the game played
- private Pane gamePanel – the main game panel

3.1.4 GameOver Frame

GameOverFrame
-player : ImageView
-companionLeft : ImageView
-companionRight : ImageView
-timeResult : Label
-gameCurrencyResult : Label
-ScoreResult : Label
-HighScoreResult : Label
-backButton : Button
+getBackButton()

Figure X. GameOver Frame

Attributes

- private ImageView player – the Player model
- private ImageView companionRight – the right Companion model
- private ImageView companionLeft – the left Companion model
- private Label timeResult – the final time result of the game played
- private Label gameCurrencyResult – the collected currency
- private Label ScoreResult – the final score the player got
- private Label HighScoreResult – the high score the player has: beaten/unbeaten
- private Button backButton – back button, returns the player to the main menu

Methods:

- public Button getBackButton() – returns the Back button to the controller

3.1.5 Character Screen

Character Screen
<div><div>-backButton : Button</div><div>-title : Label</div><div>-defaultIcon : ImageView</div><div>-quickLearnerIcon : ImageView</div><div>-highAchieverIcon : ImageView</div><div>-backBenderIcon : ImageView</div><div>-health : ImageView</div><div>-speed : ImageView</div><div>-attack : ImageView</div><div>-defaultTitle : Label</div><div>-quickLearnerTitle : Label</div><div>-highAchieverTitle : Label</div><div>-backBenderTitle : Label</div><div>-titlePanel : Pane</div><div>-defaultPanel : Pane</div><div>-quickLearnerPanel : Pane</div><div>-highAchieverPanel : Pane</div><div>-backBenderPanel : Pane</div></div>
<div><div>+getBackButton()</div><div>+getDefault()</div><div>+getQuickLearner()</div><div>+getHighAchiever()</div><div>+getBackBender()</div></div>

Figure X. Character Screen

Attributes

- private Button backButton - back button, returns the player to the main menu
- private Label title - the title of the Screen Class that is currently executing
- private ImageView defaultIcon - the icon of the default character
- private ImageView quickLearnerIcon - the icon of the Quick Learner character
- private ImageView highAchiever - the icon of the highAchiever character
- private ImageView backBenderIcon - the icon of the Back Bender character
- private ImageView health - health of the characters
- private ImageView speed - speed of the characters
- private ImageView attack - attack of the characters
- private Label defaultTitle - the title that shows the name of the default character
- private Label quickLearnerTitle - the title that shows the name of the Quick Learner character
- private Label highAchieverTitle - the title that shows the name of the High Achiever character
- private Label backBenderTitle - the title that shows the name of the Back Bender character
- private Pane titlePanel - panel for the title
- private Pane defaultPanel - panel for the default character, his statistics and icon
- private Pane quickLearnerPanel - panel for the Quick Learner character, his statistics and icon
- private Pane highAchieverPanel - panel for the High Achiever character, his statistics and icon
- private Pane backBenderPanel - panel for the Back Bender character, his statistics and icon

Methods:

- public Button getBackButton() - returns the Back button to the controller
- public Button getDefault() - returns the Default character choice to the controller
- public Button getQuickLearner() - returns the Quick Learner character choice to the controller
- public Button getHighAchiever() - returns the High Achiever character choice to the controller
- public Button getBackBender() - returns the Back Bender character choice to the controller

3.1.6 Companion Screen

Companion Screen
-backButton : Button -title : Label -googleIcon : ImageView -StackOverFlowIcon : Image View -googleDescription : Label -stackOverFlowDescription : Label -googleTitle : Label -stackOverFlowTitle : Label -titlePanel : Pane -googlePanel : Pane -stackOverFlowPanel : Pane
+getBackButton() +getGoogle() +getStackOverFlow()

Figure X. Companion Screen

Attributes

- private Button backButton - back button, returns the player to the main menu
- private Label title - the title of the Screen Class that is currently executing
- private ImageView googleIcon - icon for the Google companion
- private ImageView stackOverFlowIcon - icon for the Stack OverFlow companion
- private Label googleDescription - description text of the Google companion
- private Label stackOverFlowDescription - description text of the Stack OverFlow companion
- private Label googleTitle - the title that shows the name of the Google companion
- private Label stackOverFlowTitle - the title that shows the name of the Stack OverFlow companion
- private Pane titlePanel - panel for the title
- private Pane googlePanel - panel for the Google Companion: icon, title and description
- private Pane stackOverFlowPanel - panel for the Stack OverFlow Companion: icon, title and description

Methods:

- public Button getBackButton() - returns the Back button to the controller
- public Button getGoogle() - returns the Google companion choice to the controller
- public Button getStackOverFlow() - returns the Stack OverFlow companion choice to the controller

3.1.7 Shop Screen

Shop Screen
<div><div>-backButton : Button</div><div>-title : Label</div><div>-totalCurrency : Label</div><div>-cheatSheetIcon : ImageView</div><div>-coffeelcon : ImageView</div><div>-yemekSepetilcon : ImageView</div><div>-mipsGreenSheetIcon : ImageView</div><div>-cheatSheetDescription : Label</div><div>-coffeeDescription : Label</div><div>-yemekSepetiDescription : Label</div><div>-mipsGreenSheetDescription : Label</div><div>-cheatSheetPrice : Label</div><div>-coffeePrice : Label</div><div>-yemekSepetiPrice : Label</div><div>-mipsGreenSheetPrice : Label</div><div>-titlePanel : Pane</div><div>-cheatSheetPanel : Pane</div><div>-coffeePanel : Pane</div><div>-yemekSepetiPanel : Pane</div><div>-mipsGreenSheetPanel : Pane</div></div>
<div><div>+getBackButton()</div><div>+getCheatSheet()</div><div>+getCoffee()</div><div>+getYemekSepeti()</div><div>+getMipsGreenSheet()</div></div>

Figure X. Shop Screen

Attributes

- private Button backButton – back button, returns the player to the main menu
- private Label title – the title of the Screen Class that is currently executing
- private Label totalCurrency – represents total currency of the user currently
- private ImageView cheatSheetIcon – icon of the Cheat Sheet item
- private ImageView coffeeIcon – icon of the Coffee item
- private ImageView yemekSepetiIcon – icon of the YemekSepeti item
- private ImageView mipsGreenSheetIcon – icon of the MIPS Green Sheet item
- private Label cheatSheetDescription – represents description for the Cheat Sheet item
- private Label coffeeDescription – represents description for the Coffee item
- private Label yemekSepetiDescription – represents description for the YemekSepeti item
- private Label mipsGreenSheetDescription – represents description for the MIPS Green Sheet item

- private Label cheatSheetPrice - represents price for the Cheat Sheet item
- private Label coffeePrice- represents price for the Coffee item
- private Label yemeksepetiPrice - represents price for the YemekSepeti item
- private Label mipsGreenSheetPrice - represents price for the MIPS Green Sheet item
- private Pane titlePanel - panel for the title
- private Pane cheatSheetPanel - panel for the Cheat Sheet item: icon, title, description, price
- private Pane coffeePanel - panel for the Coffee item: icon, title, description, price
- private Pane yemekSepetiPanel - panel for the YemekSepeti item: icon, title, description, price
- private Pane mipsGreenSheetPanel - panel for the MIPS Green Sheet item: icon, title, description, price

Methods:

- public Button getBackButton() - returns the Back button to the controller
- public Button getCheatSheet() - returns the Cheat Sheet button to the controller
- public Button getCoffee() - returns the Coffee button to the controller
- public Button getYemekSepeti() - returns the YemekSepeti button to the controller
- public Button getMIPSGreenSheet() - returns the MIPS Green Sheet button to the controller

3.1.8 Settings Screen

Settings Screen
-backButton : Button -title : Label -musicBox : CheckBox -audioBox : CheckBox -difficultyBox : Buttons -musicTitle : Label -audioTitle : Label -difficultyTitle : Label -titlePanel : Pane -settingsPanel : Pane
+getBackButton() +getMusicCheckBox() +getAudioMusicBox() +getDifficultyButtons()

Figure X. Settings Screen

Attributes

- private Button backButton - back button, returns the player to the main menu
- private Label title - the title of the Screen Class that is currently executing
- private CheckBox musicBox - checked music is on, unchecked music is muted
- private CheckBox audioBox - checked audio is on, unchecked audio is muted
- private Button difficultyBox - 3 buttons together as a group: easy normal and hard difficulty selector
- private Label musicTitle - title for the music check box
- private Label audioTitle - title for the audio check box
- private Label difficultyTitle - title for the difficulty selector button
- private Pane titlePanel - panel for the title
- private Pane settingsPanel - panel for the settings options, checkboxes and buttons

Methods:

- public Button getBackButton() - returns the Back button to the controller
- public Button getMusicCheckBox() - returns the music box checked/unchecked to the controller
- public Button getAudioCheckBox() - returns the audio box checked/unchecked to the controller
- public Button getDifficultyButtons() - returns the selected difficulty button to the controller

3.1.9 HighScore Screen

HighScore Screen
-backButton : Button -title : Label -text : Label -titlePanel : Pane -highScorePanel : Pane
+getBackButton()

Figure X. HighScore Screen

Attributes

- private Button backButton - back button, returns the player to the main menu
- private Label title - the title of the Screen Class that is currently executing

- private Label text - represents the high scores
- private Pane titlePanel - panel for the title
- private Pane highScorePanel - panel for the high scores represented as texts

Methods:

- public Button getBackButton() - returns the Back button to the controller

3.1.10 Credits Screen

Credits Screen
-backButton : Button -title : Label -credits : Label -titlePanel : Pane -creditsPanel : Pane
+getBackButton()

Figure X. HighScore Screen

Attributes

- private Button backButton - back button, returns the player to the main menu
- private Label title - the title of the Screen Class that is currently executing
- private Label credits - represents the credits rolling from bottom to top
- private Pane titlePanel - panel for the title
- private Pane creditsPanel - panel for the credits represented as a text

Methods:

- public Button getBackButton() - returns the Back button to the controller