**flappybruin.py** defines the overall interface of the game and is responsible for a variety of miscellaneous game-related tasks.

We used **QtDesigner** to create the barebones layout of the game.



First, we create a **background widget** with a **vertical layout**. We have three **frames**, the **top frame** (scores), the **middle frame** (game) and **bottom frame** (options). The first frame has a **horizontal layout**, from left to right, two **labels**, a **horizontal spacer** and another two **labels**. The center frame has a **vertical layout**, from top to bottom, a **vertical spacer**, a **label**, a **tool button** and a **vertical spacer**. The bottom frame has a **horizontal layout**, from left to right, a **label**, a **horizontal spacer**, a **tool button**, a **horizontal spacer**, a **label** and a **combo box**.

**Imports**

**sys**: For starting and closing the game safely

**PyQt**: **QtCore**, **QtGui**, **QtWidgets** for the game elements

**bruin**: **Bruin** for the implementation of the bruin player

**building**: **Building** for the implementation of the buildings

**pygame**: Only for sound (because PyQt’s QtMultimedia was glitchy on one of our systems)

**FlappyBruinGame Members**

\_\_**init**\_\_: Initializes the class and all of its member variables, also calls setupUi

- Creates QTimer object used to repeatedly call **animate**

- Creates variables names **player** and **buildings** but sets them to None first

**setupAudio**: Sets up the audio input for the game

- Uses **pygame.mixer** functions to load music and play in a loop

**setupUi**: Sets up all the GUI elements of the game, is called by \_\_init\_\_

- Calls **setFixedSize** to fix the window size and prevent resizing

- Links the Start Game and Quit Game tool buttons to **gameStart** and **quitgame** respectively

- Links the combo box to **setMode**

**setMode**: Updates the difficulty mode when user interacts with the combo box

- Calls the player and buildlings’ **setDiff**

**gameStart**: Defines actions to start the game

- Stops the menu music, and starts playing the active game music

- Hides the center label and start game button

- Disables the difficulty combo box

- Calls the player and buildings’ **reset**

- Resets current score to 0

**gameEnd**: Defines actions to end the game

- Stops the active game music, and starts playing menu music

- Displays game over with the center label and restart on the start game button

- Reenables the difficulty combo box

**framePress**: Debugging tool used to output to the console the mouse position when clicking on the GUI

**keyPressEvent**: Defines what happens when user presses the ‘W’ key

- Calls player.**jump** when W is pressed

**framePaint**: Constructs a QPainter for the central frame and calls the player and building’s paint functions.

- Instantiates player and buildings when called the first time

- Sets focus on itself when mouse is not hovering over combo box

**eventFilter**: Allows game to switch event focus to the combo box

- Sets focus to the combo box when mouse is hovering over it

**updateScore**: Updates the score and high scores as the game progresses

- If score exceeds high score, update high score as well in real time

**quitgame**: Quits the game safely using **close**

**animate**: Calls update functions for Bruin, Building and the center frame, also calls updateScore

- If player is dead, calls **gameEnd**

- Calls **updateScore** while player is not dead