Техническая документация проекта “Своя Игра”

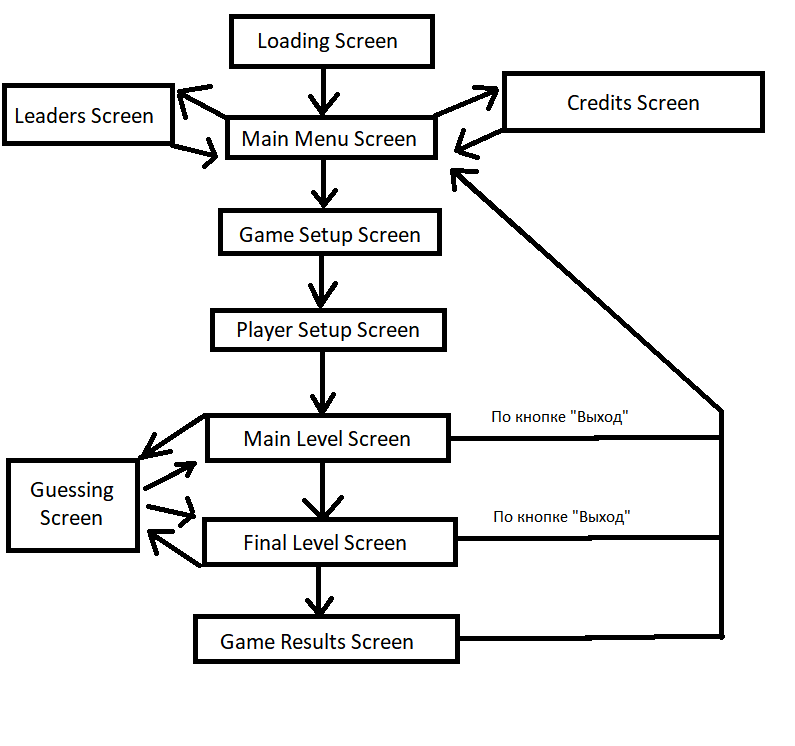
Общие детали реализации проекта:

Общая структура каждого окна:

Реализация с помощью QMainWindow, где всё содержимое хранится на QFrame, для которого при создании окон применяются следующие функции: изменение WindowTitle на “Своя Игра”, добавление к окну эффекта тени (QGraphicsDropShadowEffect), установка атрибутов, убирающих фон и рамку окна. Каждое из окон хранит одну инстанцию AudioPlayer – усовершенствованную версию QtMultimedia.QAudioPlayer со схожим интерфейсом.

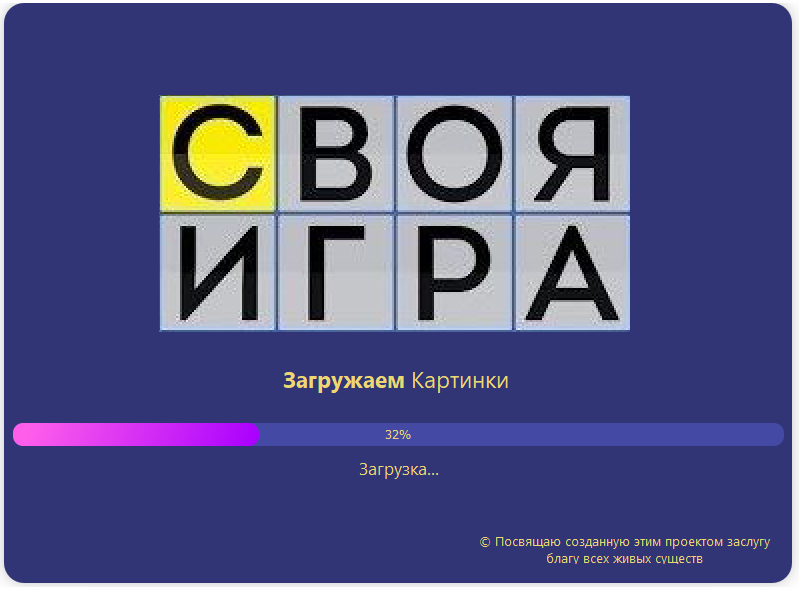
Каждое из окон, кроме главного меню и экрана загрузки, хранит ссылку на главное меню для возможности возвращения в него после закрытия этих окон (окно угадывания слова (GuessingScreen) хранит ссылку на игровой уровень, из которого оно было вызвано).

Отношения между окнами:



Структура основных классов проекта, их функционал и особенности реализации:

Экран загрузки игры (Loading Screen)



Основная функция – интерфейс для пользователя при запуске приложения. При создании окна происходит запуск вступительной композиции в AudioPlayer, создание таймера для отсчёта прогресса, транслируемого в виджет QProgressBar, таймера для отображения состояния загрузки. Информация о состоянии загрузки хранится в QLabel. Состояния загрузки читаются из файла. По завершению первого таймера происходит открытие окна-меню (MainMenuScreen) и закрытие окна загрузки.

Игровое меню (MainMenuScreen)



Функция окна – базовый пункт для перехода в другие окна: окно списка лидеров (LeadersScreen), окно благодарностей (CreditsScreen) и окно настройки параметров игры (GameSetupScreen) – при открытии которых происходит сокрытие (widget.hide()) окна-меню. Весь выше изложенный функционал (а также – функция закрытия приложения) закреплён через функции, вызываемые при нажатии на виджет QPushButton. При создании окна настройки параметра игры (GameSetupScreen) создаётся экземпляр класса GameManager, хранящего информацию о игре – сложности, статистике игрока и его противников, множестве использованных в игре тем.

Окно списка лидеров (LeadersScreen)



Функция окна – загрузка данных об игровых рекордах из базы данных, их отображение в виджете QTableWidget. Также наличествует виджет QPushButton – при нажатии на который происходит закрытие окна лидеров и возвращение (widget.show()) в главное меню (MainMenuScreen).

Окно благодарностей (CreditsScreen)



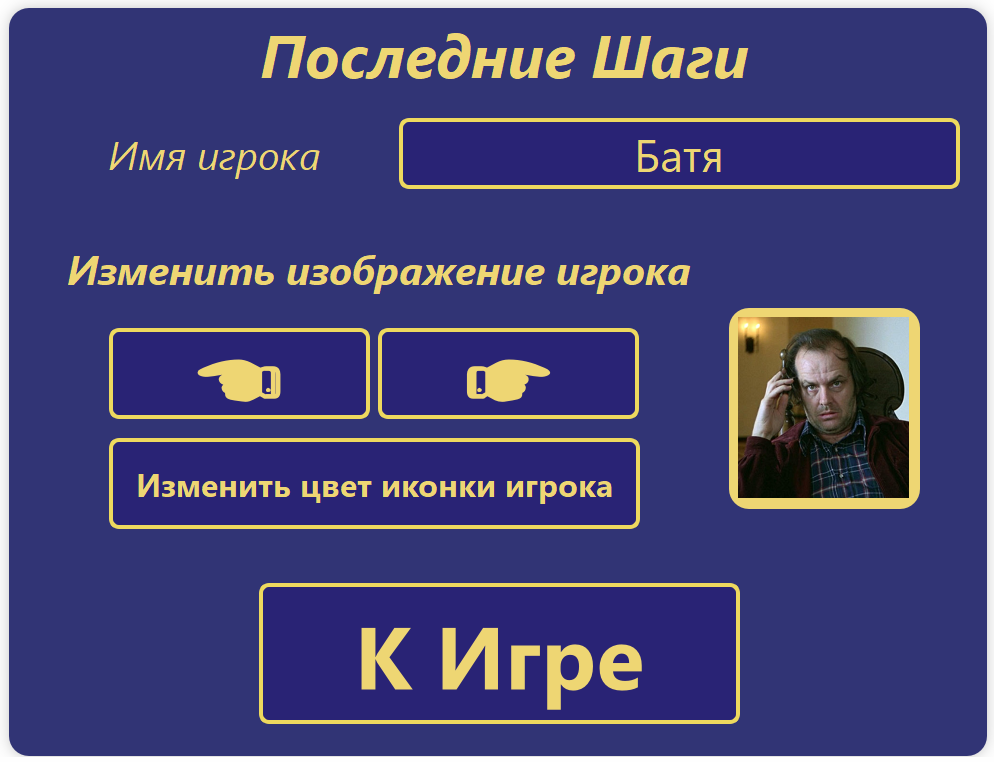
Функция окна – вывод в виджет QTextEdit записанных в файле благодарностей разработчикам, тестировщикам и людям, внесшим вклад в создание приложения. Также за виджетом QPushButton закреплён функционал закрытия окна и возвращения (widget.show()) к главному меню (MainMenuScreen).

Окно настройки игры (GameSetupScreen)



Функция окна – выбор сложности игры. При нажатии на один из виджетов QRadioButton, объединённых в QButtonGroup, из текстового файла получается информация о сложности – и отображается в соответствующий виджет QLabel, также информация о изменении сложности передаётся в GameManager, откуда также получаются изображения противников, которые отображаются в виджете QLabel. Виджет QPushButton отвечает за переход к следующему этапу подготовки игры – к открытию меню настройки игрока (PlayerSetupScreen) и закрытию текущего окна.

Окно настройки игрока (PlayerSetupScreen)



Функция окна – выбор параметров игрока – имени, изображения, цвета иконки. Изменение имени игрока происходит через пользовательский ввод в виджете QLineEdit (при пустом вводе - именем игрока автоматически становится “Игрок”). Виджеты QPushButton со стрелочками отвечают за переключение пользовательского изображения, список которых хранится в папке с проектом. Изменение цвета иконки происходит с помощью QColorDialog, изменения отображаются на QFrame. Виджет QPushButton с текстом “К Игре” отвечает за запуск основного уровня игры и закрытие текущего этапа – также в этот момент в GameManager передаётся информация о параметрах игрока, и в GameManager создаются экземпляры классов Player и Enemy для хранения данных об участниках игры.

Окно главного уровня (MainLevelScreen)



Окно отвечает за проведение основного этапа игры. При создании окна информация об игроках загружается в соответствующие виджеты (Имя – в QLabel, количество очков – в QLCDNumber, изображение – в QLabel, цвет иконки – в QFrame). Также при создании окна происходит загрузка вопросов из базы данных, названия тем отображаются в виджет QLabel, виджеты QPushButton связываются с функцией выбора вопроса, текст на них устанавливается по формуле номер\_раунда \* 100 \* номер\_кнопки\_в\_ряду.

Также в окне расположены 2 виджета QFrame – первый относится к статусу вопроса (хранит QLabel – пояснения вопроса, QLabel – содержания вопроса, QLabel – таймера-отсчёта времени на ответ, QPushButton – для перевода пользователя в меню угадывания (GuessingScreen)); второй – к статусу ответа (хранит QLabel – текущий игровой статус (ожидание отвечающих, прослушивание аудио-композиции и т.п), QLabel – статус ответа – его правильность и автор)

Основная логика уровня основана на системе состояний (записанных в py-файле game\_states) – основные игровые состояния: ожидание пользовательского выбора вопроса, вывод вопроса на экран (для аудио-вопроса – прослушивание композиции), ожидание ответа одного из игроков, пауза для вывода ответа одного из участников на экран, выбор вопроса ИИ. Состояния влияют на поведение функций выбора вопроса, функции ответа игрока - при несовпадении текущего игрового состояния с необходимым для функции – она досрочно завершается.

Выбор вопроса происходит в три этапа.

Первый – выбор вида вопроса. При создании окна случайным образом выбираются вопросы с нестандартным видом (кот в мешке – вопрос переходит к случайному игроку, вопрос без риска – при неправильном ответе очки не снимаются, вопрос с подвохом – цена вопроса увеличивается на случайное количество очков, максимум – на 2 цены вопроса) – для этих вопросов на виджете QFrame – для вопросов выводятся специальные пояснения. В зависимости от вида вопроса формируется список отвечающих на текущем ходу. Для обычного вопроса в этот список попадает каждый участник игры. Для специального – тот, кто выбирал вопрос, или тот, кому достался кот в мешке.

Второй – отображение вопроса на экран. Для аудио-вопросов в игре создаётся специальный таймер-задержка для прослушивания композиции, после завершения которого происходит третья стадия выбора вопроса, для всех остальных видов вопросов она начинается сразу.

Третий – настройка ИИ, смена игрового состояния на “ожидание ответа одного из игроков”. Настройка ИИ включает в себя создание таймеров, привязанных к функция ответа ИИ на вопрос (То, отвечает ли ИИ на текущий вопрос, время на размышления, правильность ответа – заложены во внутренней логике класса Enemy, изложенной ниже).

Функция проверки ответа универсальна для всех участников игры. При ответе пользователя её запуск происходит после открытия окна угадывания (GuessingScreen) – где и определяется правильность ответа. При ответе ИИ правильность ответа вычисляется с помощью внутренней логики класса Enemy, после чего вызывается функция проверки ответа.

При правильности ответа, данного участником игры, – в QFrame-статусе ответа выводится об этом уведомление (имя отвечающего, стоимость вопроса, QLabel-игровой статус уведомляет об ожидании выбора вопроса), участнику игры начисляются очки, к нему переходит очередь выбора вопроса, вызывается функция смены вопроса.

В случае – если дан неверный ответ – появляется соответствующее уведомление, однако очки снимаются в случае если вид вопроса не ‘Вопрос без риска’. Также в этот момент участник удаляется из списка отвечающих на текущем ходу. Если этот список оказывается пуст – игра переходит к следующему вопросу, уведомляя о том, что никто не ответил на вопрос в QFrame-статусе ответа, также эта функция вызывается при завершении таймера-отсчёта

При смене вопроса очищается QFrame-статус вопроса, отключается (widget.hide()) виджет QLabel с таймером времени ответа. В зависимости от выбирающего новый вопрос меняется игровое состояние (выбирает либо игрок, либо ИИ). Если вопросов больше не осталось – происходит загрузка финального раунда (FinalLevelScreen).

Окно финального уровня (FinalLevelScreen)



При старте окна случайным из базы данных загружается случайная (ещё не разыгранная в текущей игре) тема, один случайный вопрос из неё становится вопросом финального раунда.

Основная структура и инициализация UI элементов этого окна идентична окну главного уровня (MainLevelScreen), также окно включает в себя QFrame для получения ставки игрока (активируется в случае – если у игрока неотрицательное количество очков), на котором расположены QLabel - пояснение, QLabel – размер ставки, QSlider – для изменения размера ставки (шаг для QSlider – 10 очков), QPushButton – для запуска функции вывода вопроса на экран (в случае, если у игрока недостаточно баллов, соответствующее уведомление выводится в QFrame – статус ответа, функция вывода вопроса вызывается с задержкой).

Структура функции вывода вопроса идентична функции из окна главного уровня (MainLevelScreen) – за исключением того, что в финальном раунде вопрос всегда относится к стандартному виду.

Вопрос задаётся в 3 фазы:

1. Время на рассуждение (в случае, если игрок делал ставку – 15 секунд, в обратном – 5 секунд). Во время этой фазы происходит отсчёт оставшегося времени через QTimer, время отображается на QLabel, содержащийся на QFrame – статусе ответа.
2. Переход к окну угадывания слова (GuessingScreen) – только в случае, если игрок делал ставку.
3. Поочередный вывод состояния ответа каждого из участников игры (имя отвечающего, правильность ответа, ответ (в случае ошибки), размер ставки.

После этого происходит закрытие окна и открытие окна оглашения результатов (GameResultsScreen).

Окно оглашения результатов (GameResultsScreen)

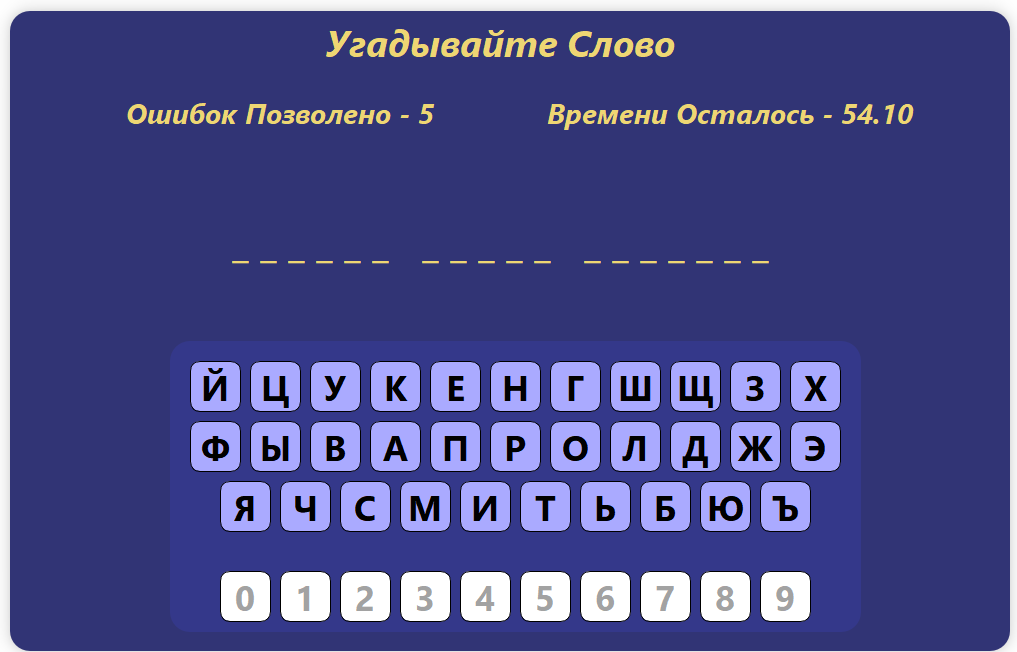


При открытии окна участники игры сортируются и выводятся в соответствии с количеством набранных баллов (при совпадении баллов приоритет отдаётся игроку). В зависимости от положения игрока (победа или поражение (2, 3 места)) – проигрывается определенная композиция, соответствующее уведомление выводится на QLabel. Также в зависимости от места игроку начисляется различное количество опыта, умноженного на (4 – номер\_места) – 25 за 3-е место, 50 за 2-е, 100 за 1-е.

В момент проигрывания композии в базе данных со списком лидеров происходит обновление информации по имени игрока. По необходимости – он добавляется в базу данных. Ему начисляется опыт за игру. Обновляется его рекорд, количество ошибок и правильных ответов.

Хранение информации об игроках аналогично игровым окнам (MainLevelScreen, FinalLevelScreen).

Окно угадывания слова (GuessingScreen)



Классу-окну в качестве параметра передаётся угадываемое слово, которые выводится на экран по следующему принципу: после каждого символа добавляется пробел, создаётся копия строки, все непробельные (а также не символы табуляции и перехода на новую строку) символы заменяются на нижние подчёркивания, вопрос загружается в виджет QLabel.

В зависимости от сложности игры и id вопроса (ответ – число или слово) GameManager возвращает разное количество допустимых ошибок игрока. Время на ответ – 60 секунд, во время которых в AudioPlayer проигрывается композиция, по истечению таймера-отсчёта времени фаза угадывания прекращается – слово считается неугаданным.

Угадывание происходит через вздаимодействие с группой QPushButton. При нажатии на кнопку происходит вызов функции проверки наличия символа в строке, при отсутствии совпадений – с игрока снимается 1 право на ошибку, при уходе в минус – угадывание прекращается, слово считается неугданным. После этого кнопка возможность взаимодействия с кнопкой отключается (widget.setEnabled(False)).

Также угадывание может быть осуществлено с помощью нажатия соответствующих кнопок на клавиатуре. В случае использование английской раскладки – происходит перевод на соответствующий русский символ. Перед запуском функции проверки наличия символа в строке происходит проверка на то, есть ли кнопка с соответствующим клавиатурному вводу символу и была ли эта кнопка уже нажата.

В конце своей работы происходит вызов функции проверки ответа игрока в окне, из которого было вызвано окно угадывания слова.

Ответственность класса GameManager

- Хранение и создания экземпляра класса Player, и двух экземпляров класса Enemy

- Хранение списка задействованных в текущей игре тем вопросов

- Хранение сложности игры

- Предоставления всем остальным классом интерфейса для взаимодействия с классами игрока и противников

Форма хранения и представления вопросов

Каждый вопрос имеет следующую структуру:

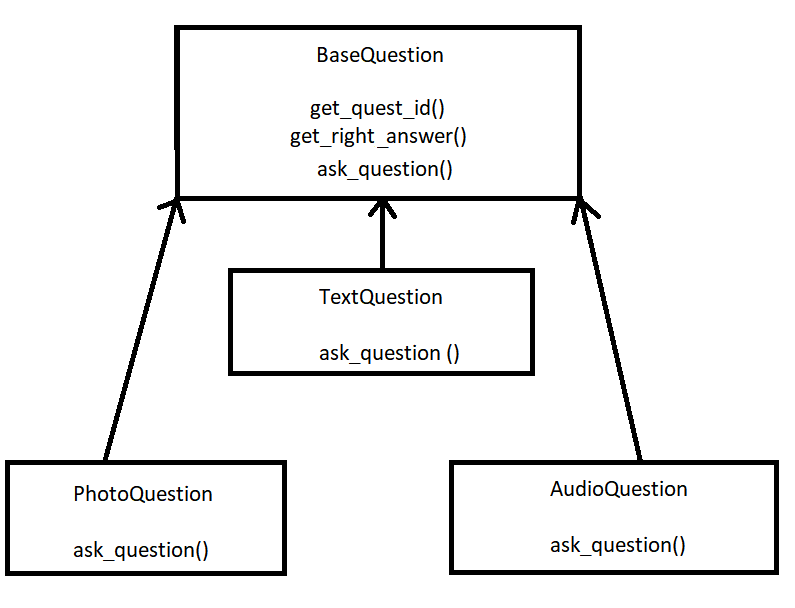
- Содержание вопроса (для аудио- и фото- вопросов – название файла с содержанием, для текстовых – сама формулировка вопроса)

- Пояснение к вопросу

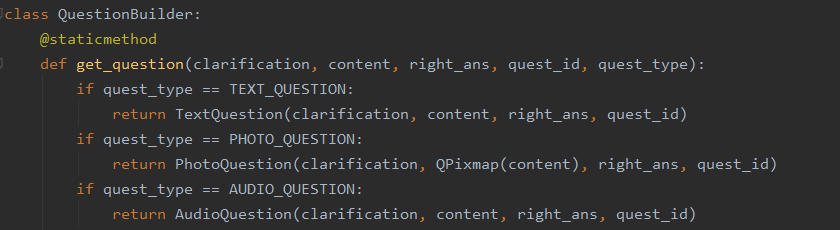
- Правильный ответ

- ID вопроса (числовая переменная: 0 – вопрос, в котором ответ – слово; 1 – вопрос, в котором ответ – число)

Общая иерархия классов вопросов:



За создание классов-вопросов отвечает класс-строитель, что позволяет абстрагировать всю работу по их инстанцированию и минимизировать работу над кодом при изменении или добавлении функционала вопросов или формы их представления, а также при добавлении новых типов вопросов.



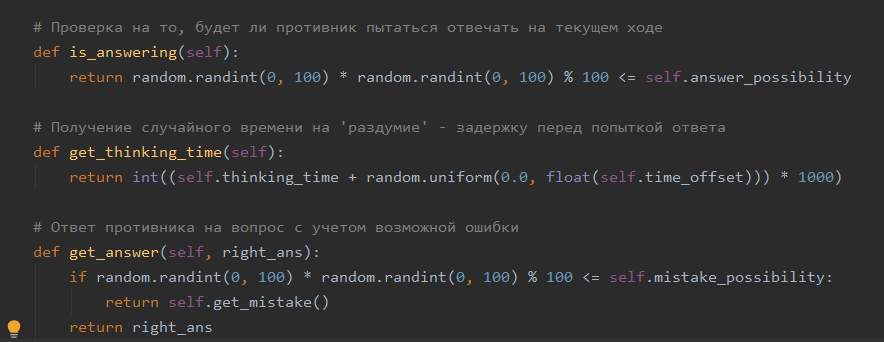
Хранение и представление участников игры

Существуют 2 класса-участника – Игрок (Player) и Оппонент (Enemy) – оба унаследованы от базового класса GameMember, ответственного за хранение и изменение очков, количества ошибок и правильных ответов участника игры, хранение имени и списка возможных ошибок (при ответе) участника игры.

Для класса игрока (Player) также хранится изображение и цвет иконки игрока.

Для класса оппонента (Enemy) хранятся возможность ошибки при ответе на вопрос; возможность того, что на текущем ходу оппонент будет отвечать; время-задержка перед ответом над вопросом; максимальное смещение времени (к времени-задержке добавляется случайное число от 0 до максимального смещения для того, чтобы оппоненты каждый раз отвечали, затрачивая разное количество времени на ‘размышления’).

Основное поведение Enemy закреплено в 3-х функциях:



Из-за присутствия в игре 3-х уровней сложности, создание классов Enemy абстрагируется и передаётся в руки классу-строителю:

