پروژه نهایی برنامه سازی پیشرفته

فاز چهارم

استاد : نرگس السادات بطحائیان

محمدرضا حیدرنیا – ۹۹۱۲۳۵۸۰۱۸

زمستان ۱۴۰۰

شرح پروژه

در فاز چهارم باید برنامه ایی را که در سه فاز قبلی نوشته بودیم؛ به صورت گرافیگی طراحی میکردیم. که من در این فاز از کیوت و زبان qml استفاده کردم.

یادگیری qml و همچنین ارتباط کد cpp با کیوت از اصلی ترین چالش های این فاز بود.

ساختار برنامه

در ارتباط cpp و qml در بیشتر کلاس ها؛ توابع get به عنوان اسلات در نظر گرفته شده است.

کلاس doctor :

اسلات ها :

```
public slots:
    int get_coin();                         //return coin
    int get_xp() const;                     //return experience value
    int get_energy() const;                 //return enegy value
    void boost_energy();                     //boost energy
    int get_credit() const;                 //return credit value
};
```

سیگنال ها:

```
signals:
    void noticecoin(QVariant data);//a signal for return coin value
    void noticexp(QVariant data) const;//a signal for return XP value
    void noticeenergy(QVariant data) const;//a signal for return energy value
    void noticecredit(QVariant data) const;//a signal for return credit value
    void noticemessagedoctor(QVariant note) const;//a signal for message
    void gameover(QVariant message)const;//a signal for show game over
```

سیگنال ها مقدار سکه ، xp , انرژی ، اعتبار را emit میکنند. البته یک سیگنال هم برای باختن(gameover) وجود دارد که اگر مقدار energy به صفر برسد emit میشود.

کلاس patient:

اسلات ها:

```
public slots:
    int get_health_count();//return health count
    int get_sick_count();//return sick count
    int treatment(int g);//treatment method
```

سیگنال ها :

```
signals:
    void noticehealth(QVariant data);//a signal for return count of healthy person
    void noticesick(QVariant data);//a signal for return count og sick person
    void noticetreatmessage(QVariant note);//a signal for show messages
```

سیگنال ها تعداد افراد بهبود یافته، افراد مریض ، و خطا ها را emit میکند.

همه این سیگنال ها در اسلات های مربوطه فراخوانی میشوند.

کلاس medicaldevice:

اسلات ها:

```
public slots:
    virtual void buy() = 0; // buy device(abstract class)
    virtual bool use() = 0;      // how to use(abstract class)
    virtual int get_count() = 0;         //return count of pill or ampoule
```

سیگنال:

```
signals:
    void noticegetpillcount(QVariant data);
```

کلاس pill:

اسلات:

```
public slots:
    virtual int get_count() override;//return value
    virtual void buy( ) override;//buy method
    virtual bool use() override;//use method
```

سیگنال:

```
signals:
    void noticegetpillcount(QVariant data);//a signal for show count of pill
    void noticemessage(QVariant note);//a signal for show error message
```

سیگنال ها تعداد دارو ها و خطا را نشان می دهند.

کلاس ampoule:

اسلات ها :

```
public slots:
    virtual void buy() override;//buy method
    virtual bool use() override;//use method
    virtual int get_count() override;//return value
```

سیگنال:

```
signals:
    void noticgetampoulecount(QVariant data);//a signal for update ampoule count
    void noticeampoulemessage(QVariant note);//a signal for show error mrssage
```

تعداد آمپول و خطا ها را به qml می فرستد.

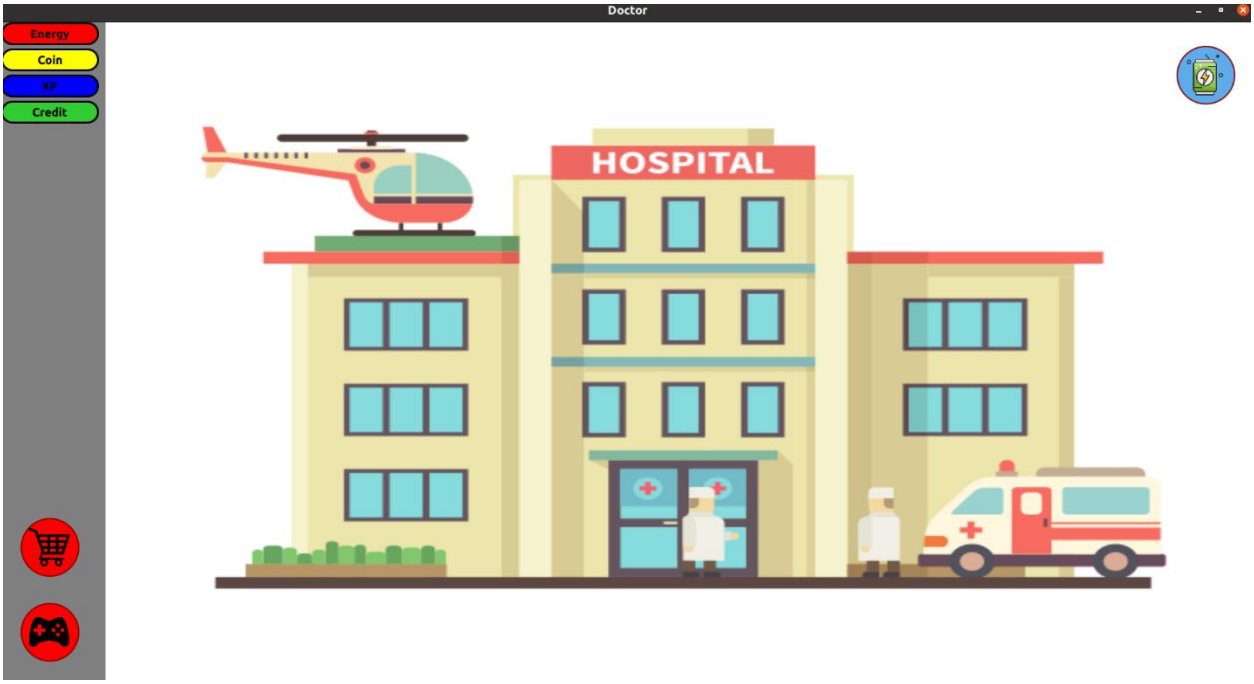
:main.cpp

```cpp
int main(int argc, char *argv[])
{
    QCoreApplication::setAttribute(Qt::AA_EnableHighDpiScaling);

    QGuiApplication app(argc, argv);

    QQmlApplicationEngine engine;
    doctor d("doc");
    engine.rootContext()->setContextProperty("doctor", &d);
    pill P{"pill"};
    engine.rootContext()->setContextProperty("pilldevice", &P);
    ampoule a{"ampoule"};
    engine.rootContext()->setContextProperty("ampouledevice", &a);
    patient patient;
    engine.rootContext()->setContextProperty("person", &patient);
    const QUrl url(QStringLiteral("qrc:/main.qml"));
    QObject::connect(&engine, &QQmlApplicationEngine::objectCreated,
                     &app, [url](QObject *obj, const QUrl &objUrl) {
        if (!obj && url == objUrl)
            QCoreApplication::exit(-1);
    }, Qt::QueuedConnection);
    engine.load(url);

    return app.exec();
}
```

ساختار گرافیکی

برنامه به شکل stackveiw طراحی شده بنابراین در main.qml ؛ stackveiw.qml لود شده و در آنجا دکمه های مربوط به صفحه نخست طراحی شده. البته یک label هم برای نمایش خطا در صفحه نخست موجود است.

صفحه نخست:



در گوشه سمت چپ بالا یکسری مستطیل (که در Myrectangel.qml طراحی شده) وجود دارد برای نمایش xp، سکه ، انرژی ، و اعتبار.

در ابتدا انرژی برابر با ۱۰۰و تعداد سکه ها برابر با ۳۰است.

در گوشه سمت چپ پایین دکمه :

برای شروع بازی، و دکمه:

برای خرید وسایل برنامه ریزی شده است.

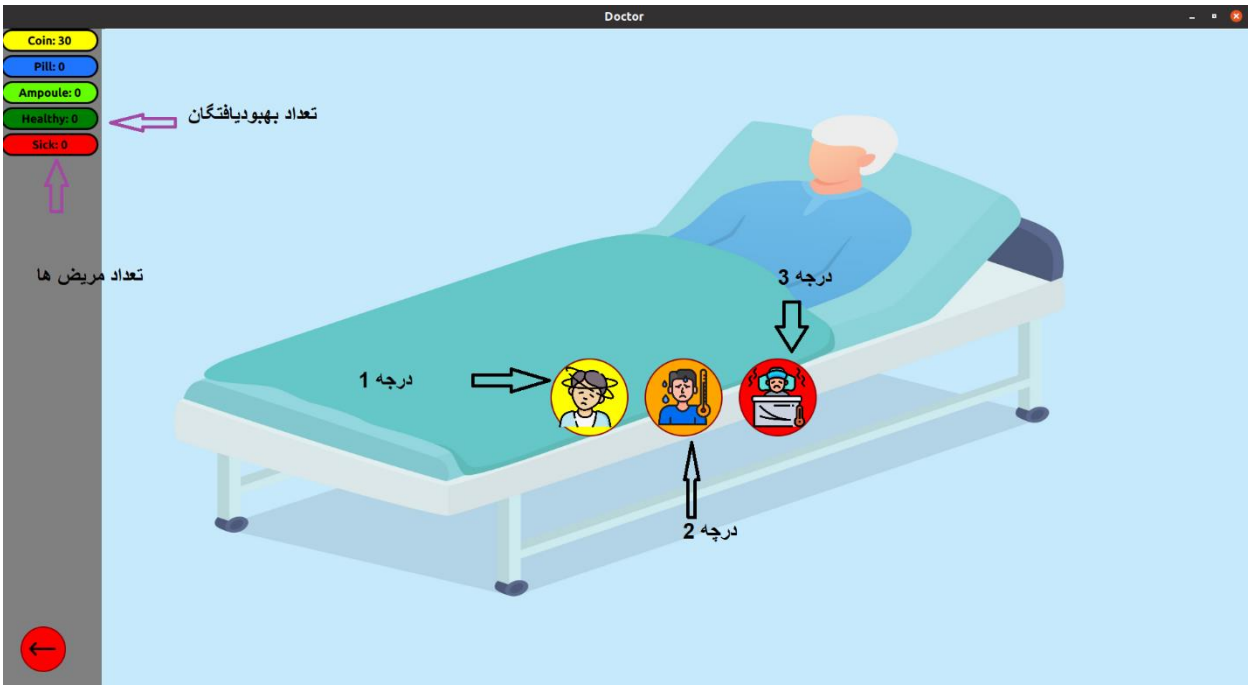
البته دکمه:

برای افزایش انرژی موجود است.(xp کم میشود و به انرژی اضافه میشود.)

با کلیک بر روی دکمه خرید وارد صفحه (Buyzone.qml) میشویم:

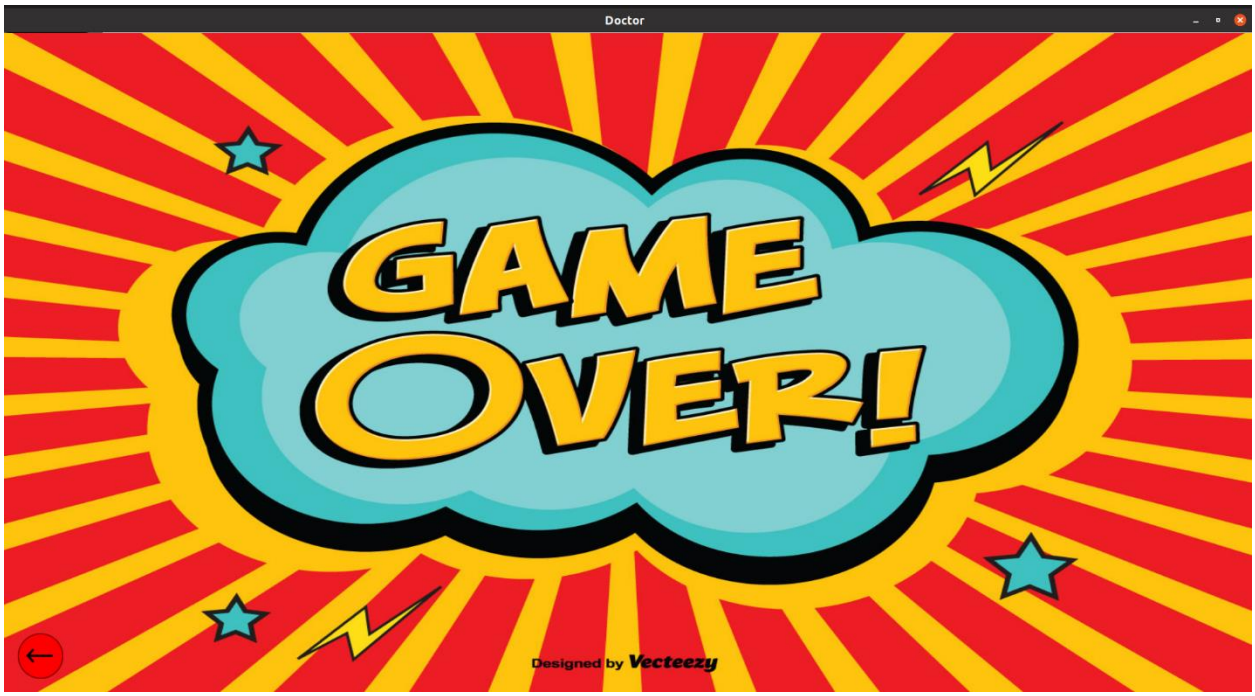
که در اینجا مستطیل هایی برای نمایش دارو و آمپول و همچنین سکه موجود است. در اینجا میتوانیم دارو و آمپول بخریم. و دکمه قرمز رنگ هم برای باز گشت به صفحه نخست است.

با کلیک بر روی دکمه بازی وارد صفحه (Treatment.qml) میشویم:



- برای در مان مریض درجه یک فقط به دارو نیاز است.
- برای درمان مریض درجه دو به آمپول و xp بالای ۴۰ نیاز است.
- اما برای درمان مریض شماره سه به دارو و آمپول و xp بالای ۸۰ نیاز است.

اگر در حین انجام بازی انرژی به صفر برسد:



Gameover اتفاق می افتد و با زدن دکمه بازگشت میتوان اطلاعات را در صفحه نخست تماشا کرد.

با تشکر

محمدرضا حیدرنیا-۹۹۱۲۳۵۸۰۱۸