

Hanoi University of Science and Technology



IT3103-143577 Object-oriented Programming

Kì 20231

Mini-Project

Demonstration of types of cell division

Giảng viên hướng dẫn: *PhD. Nguyen Thi Thu Trang*

Group 9

Student Name	Student ID
Nguyễn Văn Dũng	20215013
Vũ Đức Dũng	20194527
Đinh Huy Dương	20215020
Hoàng Đức Dương	20215021

I. Phân chia công việc

Công việc	Công việc quá trình	Thành viên
Use case diagram, General Class diagram	Ý tưởng	Nguyễn Văn Dũng
	Use case diagram	Vũ Đức Dũng
	General class diagram	Vũ Đức Dũng
	Detail class diagram	Vũ Đức Dũng
Main Menu Design	Main menu	Đinh Huy Dương
	Help Menu	Đinh Huy Dương
	Main action events	Nguyễn Văn Dũng
	Quit Alert	Nguyễn Văn Dũng
Cell Type Menu Design	Cell Type View	Đinh Huy Dương
	Cell Type Controller	Nguyễn Văn Dũng
	Back Button fixes	Vũ Đức Dũng
Division Type Menu Design	Division Type View	Đinh Huy Dương
	Division Type Controller – initial	Đinh Huy Dương
	Different case of Division for different Cell at Controller	Đinh Huy Dương
Media Player Design	Media player View	Đinh Huy Dương
	Set time stamps	Hoàng Đức Dương
	Media Player Controller	Hoàng Đức Dương
GUI design - Component view	Component view + controller for Prokaryotic	Đinh Huy Dương

	Component view + controller for Eukaryotic	Đinh Huy Dương
	Lambda expression for setButtonAction method	Đinh Huy Dương
Documents preparation	Eukaryotic and Prokaryotic Cell image	Nguyễn Văn Dũng
	Videos required for media player	Hoàng Đức Dương
	Editing videos (Video for media player + Demo video)	Đinh Huy Dương
	Components' name and functions – Prokaryotic	Nguyễn Văn Dũng
	Components' name and functions – Eukaryotic	Hoàng Đức Dương
Report and Slides	Report outline	Nguyễn Văn Dũng
	Report visual fixes	Nguyễn Văn Dũng
	Slides preparation - Contents	Vũ Đức Dũng

Nhóm có tham khảo ý tưởng từ project Cell Division

App tại github:

<https://github.com/hmkhoi2701/CellDivisionApp>

II. Mô tả Mini-Project

1. Yêu cầu: Một ứng dụng cho việc quan sát quá trình phân bào (Cell Division) và nghiên cứu các bộ phận tế bào (Cell Component)

App có các lựa chọn dành cho người dùng chọn chức năng (View Cell Component hoặc View Division), Help menu, và Quit.

- Help dùng để xem cách sử dụng app
- Quit dùng để thoát chương trình. App sẽ hỏi trước khi thoát
- View cell component:

Yêu cầu người dùng chọn loại tế bào (Eukaryotic hoặc Prokaryotic)

Sau đó sẽ chương trình sẽ hiện ra một biểu đồ cấu trúc tế bào tương ứng.

Khi xem cấu trúc, người dùng có thể nhấp vào tên của thành phần để xem tên và chức năng của thành phần đó.

Trong suốt quá trình, luôn có một nút "Back" để người dùng quay trở lại.

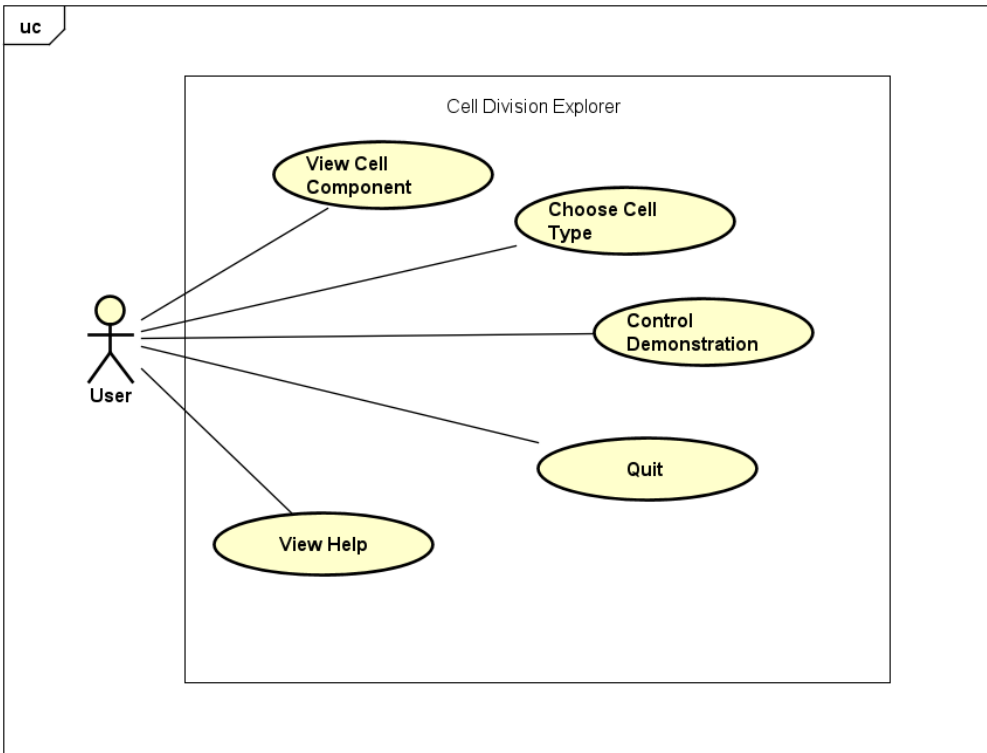
- View cell division:

Yêu cầu người dùng chọn loại tế bào, sau đó chọn phương pháp phân chia liên quan đến loại tế bào đó.

Sau đó, một trình phát phương tiện sẽ xuất hiện tương ứng với phương pháp phân chia. Người dùng sẽ có thể dừng, tiếp tục, hoặc nhảy đến các giai đoạn tiếp hoặc trước (phase) của quá trình phân bào, và Replay.

Trong suốt quá trình, luôn có một nút "Back" để người dùng quay trở lại.

2. Use case Diagram

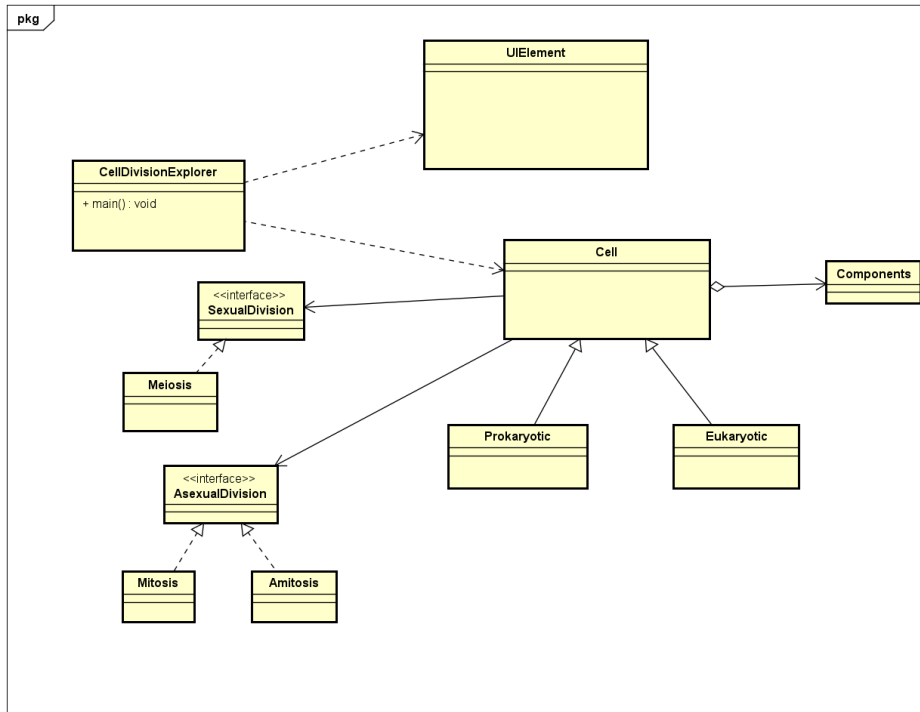


Đầu tiên, phần mềm sẽ hiển thị hai loại tế bào cho người dùng: tế bào nhân sơ và tế bào nhân thực. Người dùng tương tác với phần mềm bằng cách chọn một trong chúng (Choose Cell Type).

1. Xem Các Bộ Phận của Tế Bào (View Cell Component): Phần mềm sẽ hiển thị các thành phần của loại tế bào và người dùng có thể nhấp vào các thành phần để tìm hiểu tên, hình ảnh và chức năng của chúng.
2. Xem Quá Trình Phân Chia của Tế Bào (Control Demonstration): Phần mềm sẽ hiển thị quá trình phân chia của loại tế bào mà người dùng đã chọn.
3. “View Help”: Hiển thị hướng dẫn sử dụng cho người dùng.
4. "Quit": Yêu cầu xác nhận từ người dùng về việc họ muốn rời khỏi/ tiếp tục ở trong ứng dụng. Sau đó đóng/ giữ nguyên cửa sổ hiện tại.

III. Thiết kế (Design) – Class Diagram

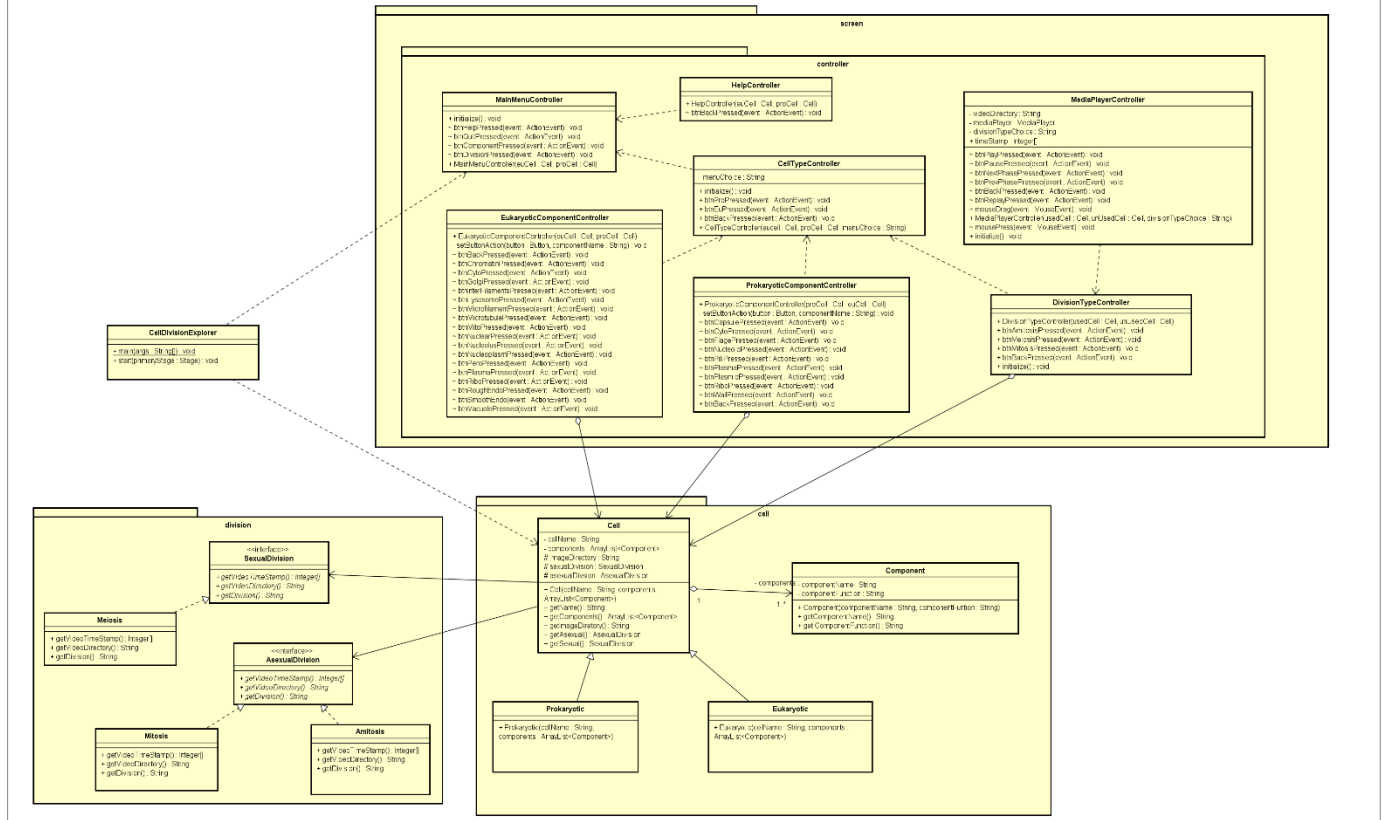
1. General Class Diagram



- Thiết kế tập trung vào đối tượng chính là tế bào, với một tế bào có thể được phân ra thành nhân sơ (Prokaryotic Cell) hoặc nhân thực (Eukaryotic Cell). Vậy nên hiển nhiên một lớp tế bào (Cell) sẽ có 2 lớp con để kế thừa là Prokaryotic và Eukaryotic.
- Tiếp theo tế bào sẽ bao gồm các bộ phận (Component), vậy nên lớp Cell sẽ mang quan hệ kết tập (Aggregation) với lớp Component.
- Một tế bào có thể phân chia (Division). Nhóm đã quyết định phân ra 2 loại phân bào, đó là phân bào theo kiểu sinh sản vô tính (Asexual Reproduction Division) và sinh sản hữu tính (Sexual Reproduction Division). Khi đó 2 kiểu phân bào này sẽ là một Interface, và các lớp phân bào cụ thể sẽ có thể Implement Interface này để có thể cụ thể hóa các quá trình phân bào của chính các phân bào (Override method). Như ta thấy trong sơ đồ, Interface AsexualDivision sẽ được implement bởi lớp Amitosis (Trực phân) và Mitosis (Nguyên phân). và tương tự Interface SexualDivision sẽ được implement bởi lớp (Meiosis).
- Điểm đặc biệt trong sơ đồ lớp này, mà nhóm 9 đã học hỏi được từ nguồn tham khảo, đó chính là việc lớp Cell không implement các Interface, mà thay vào đó sẽ có mối quan hệ Association (liên kết), và cụ thể hóa là việc sử

dụng AsexualDivision và SexualDivision là một thuộc tính (Attribute) của Cell, chứ không Override các phương thức (method) của 2 Interface này. Lúc này mỗi Cell sẽ bao gồm 2 thuộc tính asexual, và sexual, và được để ở trạng thái (visibility) Protected để chỉ có lớp Cell được sử dụng (sẽ được trình bày ở phần Detail Class Diagram).

- Lớp chính là CellDivisionExplorer, là nơi chương trình chạy, sử dụng Cell và thành phần UI (UI Element). Nhóm sử dụng công cụ thiết kế UI là JavaFX (được nói rõ trong phần sau).



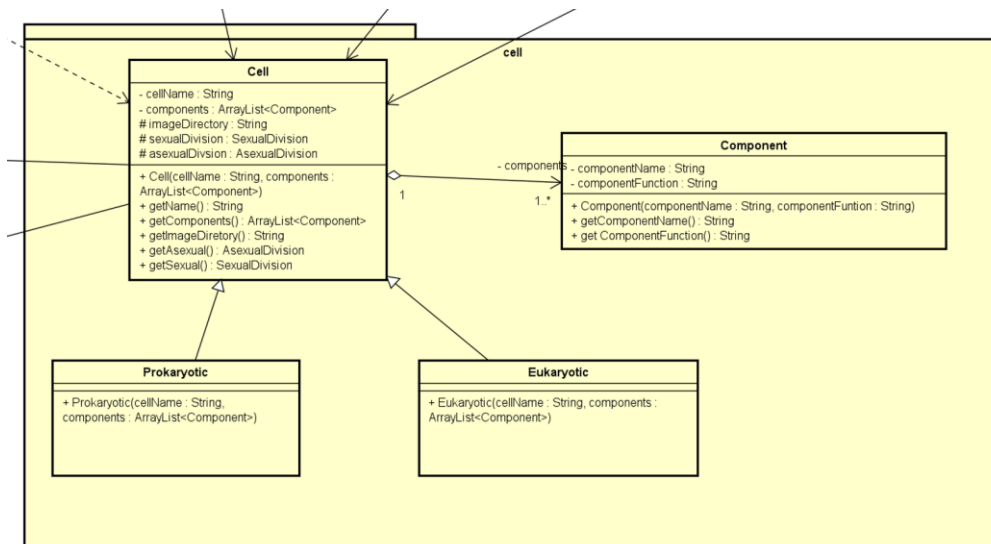
2. Detail Class Diagram

Nhóm đã áp dụng mô hình MVC để phù hợp với một ứng dụng JavaFX. Ta thấy rõ trong Class Diagram, trong package screen, là package controller chứa các Controller cho các View được thiết kế làm màn hình. và package cell với division sẽ là phần Model, lõi của chương trình.

Cụ thể, chương trình sẽ là việc trình chiếu các hình ảnh, video được đóng gói trong các lớp của phần Model để hiện ra cho phần View, nên trong các lớp như Cell, hay các lớp trong division sẽ có chứa các đường dẫn, hoặc

Chúng ta sẽ đi sâu vào phân tích các package:

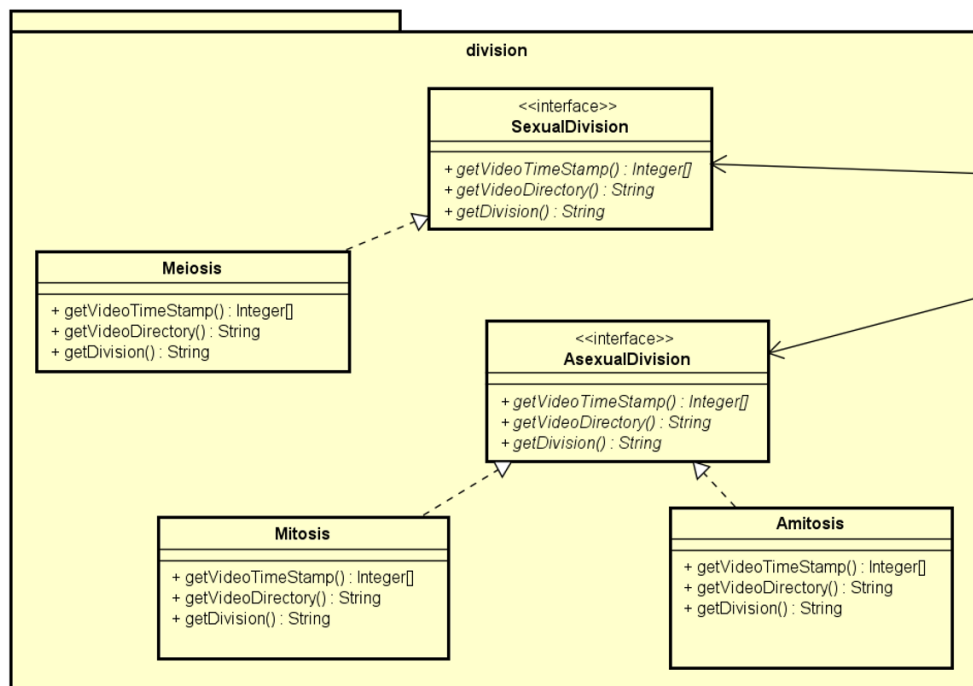
a. Package cell:



Ta có thể phân tích những kỹ thuật OOP đã được áp dụng:

- **Trừu tượng hóa (Abstraction):** Ta chỉ trích xuất những thông tin của một tế bào trên một khía cạnh (view) nào đó. Cụ thể ở đây, trong lớp `Cell` bao gồm tên và đường dẫn đến hình ảnh của loại tế bào, và 2 thuộc tính của việc phân bào để cập phần trước ở `protected`: `asexualDivision`, `sexualDivision`.
- **Đóng gói (Encapsulation):** Việc đóng gói thực hiện để cho các lớp còn lại không thể sử dụng các thông tin thuộc tính của lớp, mà chỉ qua các phương thức getter (Accessor). Đó là `cellName`, mảng `components`, và các thuộc tính của `Component` được để ở `private`; `imageDirectory`, `sexualDivision`, `asexualDivision` đặt ở `protected` để chỉ có các lớp con của `Cell` (`Eukaryotic`, `Prokaryotic`) có thể sử dụng.
- **Kết tập (Aggregation):** Với việc một lớp có quan hệ chứa là một thành phần của lớp khác (Has – a), lớp `Cell` sẽ mang kết tập các đối tượng của lớp `Component` để tạo thành mảng `components`.
- **Kế thừa (Inheritance):** Các lớp con của `Cell` là `Prokaryotic` và `Eukaryotic` sẽ kế thừa các thuộc tính và phương thức của lớp cha để sử dụng lại tài nguyên code.
- **Đa hình (Polymorphism):** `Prokaryotic` và `Eukaryotic` sẽ có `asexualDivision` và `sexualDivision` khác nhau. Ví dụ, `Eukaryotic` đặt `asexualDivision` là `Mitosis` còn `Prokaryotic` đặt là `Amitosis`.

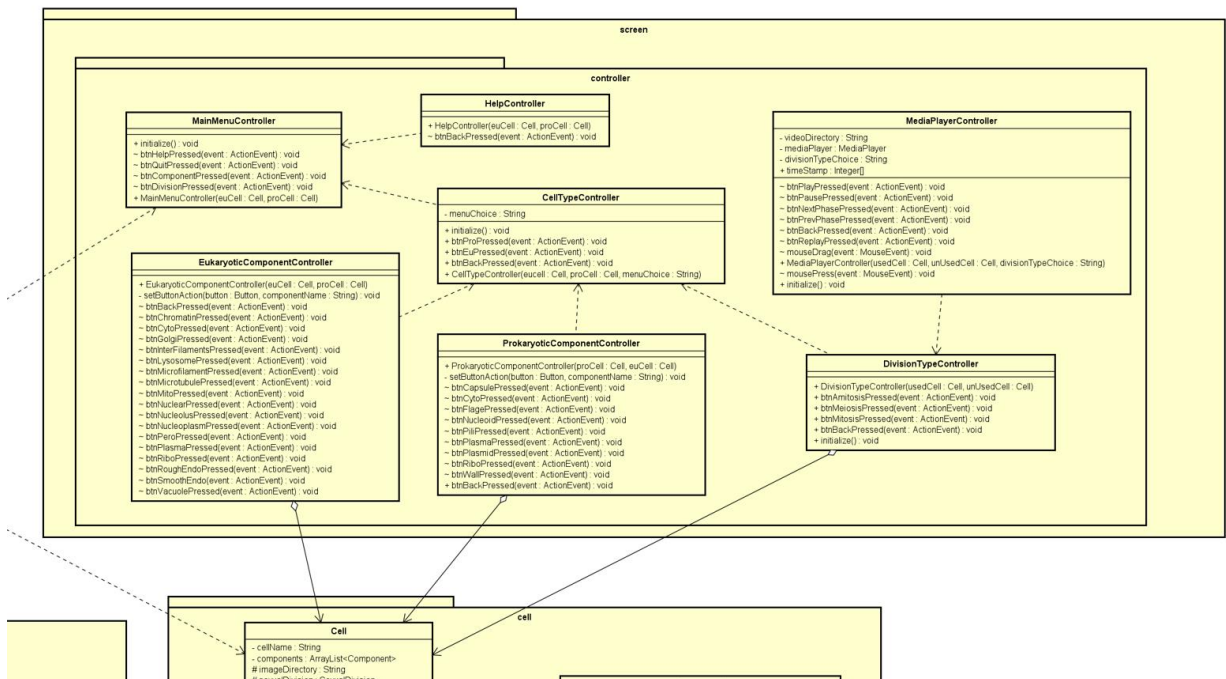
b. Package division:



Nguyên lý OOP được sử dụng rõ ràng có thể thấy trong đây là **Đa hình (Polymorphism)**, khi các lớp được sử dụng các Interface có thể biến đổi các phương thức để linh hoạt sử dụng. Cụ thể ở đây:

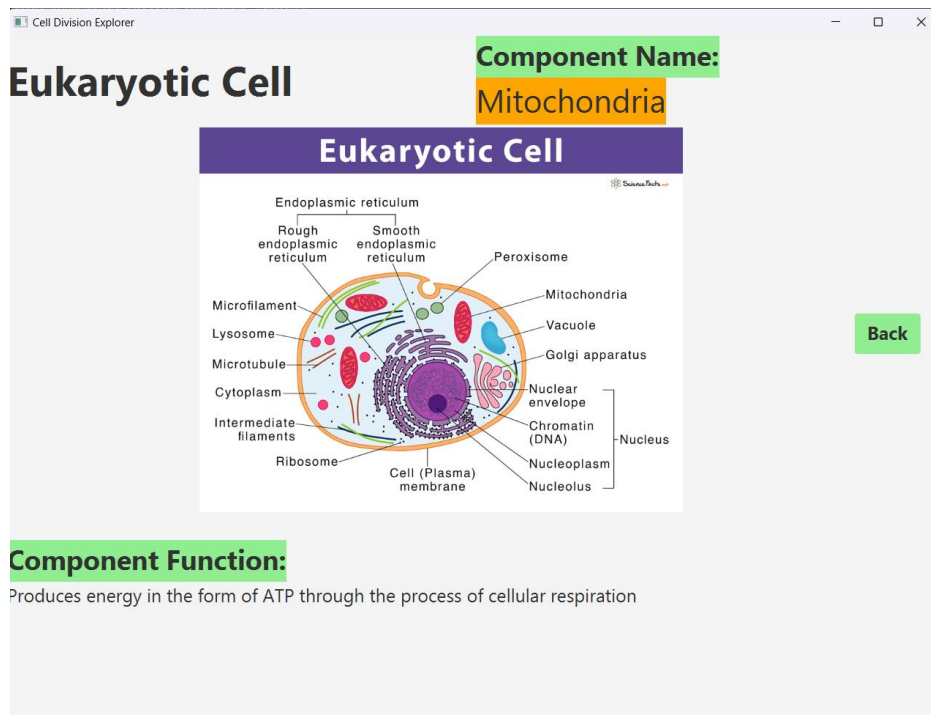
- `getVideoTimeStamp()`: trả về mảng chứa các mốc thời gian của các giai đoạn (Phase) trong quá trình phân bào,
- `getVideoDirectory()`: trả về đường dẫn đến video, được lưu trong thư mục `resource/video` ở thư mục `src` của Project.
- `getDivision()`: trả về tên của loại phân bào, ví dụ “Mitosis”, “Amitosis”,...

c, Package Controller:



Ta có thể thấy rõ ở đây các lớp Controller sẽ có quan hệ kết tập (Aggregation) với Cell, nghĩa là trong lớp này đang chứa một thành phần Cell để sử dụng cho trình diễn. Cụ thể:

- Các Constructor của mọi lớp Controller đều phải chứa 2 tế bào: Prokaryotic Cell và Eukaryotic Cell.
- Chương trình sẽ bắt đầu ở MainMenuController, rồi đến CellTypeController. Trong Constructor của CellTypeController sẽ lưu lại một biến String: menuChoice để lưu lại giá trị đã được chọn trước để có thể quyết định được đã chọn chức năng nào trong Main Menu (View Component hay View Division)
- Sau màn hình của CellTypeController sẽ phụ thuộc vào giá trị của menuChoice để tới DivisionTypeController (Chọn kiểu phân bào), hoặc xem các thành phần tế bào ở 2 lớp EukaryoticComponentController hoặc ProkaryoticComponentController. Khi đó, vì đã chọn 1 loại tế bào, nên constructor sẽ bao gồm usedCell và unUsedCell tùy vào kiểu của Cell đã chọn bằng phương thức instance of.

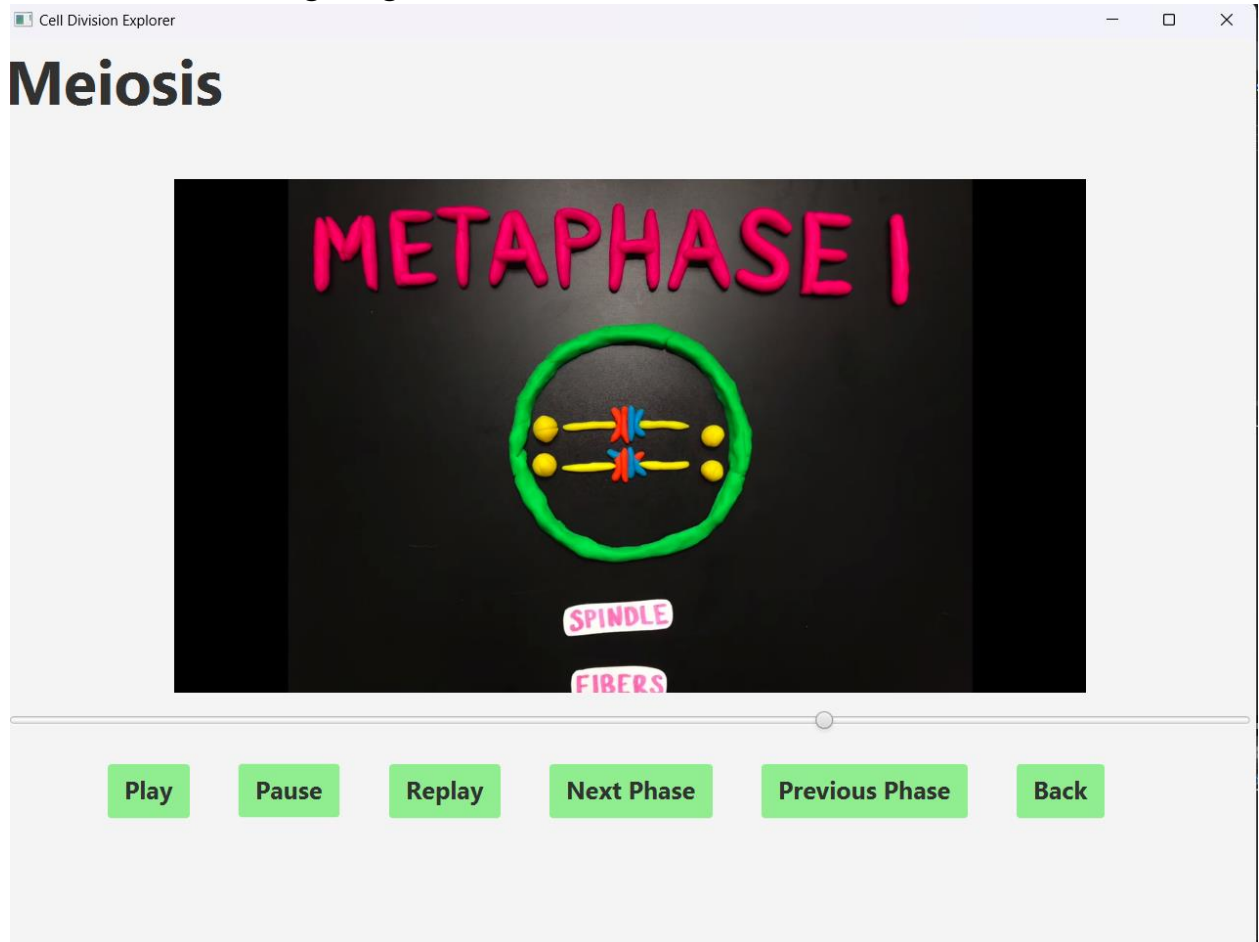


- Lý do cần có `usedCell` và `unUsedCell` là do để quay lại màn hình trước ta cần có đủ 2 kiểu tế bào, vậy nên vẫn cần phải lưu lại.
- Trong quá trình xem các bộ phận ở các lớp `ComponentController`, ta sử dụng các nút bấm được đặt ở `Anchor Pane` (tự do trong khung) và sử dụng phương thức `setButtonAction(button, componentName)` để có thể hiện tên và chức năng dựa vào tìm kiếm thành phần trong mảng `components` của `Cell`. Cụ thể hơn, ở đây có sử dụng một hàm `lambda`:

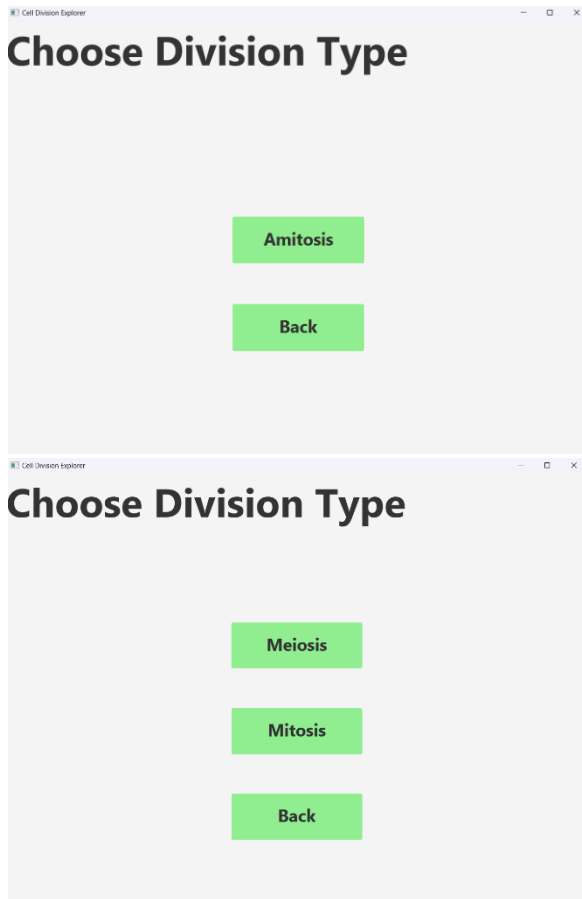
```
private void setButtonAction(Button button, String componentName) {
    button.setOnAction(event -> {
        ArrayList<Component> components = ((Prokaryotic)
usedCell).getComponents();
        for (Component component : components) {
            if (component.getComponentName().equals(componentName))
            {
                lblFunction.setText(component.getComponentFunction());
                lblName.setText(component.getComponentName());
                break;
            }
        }
    });
}
```

Với 1 sự kiện `event`, ta có thể thực hiện được chuỗi lệnh sau “->”.

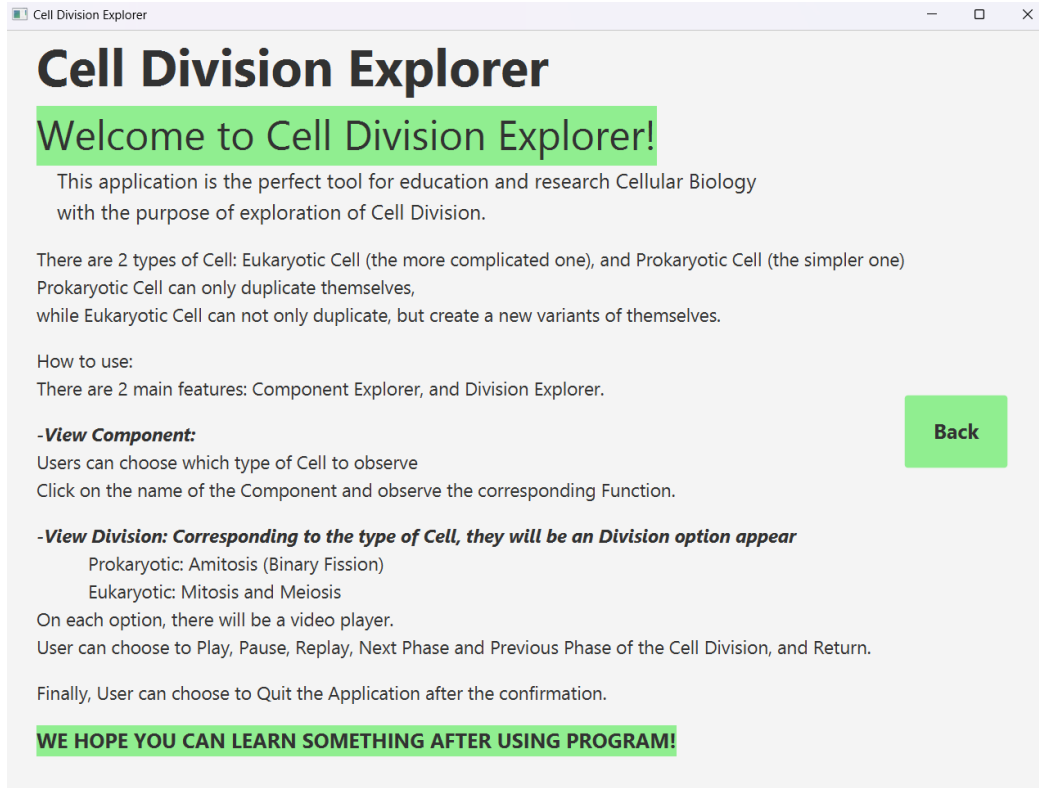
- Sau DivisionTypeController là MediaPlayerController cho người dùng xem video của phân bào, ở đó sử dụng thư viện media của JavaFX, và cụ thể hơn trong các phương thức có btnNextPhasePressed và btnPrevPhasePressed để có thể nhảy đến giai đoạn tiếp/ trước của quá trình, bằng việc đối chiếu với các giá trị trong mảng timeStamp, được lấy từ thuộc tính asexualDivison và sexualDivision trong từng Cell.



- Mỗi Controller sẽ bao gồm một phương thức btnBackPressed để quay lại về màn hình trước, cũng như một vài Controller có Override lại initialize() để có thể phù hợp, ví dụ sau khi chọn Prokaryotic, thì chỉ có một phương thức phân bào là Amitosis, nên sẽ chỉ hiện ra Amitosis và Back ở DivisionType, ngược lại Eukaryotic sẽ chỉ bao gồm Mitosis, Meiosis và Back.



- Ngoài ra còn có màn hình trợ giúp Help:



IV. Đánh giá và Hạn chế

- Chương trình đặt ra đã thỏa mãn được yêu cầu của các Use-case, và đã có sự áp dụng của một vài kỹ thuật trong OOP
- Chương trình phụ thuộc nhiều vào nền tảng của JavaFX, khiến cho việc thêm thư viện vào để chạy được trở nên khó khăn
- Trong Media Player, các video đôi lúc load vào rất chậm chạp, phải Back xong quay lại mới hiện ra
- Trong quá trình View Component, các nút bấm trên màn hình đôi lúc phải bấm 2 lần mới hiện ra chức năng và tên
- Việc mở rộng màn hình không áp dụng được cho nút Back (Do kích thước lúc initialize đã cố định).

