



Daffodil
International
University

Life At Cadet College

A Third Person Virtual Experience Based on Cadets Life

By

Md. Nahasat Hossain Nibir
(161-35-1470)

&

Supervisor: Asif Khan Shakir
Lecturer & Member
Project/Thesis Committee

A project submitted in partial fulfillment of the requirement for the
degree of Bachelor of Science in Software Engineering

Department of Software Engineering
DAFFODIL INTERNATIONAL UNIVERSITY

Fall– 2019

APPROVAL

This **Project** titled “**Life At Cadet College**”, submitted by **Md.Nahasat Hossain Nibir,161-35-1470** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc in Software Engineering and approved as to its style and contents.

BOARD OF EXAMINERS

Dr. Touhid Bhuiyan
Professor and Head

Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Chairman

Dr. Md. Asraf Ali
Associate Professor

Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1

Asif Khan Shakir
Lecturer

Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

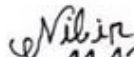
Dr. Md. Nasim Akhtar
Professor

Department of Computer Science and Engineering
Faculty of Electrical and Electronic Engineering
Dhaka University of Engineering & Technology, Gazipur

External Examiner

Declaration

I hereby declare that I have taken this project under the supervision of **Mr. Asif Khan Shakir, Lecturer & Member Project/Thesis Committee, Department of Software Engineering, Daffodil International University.** I also declare that neither this project nor any part of this has been submitted elsewhere for award of any degree.


11.12.19

Md. Nahasat Hossain Nibir
ID: 161-35-1470
Batch: 19th
Department of Software Engineering
Daffodil International University

Certified by


11.12.19

Mr. Asif Khan Shakir
Lecturer & Member
Project/Thesis Committee
Department of Software Engineering
Daffodil International University

Acknowledgement

It stands to reason that the completion of main project needs the support of many people. I take this opportunity to express my boundless thanks and commitment to each and every one, who helped me in successful completion of my main project. I am happy to acknowledge the help of all the individuals to fulfill my attempt.

Most importantly I wish to express wholehearted obligation to my creator **ALLAH** for his thoughtful steady care and unselfishness showered euphorically over me during this Endeavor.

I would like to take the opportunity to express my gratitude to **Mr. Asif Khan Shakir, Lecturer & Member, Project/Thesis committee. Department of Software Engineering, Daffodil International University.** Although he was always loaded with several other activities, he gave me more than enough time in this work. He not only gave me time, but also gave me proper guidance and valuable advice whenever I faced with some difficulties. His comments and guidance helped me in preparing my project report.

Finally, I thank to our teachers for supporting us throughout all our studies at Daffodil International University.

Contents

1.Introduction.....	1
1.1 Project Overview.....	1
1.2 Project Purpose.....	1
1.2.1 Background.....	1
1.2.2 Benefits & Beneficiaries.....	1
1.2.3 Goals.....	2
1.3 Stakeholders	2
1.4 Proposed System Model (Block Diagram).....	3
1.5 Project Schedule.....	5
1.5.1 Gantt Chart	5
1.5.2 Release Plan/Milestone	5
2. Software Requirement Specification	6
2.1 Functional Requirements.....	6
2.1.1 Walk.....	6
2.1.2 Camera Rotation	7
2.1.3 Take A Seat.....	7
2.1.4 Read Text	7
2.1.5 UI Interaction	8
2.1.6 Talk	8
2.1.7 Overlap with Collision Boxes.....	8
2.2 Data Requirements.....	9
2.3 Performance Requirements.....	9
2.3.1 Speed and Latency Requirements	9
2.3.2 Capacity Requirements.....	9
2.4 Maintainability and Support ability Requirements	10
2.4.1 Maintenance Requirements.....	10
2.5.1 Support ability Requirements	10

2.5 Security Requirements	10
2.6 Usability and Human-Interaction Requirements.....	10
2.6.1 Ease of Use Requirements	10
2.7 Look and Feel Requirements.....	11
3. System Analysis	12
3.1 Use Case Diagram.....	12
3.2 Use Case Description.....	13
3.3 Activity Diagram	19
3.3.1 Asking for Admit Card.....	19
3.3.2 Showing Admit Card	20
3.3.3 Briefing About Exam	21
3.3.4 Providing Exam Paper	22
3.3.5 Interacting With UI	23
3.3.6 Asking Questions.....	24
3.3.6.1 Doctor	24
3.3.6.2 Army	25
3.3.6.3 Civil 1	26
3.3.6.4 Civil 2	27
3.3.7 Answering Questions.....	28
Fig: Answering Questions	28
3.3.8 Seating on Chair	29
3.3.9 Calculating Marks.....	30
3.3.10 Keeping Track of User Progress.....	31
3.3.11 Talking with Parents	33
3.3.12 Talking with House Master	34
3.4 System Sequence Diagram.....	35
3.4.1 Asking for Admit Card.....	35
3.4.2 Showing Admit Card	36
3.4.3 Briefing About Exam	37

3.4.4 Providing Exam Paper	38
3.4.5 Interacting With UI	39
3.4.6 Asking Questions	40
3.4.7 Answering Questions	41
3.4.8 Seating on Chair	42
3.4.9 Calculating Marks	43
3.4.10 Keeping Track of User Progress	44
3.4.11 Talking with Parents	45
3.4.12 Talking with House Master	46
4. System Design Specification	47
4.1 Class Diagram of Classes	47
4.2 Relationships among classes	56
4.3 Database Design Diagram	57
4.4 Development Tools & Technology	58
5. System Testing	58
5.1 Testing Features	58
5.1.1 Features to be tested	58
5.1.2 Features not to be tested	59
5.2 Testing Strategy	59
5.3 Testing Environment	59
5.4 Test Case	60
6. Project Summary	61
6.1 GitHub Link	61
6.2 Critical Evolution	61
6.3 Limitations	61
6.4 Future Scope	62

1.Introduction

1.1 Project Overview

This project is a 3D computer game based on Virtual Reality Concept that represents the struggle and day to day activities that happens in a cadet's life.

1.2 Project Purpose

The proposed game product is a Real time third person virtual experience. Where the users will be able to face and meet the challenges that a cadet graduate faces during his life. The main purpose is to let the non-cadets experience cadet's life by creating an indoor virtual reality-based entertainment.

1.2.1 Background

This is not merely a virtual tour or walk through rather it's a 3D game where the users will be able to interact with 3D objects and gain virtual experience where things are basically simulated via programming and AI based. The users will have to follow the rules and do task as scheduled so that they can realize what happens or what a cadet student needs to do during his college life in a cadet college. From admission to graduation is the full specification.

1.2.2 Benefits & Beneficiaries

As of speaking "Benefits", by developing this product, us developers will be able to create a source of passive income as well as to offer our end users a new way of experiencing cadet's life.

For the part "Beneficiaries", they will be able to experience what actually happens in a cadet's life and also who wasn't able to get into cadets' college. It's like a "dream comes to true" to them. Besides virtual challenging 3D environments will help developing their brains thinking capacity.

1.2.3 Goals

Our Goal is to bring a new franchise in the field of Gaming Industry. Also, to create new scopes in Bangladesh's Game Development Sector. Apart from that, developing a way to experiencing real world scenarios that happens in our life.

1.3 Stakeholders

1. The game developers are the main stake holders since they are also gamers.
2. People who loves to play games a lot.
3. People who wanted to get into cadet colleges but couldn't make it.
4. People who are already studying into cadet colleges.
5. Teachers who teach at cadet colleges.
6. People who graduated from cadet colleges.
7. People who are interested into virtual reality.
8. People who do research on students/cadets' life.
9. People who just wants to experience a cadet's life out of curiosity.

1.4 Proposed System Model (Block Diagram)

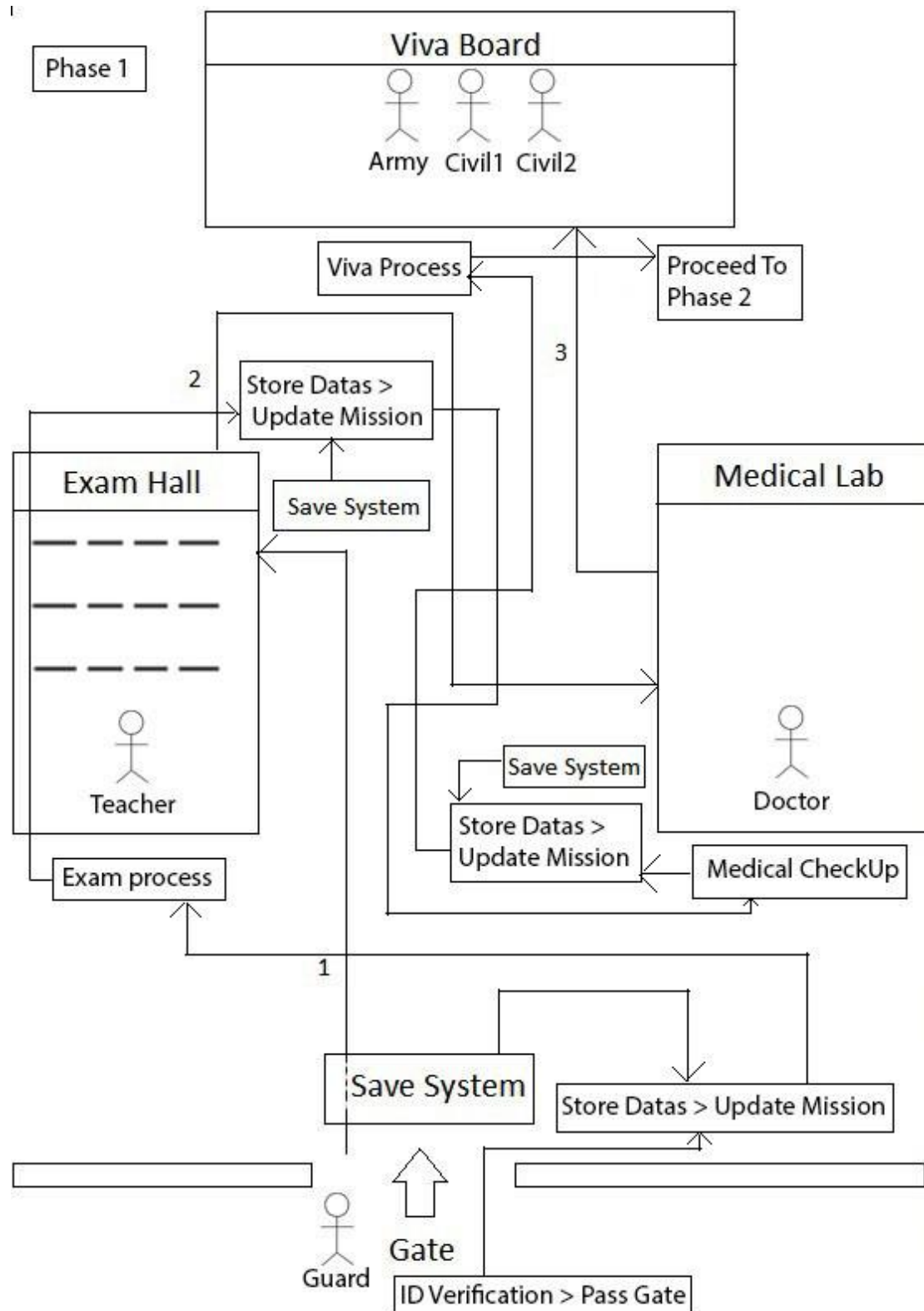


Fig: Block Diagram 1

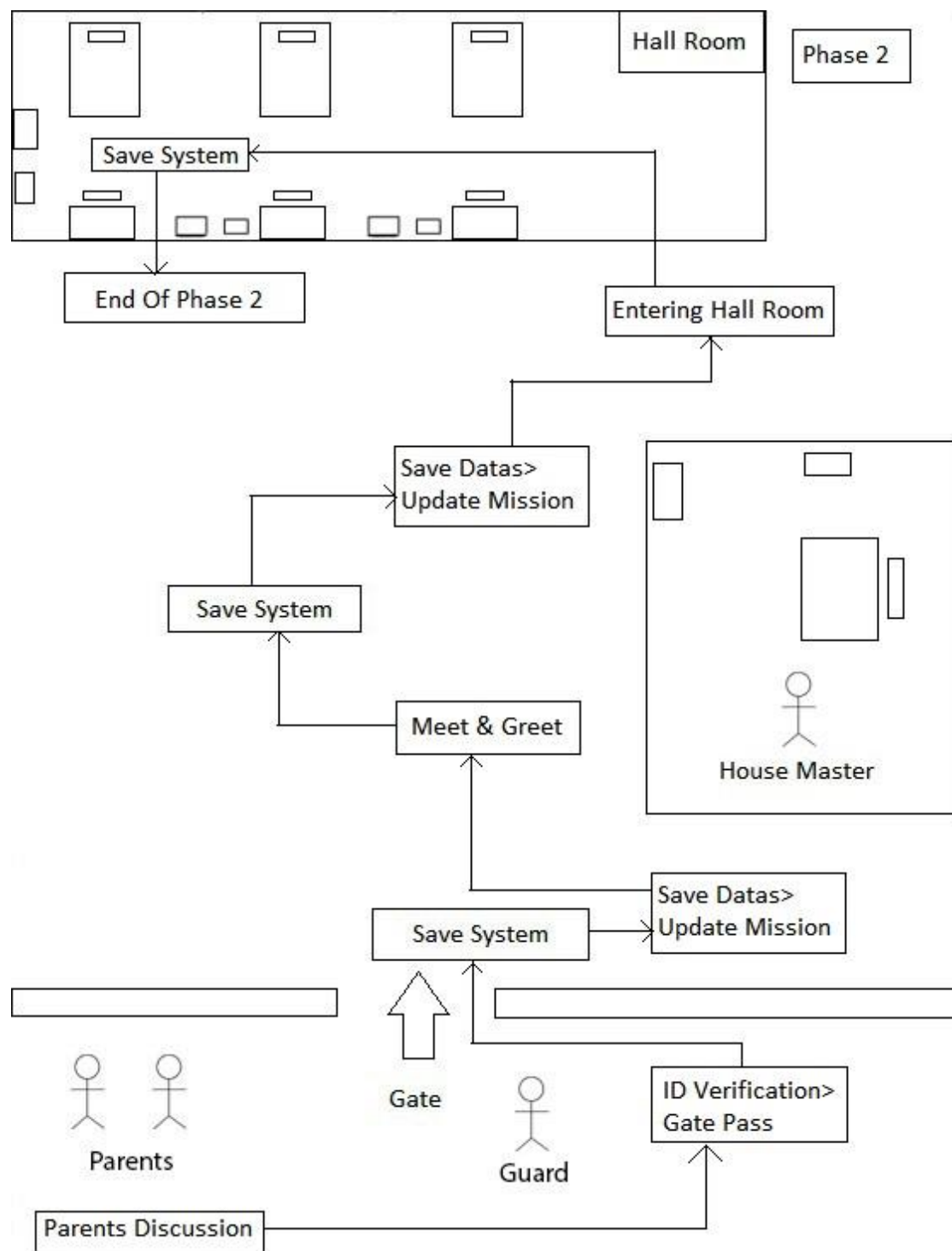


Fig: Block Diagram 2

1.5 Project Schedule

This part mainly focuses on the time duration and work schedule.

1.5.1 Gantt Chart

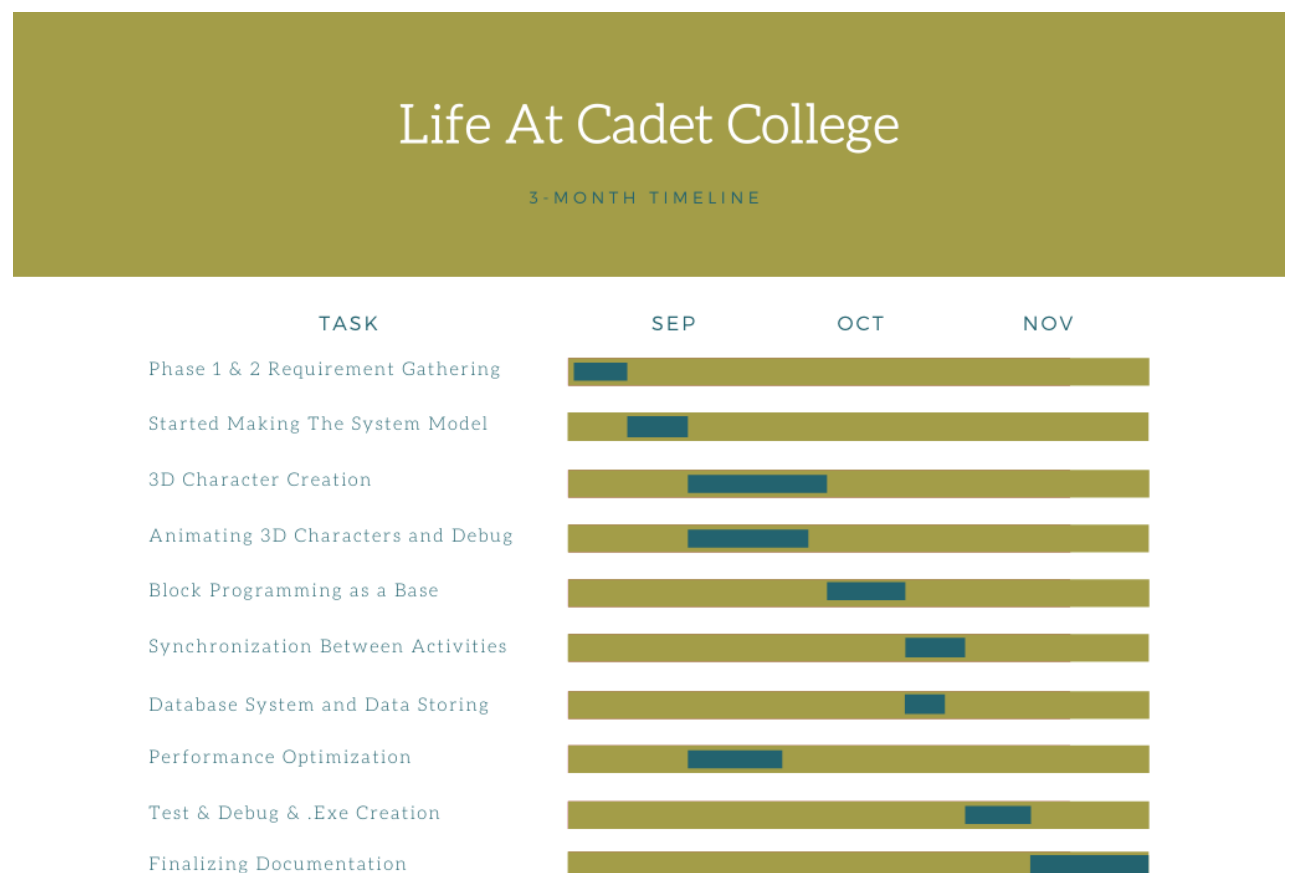


Fig: Gantt Chart

1.5.2 Release Plan/Milestone

We initially made Phase 1 & 2. Those Phases are totally complete but subject can be changed later based on fun factors. Now we are focusing on Developing phase 3 which is the First day of a cadet in Cadet college. The total planning for Phase 3 has been visualized in the below release plan diagram.

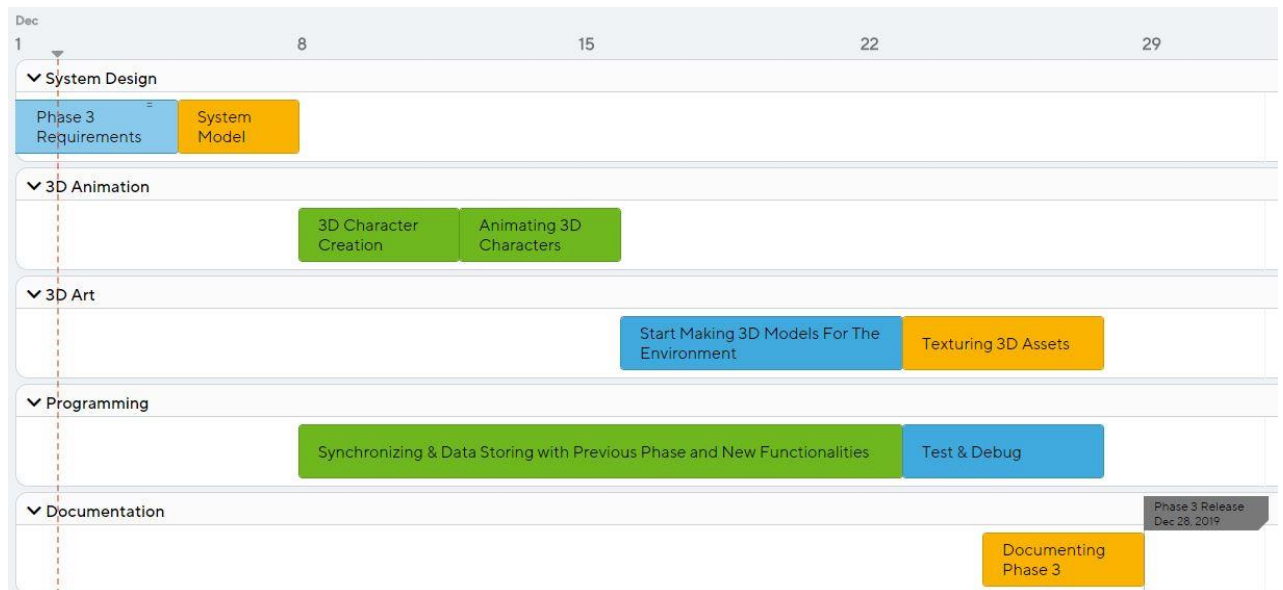


Fig: Release Plan/Milestone

2. Software Requirement Specification

2.1 Functional Requirements

Before distinguishing proof of the necessities, we required the exhaustive commitment and lighting fast coordination with the partners. This quickens the whole necessities the board procedure by arranging the progression of data and procedures crosswise over various colleagues and partners. Again, this is joined with half and half nimble and cascade advancement techniques and devices. Adaptable work processes and programmed warnings streamline correspondence, survey, and endorsement of prerequisites crosswise over partners, while normal measurements and dashboards guarantee everybody is in agreement. Along these lines, the recorded necessities go with all the past procedures.

For Phase 1 & 2 We only need these functional Requirements.

2.1.1 Walk

Walking is the most important task of player where the key role is to be able to go to any destination.

Requirement No.	Requirement
W-001	Be able to press something on the keyboard
W-002	Based on the pressed button the player body will be notified
W-003	It will then start walking like in real life in the 3D Space
W-004	It will take direction data based on mouse interaction
W-005	On undressing the button, the player will stop walking

2.1.2 Camera Rotation

Camera rotation is very important since our player is in a 3D space and need to rotate to change direction.

Requirement No.	Requirement
CR-001	Be able to move the mouse
CR-002	Based on the mouse sensitivity the camera will receive input value
CR-003	Camera will start rotating based on the received data

2.1.3 Take A Seat

The player needs to take a seat during exam and viva and other moments like these.

Requirement No.	Requirement
TS-001	Be able to press something on the keyboard
TS-002	Based on the pressed button the player will take a seat
TS-003	While seating a “taking a seat “animation will be played

2.1.4 Read Text

Be able to read messages and information presented on the UI

Requirement No.	Requirement
RT-001	Be able to press something on the keyboard
RT-002	Based on the pressed button the message will be popped up
RT-003	The player needs to be in front of the message/text that he wants to read
RT-004	The message will be shown in an UI
RT-005	Closing will close the message

2.1.5 UI Interaction

Player needs to interact with UI

Requirement No.	Requirement
UI-001	Be able to click using the mouse
UI-002	Based on the click the UI will receive input
UI-003	The player needs to be in front of the event where UI's will pop up
UI-004	Data's are shown in UI and manipulates based on mouse interaction
UI-005	Closing will close the UI

2.1.6 Talk

Player Talks with NPC's (Non-Playable Characters) while interacting with them.

Requirement No.	Requirement
T-001	Be able to interact with NPC's
T-002	After interaction the player starts talking
T-003	During this process the player shows talking animation.
T-004	After the end of conversation, the player stops talking

2.1.7 Overlap with Collision Boxes

Requirement No.	Requirement
OCB-001	Be able to overlap with collision boxes

OCB-002	After overlapping, events will fire
OCB-003	We need to make sure the players collision is working
OCB-004	Event wont fire if the collision responses aren't set properly

2.2 Data Requirements

If the player played the game and then stopped playing at a certain point and closes the game then his data's will be stored on the last checkpoint. So when the player starts playing the game again and chooses Resume option from the menu then he will be able to play from the last checkpoint where he stopped playing. That's why we need to store those data's properly otherwise we will have to play from the beginning all the time.

We need those last played data to resume from the last point properly.

2.3 Performance Requirements

Since this is a 3D computer game, performance is a major topic to look at.

2.3.1 Speed and Latency Requirements

The more Ram + GPU + CPU the players have the better the performance will be. But Players without GPU will face a hard time playing this game since most of the 3D assets are processed using the GPU. But CPU's with built in graphics will be able to play. With that said those CPU's will have to be powerful and built in graphics integrated.

2.3.2 Capacity Requirements

We cannot give the exact data about the size of the game and how much the HDD/SDD capacity is needed to store the game. Because this is not the full game yet rather its only Phase 1 & 2.

But since modern computers have enough storage data's so it won't be that of a troublesome.

2.4

Maintainability and Support ability Requirements

2.4.1 Maintenance Requirements

Sine this is not an online game there is no need for maintenance for server errors or connectivity issues.

2.5.1 Support ability Requirements

As it's an offline game and there maybe bugs or glitches. So we may provide patches or updates if needed.

2.5 Security Requirements

Since this is not an online game there is no possibility on cracking players data and using it for cheating. But for offline games there is obviously ways for cracking games and using it for fun.

That's a major threat in the game industry. Since this is not a Big budget game rather it's a fun project, we are thinking of releasing it for free. So even if people crack it there won't be any issues.

2.6 Usability and Human-Interaction Requirements

2.6.1 Ease of Use Requirements

This is a 3D computer game and it needs to be user friendly so that the players can enjoy playing it without any problem. The UI will be much user friendly because a game is supposed to entertain a player to its max level. If the usability isn't friendly enough then success rate of a game is very low.

For this we will use standard keyboard + mouse interactions that is accepted globally and is massively used in most modern games.

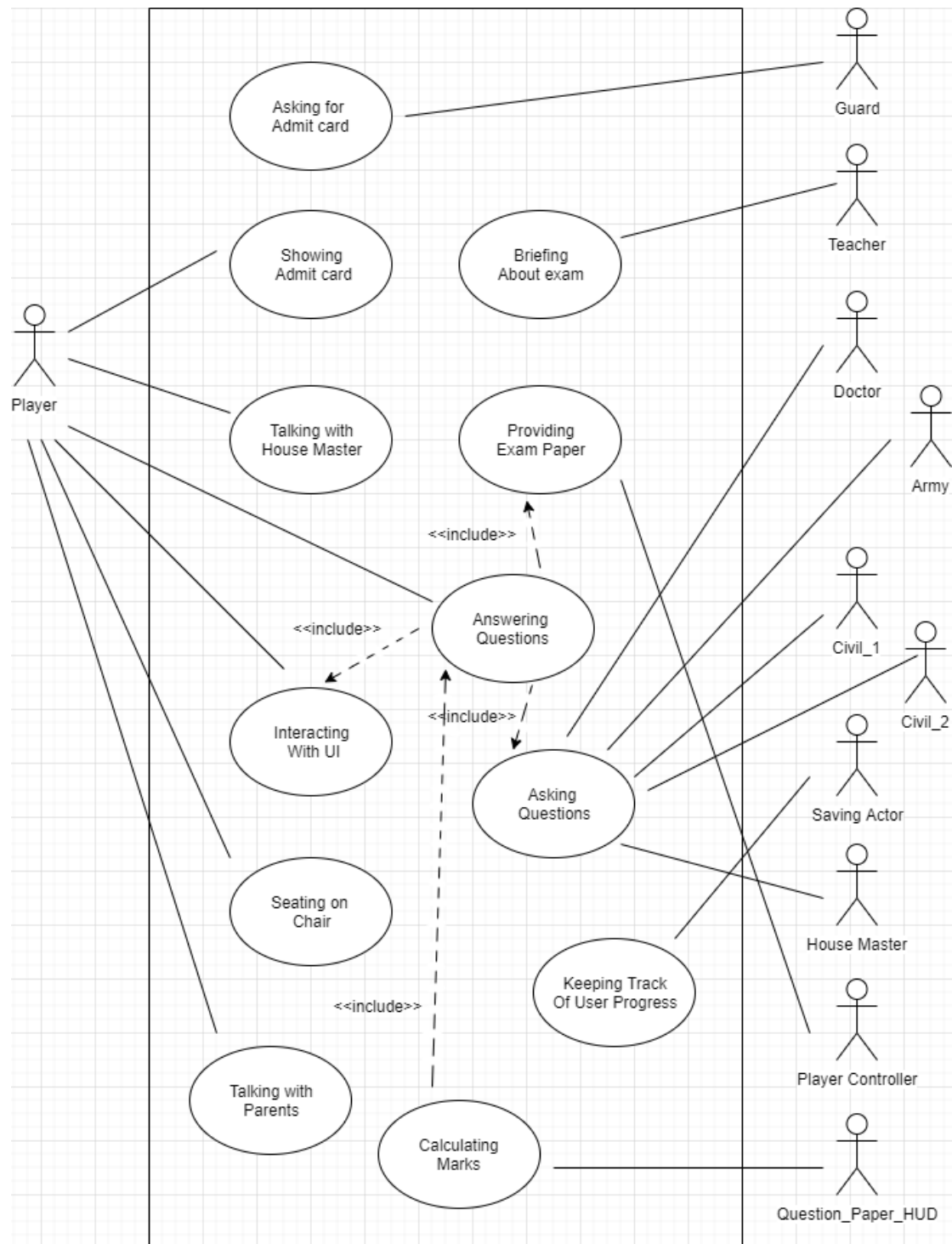
2.7 Look and Feel Requirements

3D world needs to be as beautiful as it can be so that it attracts peoples mind as well as the players. For this we will use real life references to make the 3D world as real as it can be.

If we can manage to make it look realistic with the power of modern technology and using modern graphics processing systems then this game will be one of the greatest franchises ever made.

3. System Analysis

3.1 Use Case Diagram



3.2 Use Case Description

Use Case No.	1
Use Case Name	Asking for admit card
13	Guard
Description	Guard asks for admit card while the player wants to pass the gate.
Precondition	The player must come near the guard.
Trigger	Press E for starting the process
Flow of Events	1. When the player comes in front the guard the cinematic start which is asking the player about his admit card. 2. It will only happen when the player touches the the guards notifying collision.
Post Condition	Guard stops showing asking admit card animation and returns to its Idle animation.

Use Case No.	2
Use Case Name	Showing admit card
Actor	Player
Description	Player shows admit card to the guard for passing the gate.
Precondition	The player must stay in front of the guard and touch the guards notifying collision
Trigger	Player needs to press E to show admit card
Flow of Events	1. When the player presses E in front of guard, player starts showing 'giving admit' card animation
Post Condition	The player should move away from the guard after showing admit card.

Use Case No.	3
Use Case Name	Briefing about exam
Actor	Teacher
Description	Teacher starts briefing about the exam rules.
Precondition	The player must take his seat in the classroom.
Trigger	The teacher starts briefing after the player takes his seat.
Flow of Events	The teacher starts briefing after the player seats and continues giving instructions.
Post Condition	Teacher will return to its idle state after exam finishes

Use Case No.	4
Use Case Name	Providing Exam Paper
Actor	Player Controller
Description	Player Controller adds question paper HUD to screen upon the related actors starts briefing/asking.
Precondition	The player must take his seat and actors must start briefing/asking.
Trigger	Adds to screen after player takes his seat.
Flow of Events	The Exam Paper HUD's adds into the screen and The player may start answering the questions.
Post Condition	Upon clicking the submit button the question paper will be hidden.

Use Case No.	5
Use Case Name	Interacting With UI
Actor	Player
Description	The player needs to interact with the UI in order to answer the questions.
Precondition	The UI must be added to the screen based on triggering events.
Trigger	The UI's will pop up when player collides with triggering events.
Flow of Events	Using mouse the player will be able to interact with the UI and clicking on check boxes.
Post Condition	The UI's will be hidden after interaction with them is done.

Use Case No.	6
Use Case Name	Asking Questions
Actor	Doctor, Army, Civil_1, Civil_2
Description	Asks the player some questions related to triggering events
Precondition	Player must interact with them by overlapping with their notifying collisions.
Trigger	Starts as soon as player touches their notifying collision
Flow of Events	1. Player stands or sits in front of them based on triggering events. 2. That time the secondary actors asks questions along with playing asking animations.
Post Condition	They will return to their normal idle state after asking questions.

Use Case No.	7
Use Case Name	Answering Questions
Actor	Player
Description	Player must answer questions given in the UI based on triggering events
Precondition	The UI that asks questions should pop up.
Trigger	The UI's will pop up when they overlap with the respective (actors) notifying collisions
Flow of Events	The player will answer the questions in the UI by clicking the check boxes using mouse interaction.
Post Condition	Must return to previous state after answering the questions.

Use Case No.	8
Use Case Name	Seating on Chair
Actor	Player
Description	Player takes seat during exam and viva board in Phase 1.
Precondition	Player needs to touch the collision related to triggering events in order to be able to take seat.
Trigger	On pressing E, the player takes seat.
Flow of Events	The player touches the collision that enables seating Boolean and when the player presses E then player plays seating animation and takes a seat.
Post Condition	Player must return to idle state (standing) from seating

Use Case No.	9
Use Case Name	Calculating Marks
Actor	Question Paper HUD
Description	Calculates the total marks of the answered questions.
Precondition	Must answer the questions and click submit.
Trigger	Starts calculating after submit button has been clicked
Flow of Events	When the player answers the questions and clicks the submit button then it takes those check box answers and calculates them.
Post Condition	Must store the data's into database.

Use Case No.	10
Use Case Name	Keeping Track of User Progress
Actor	Saving Actor
Description	Saves all the necessary data's into file system.
Precondition	Must touch the collision boxes in order to begin the process.
Trigger	Starts storing as soon as player collision overlaps with the save systems collisions
Flow of Events	When the players collision overlaps with the save systems collisions the save system then starts taking the necessary values from the required classes and stores them into the file system.
Post Condition	After saving the data's the actor class will destroy itself otherwise if the player collides with it again then the saved data's will get overwritten.

Use Case No.	11
Use Case Name	Talking with Parents
Actor	Player
Description	Player talks with his parents
Precondition	Player must touch the parent's collision responses.
Trigger	Starts playing cinematic when the players collision overlaps with the parents notifying collision.
Flow of Events	When the player comes near his parents a cinematic scene takes place that he is actually talking with his parents. Mom plays talking animation during that moment.
Post Condition	Must return to players idle state after conversation.

Use Case No.	12
Use Case Name	Talking with House master
Actor	Player
Description	Player talks with the house master
Precondition	Player must touch the house master's collision responses.
Trigger	Starts playing cinematic when the players collision overlaps with the house masters notifying collision.
Flow of Events	When the player comes near the house master a cinematic scene takes place that he is actually talking with the house master. Here is just a normal talking and no questions UI from the house master.
Post Condition	Must return to players idle state after conversation.

3.3 Activity Diagram

To describe the use cases more specifically there are some activity diagrams to elucidate the system more distinctively.

3.3.1 Asking for Admit Card

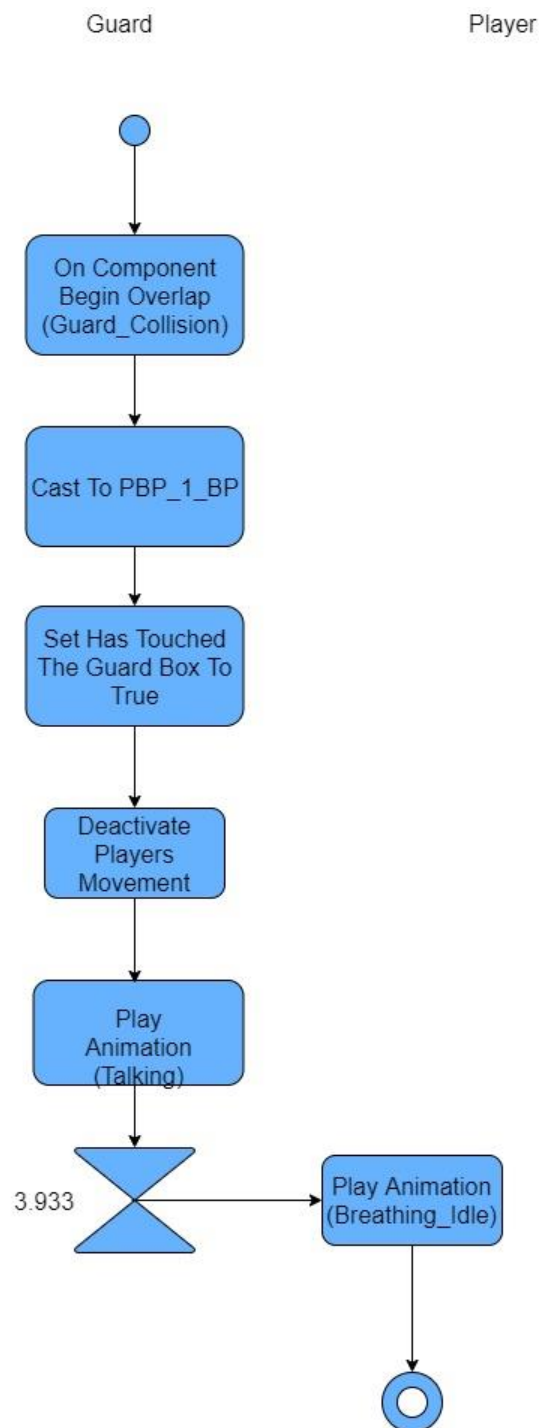


Fig: Asking for admit card

3.3.2 Showing Admit Card

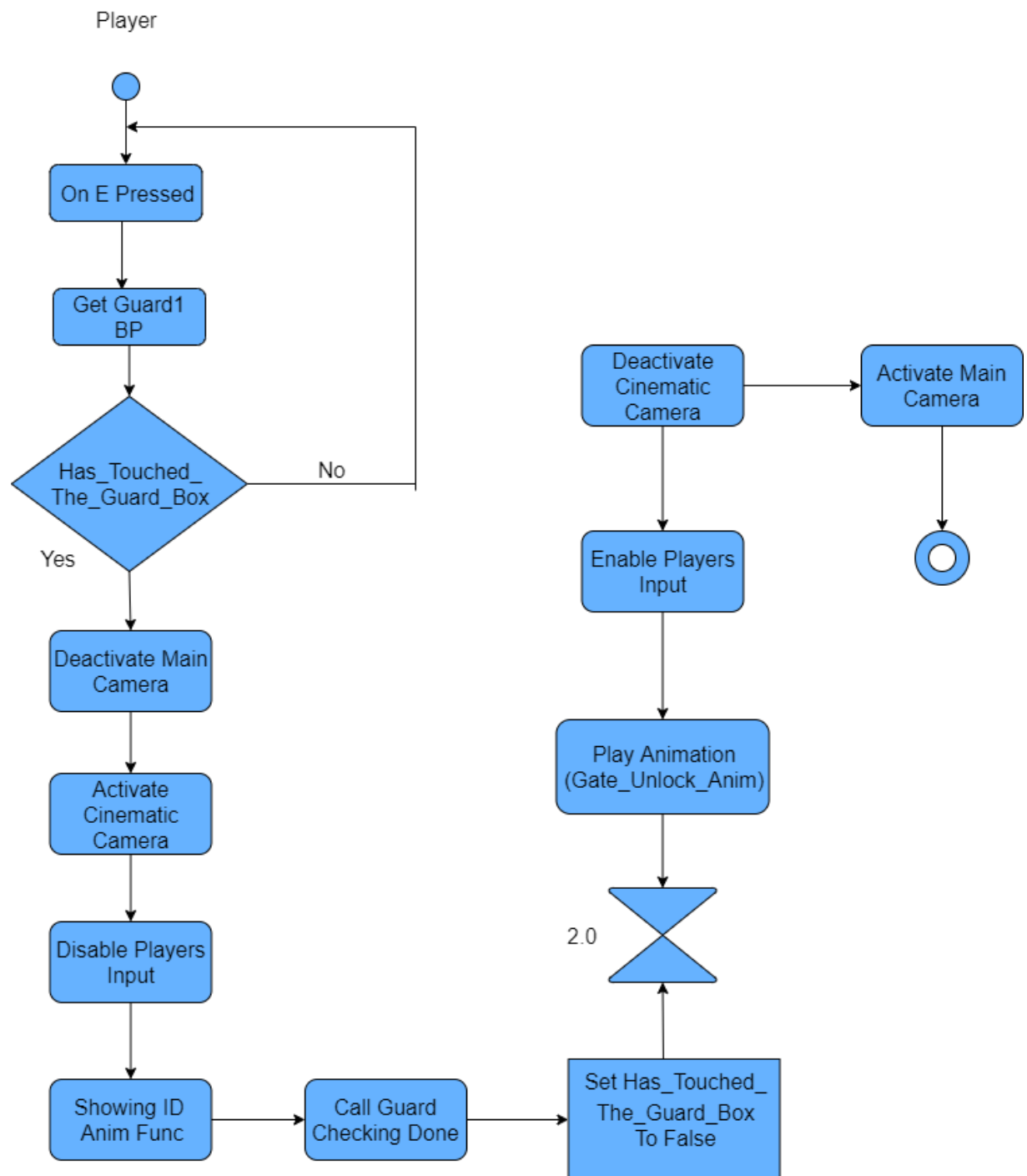


Fig: Showing admit card

3.3.3 Briefing About Exam

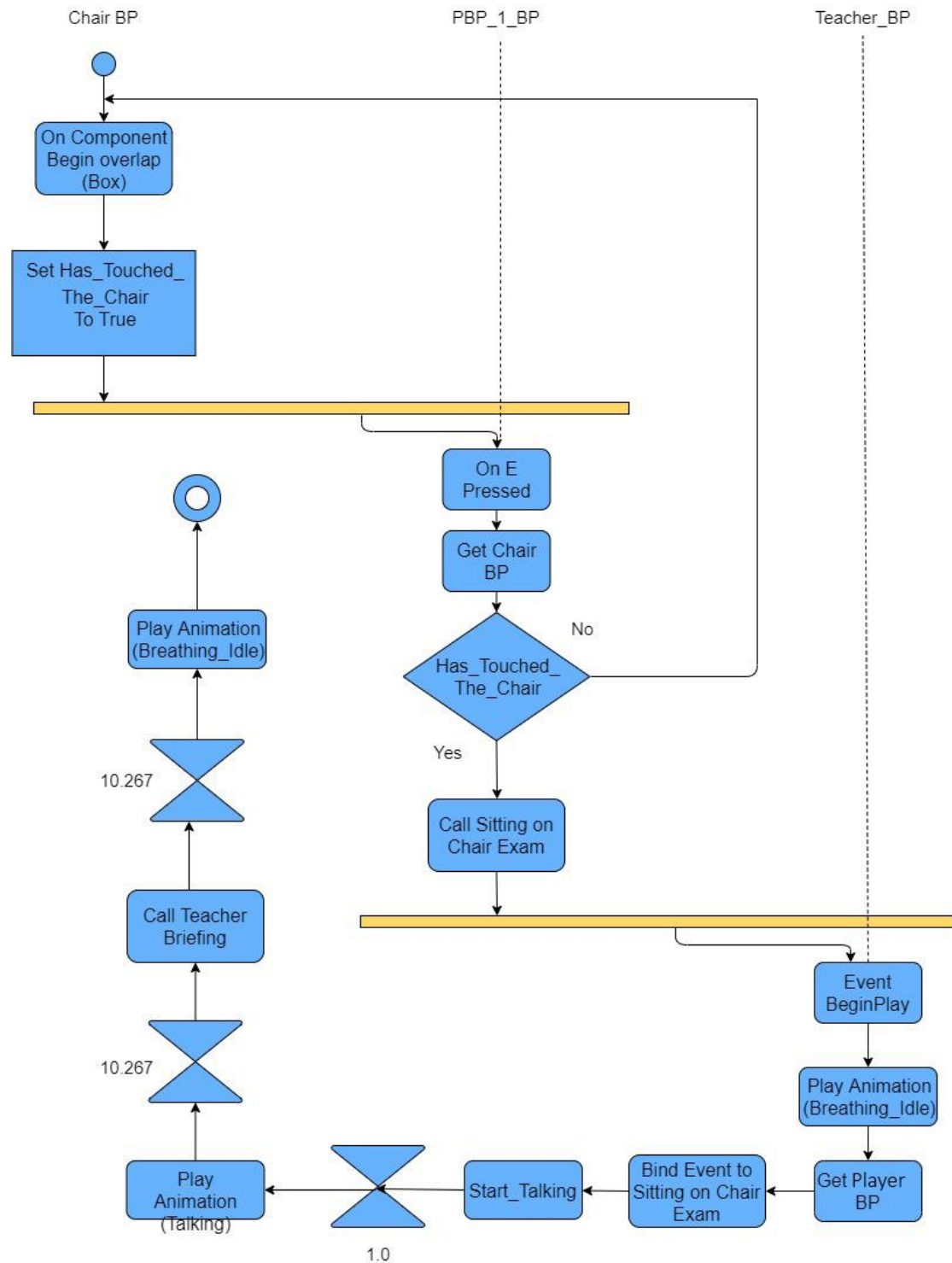


Fig: Briefing about exam

3.3.4 Providing Exam Paper

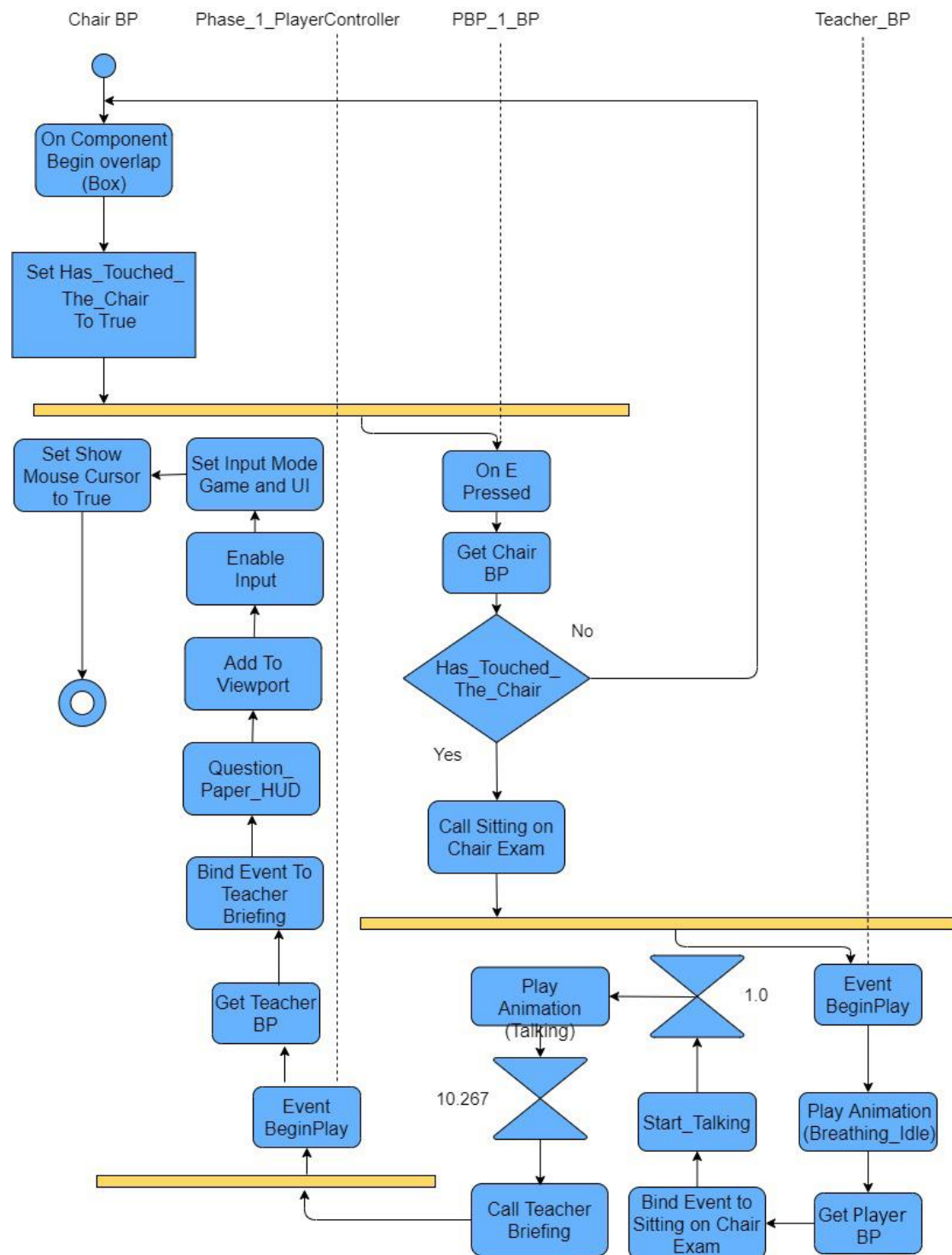


Fig: Providing exam paper

3.3.5 Interacting With UI

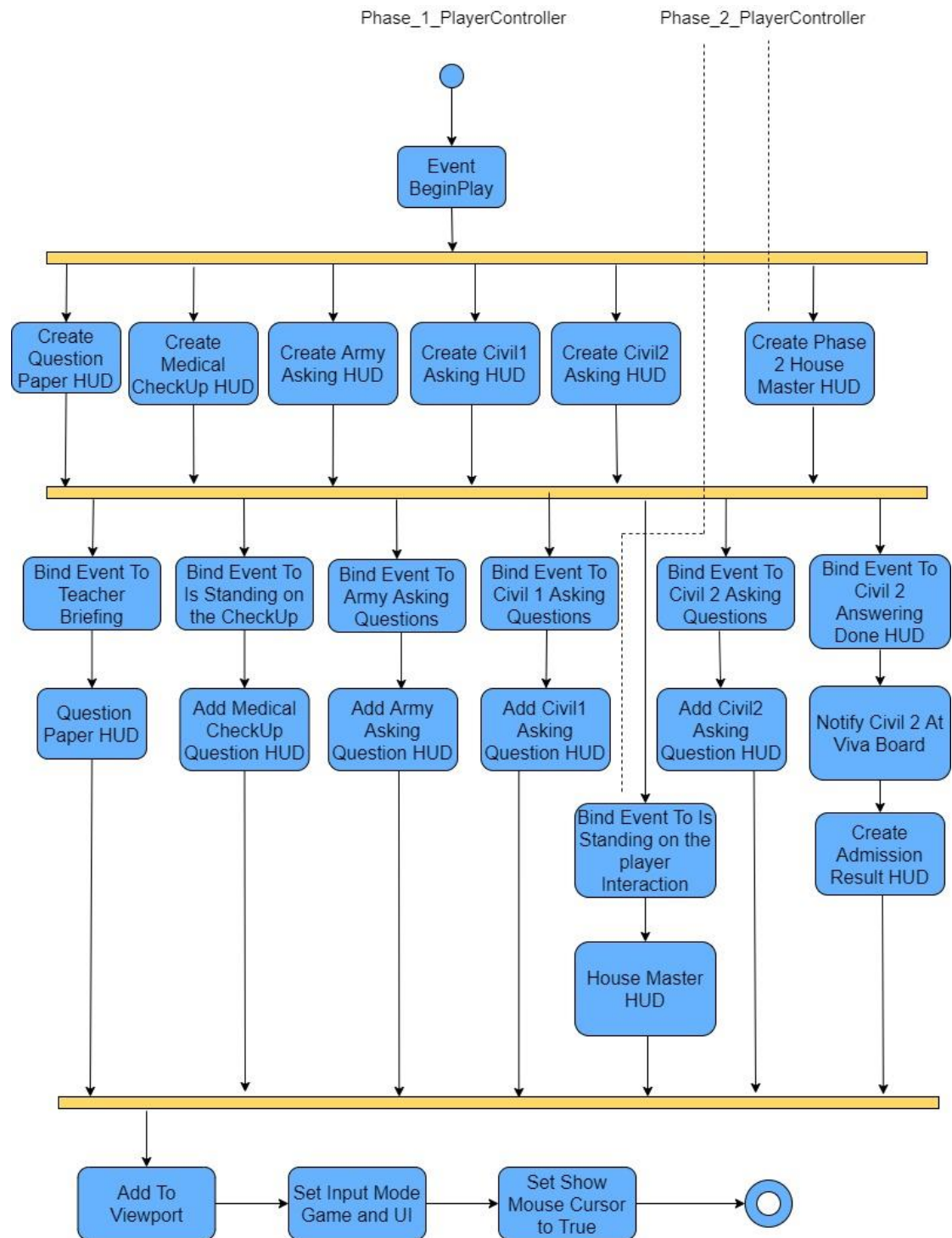


Fig: Interacting with UI

3.3.6 Asking Questions

3.3.6.1 Doctor

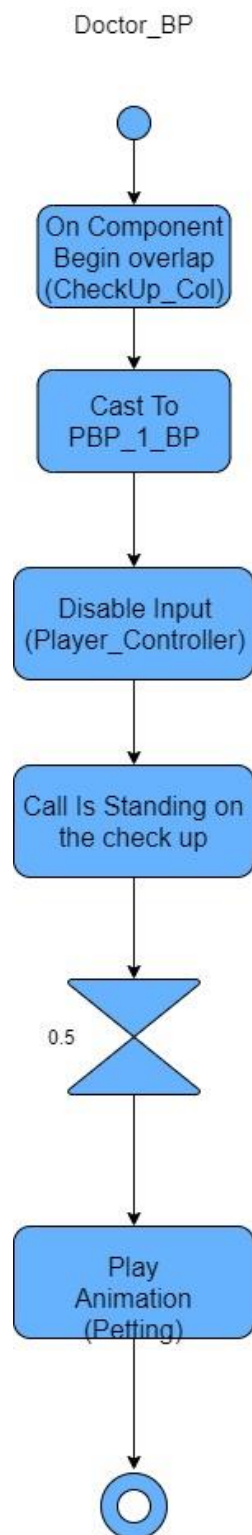


Fig: Doctor

3.3.6.2 Army

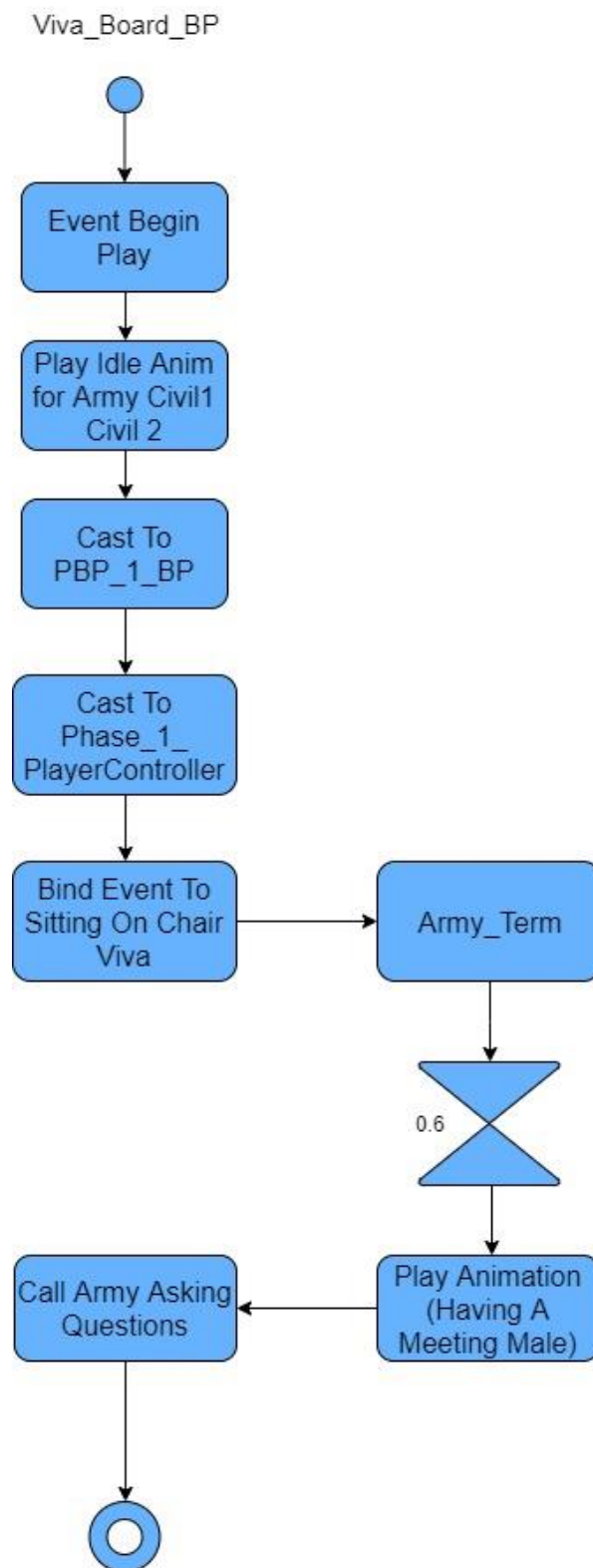


Fig: Army

3.3.6.3 Civil 1

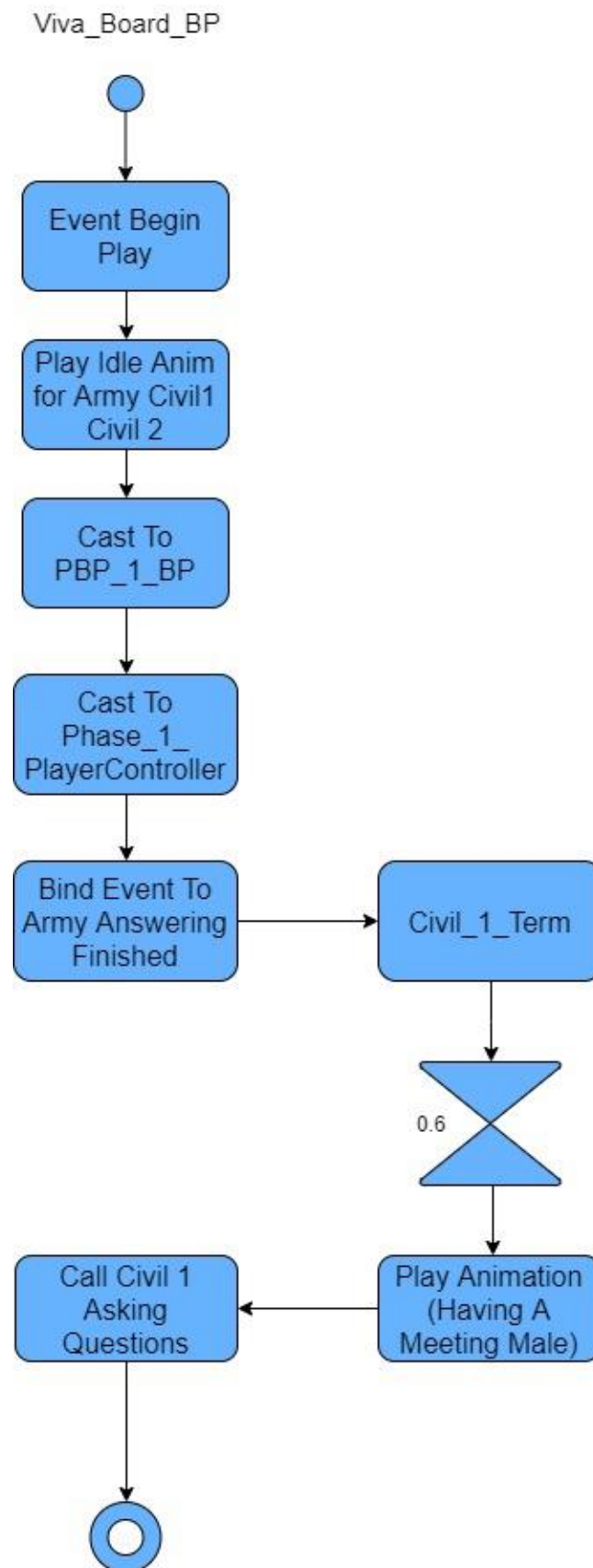


Fig: Civil1

3.3.6.4 Civil 2

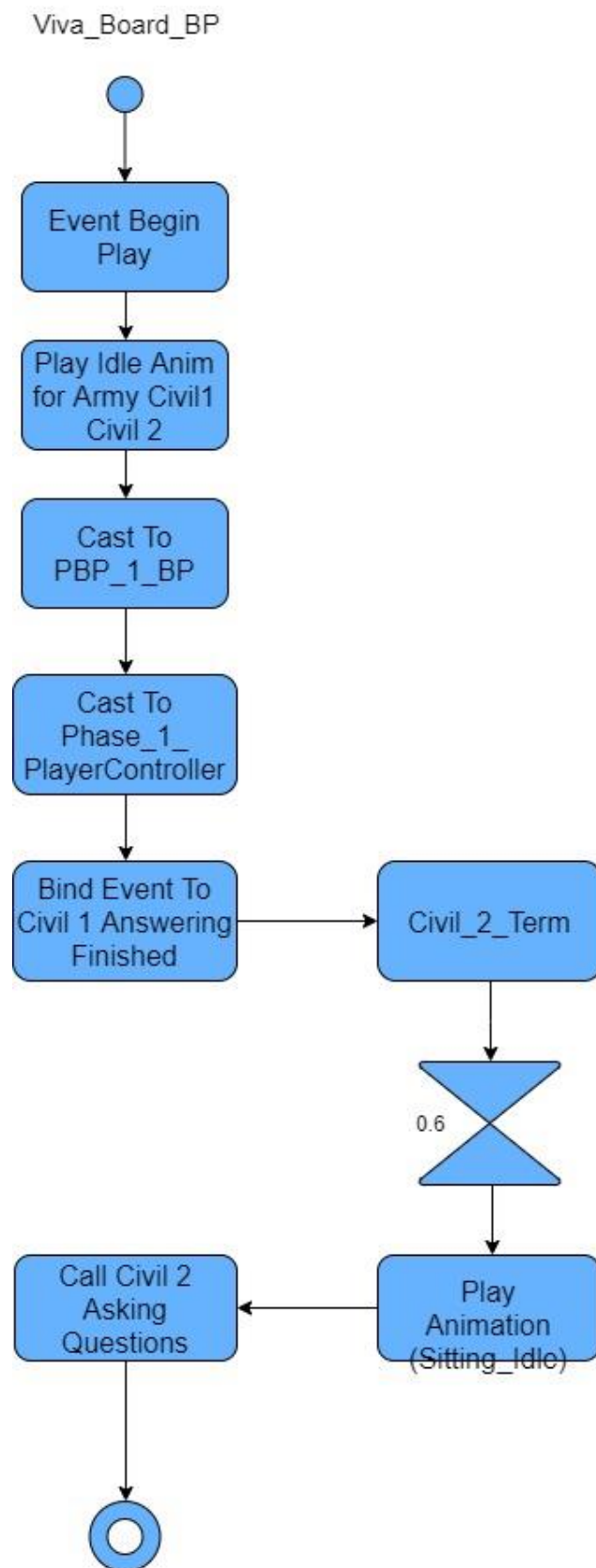


Fig: Civil2

3.3.7 Answering Questions

Player Controller (Global)

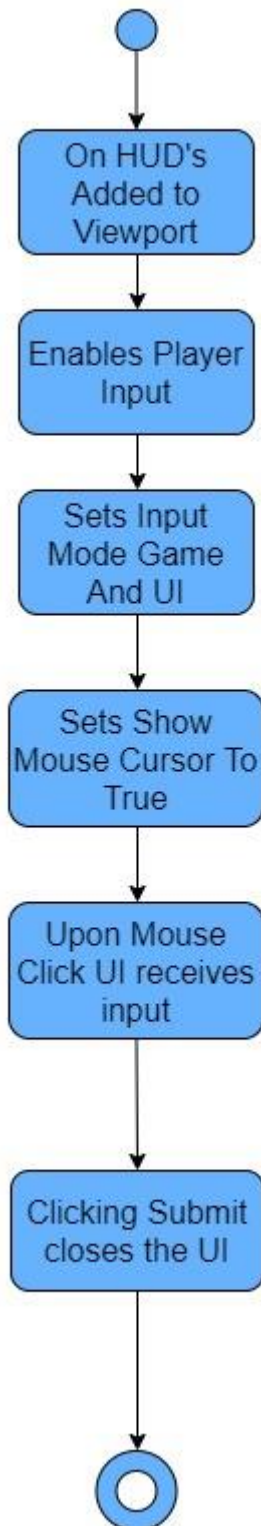


Fig: Answering Questions

3.3.8 Seating on Chair

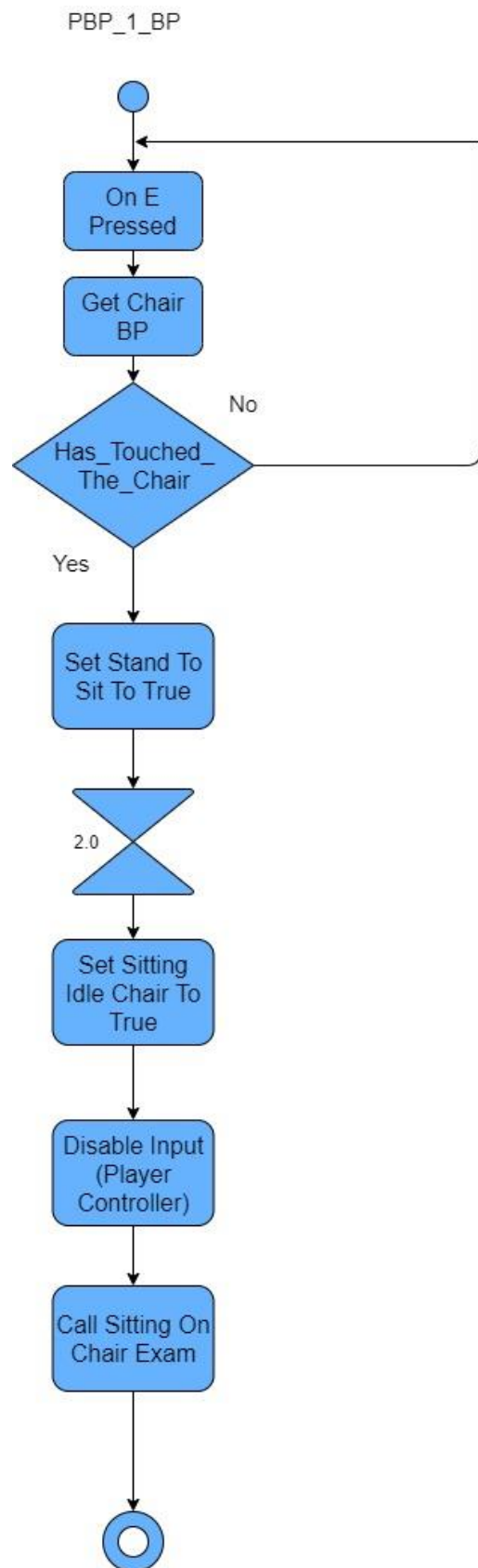


Fig: Seating on chair

3.3.9 Calculating Marks

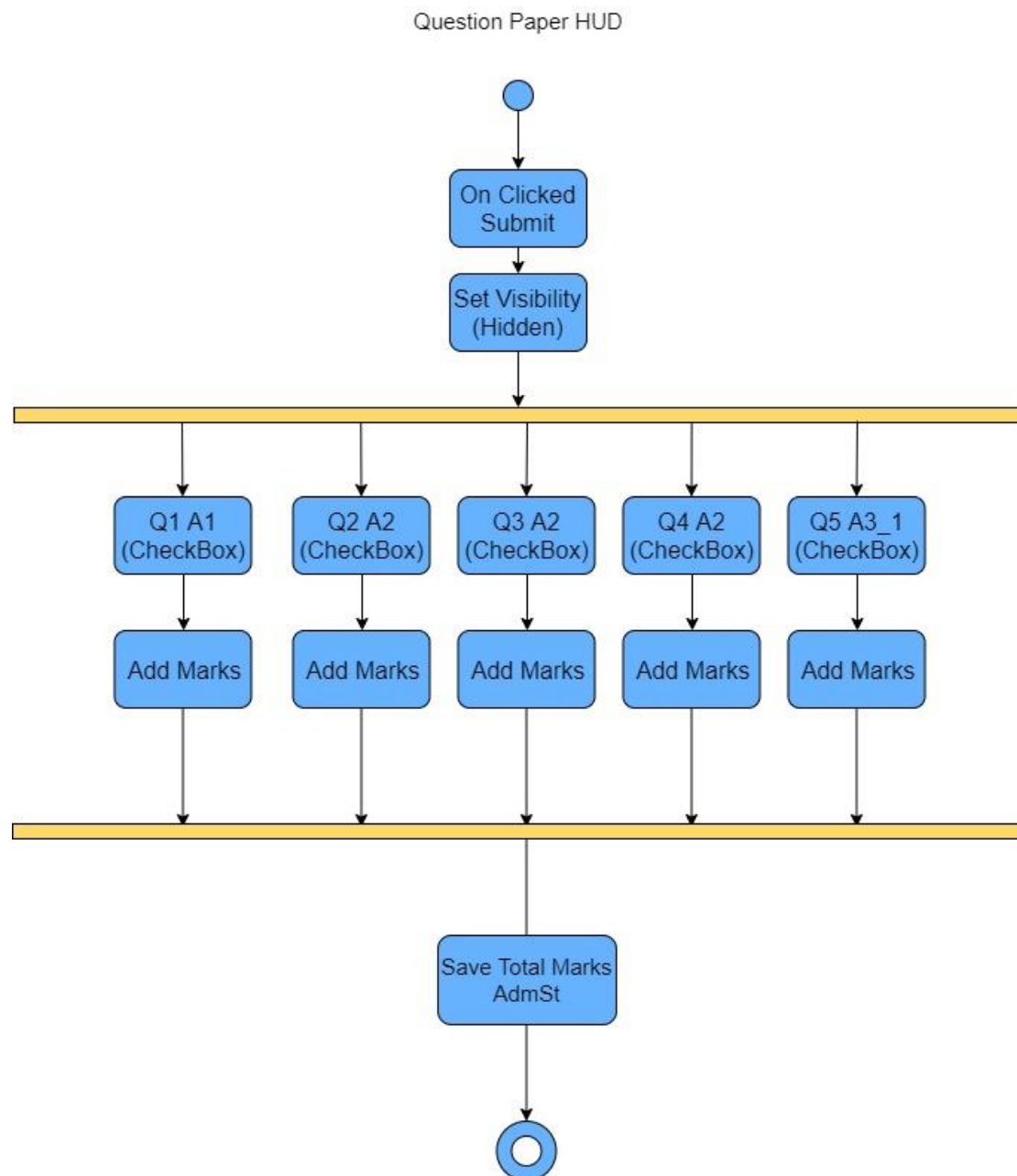
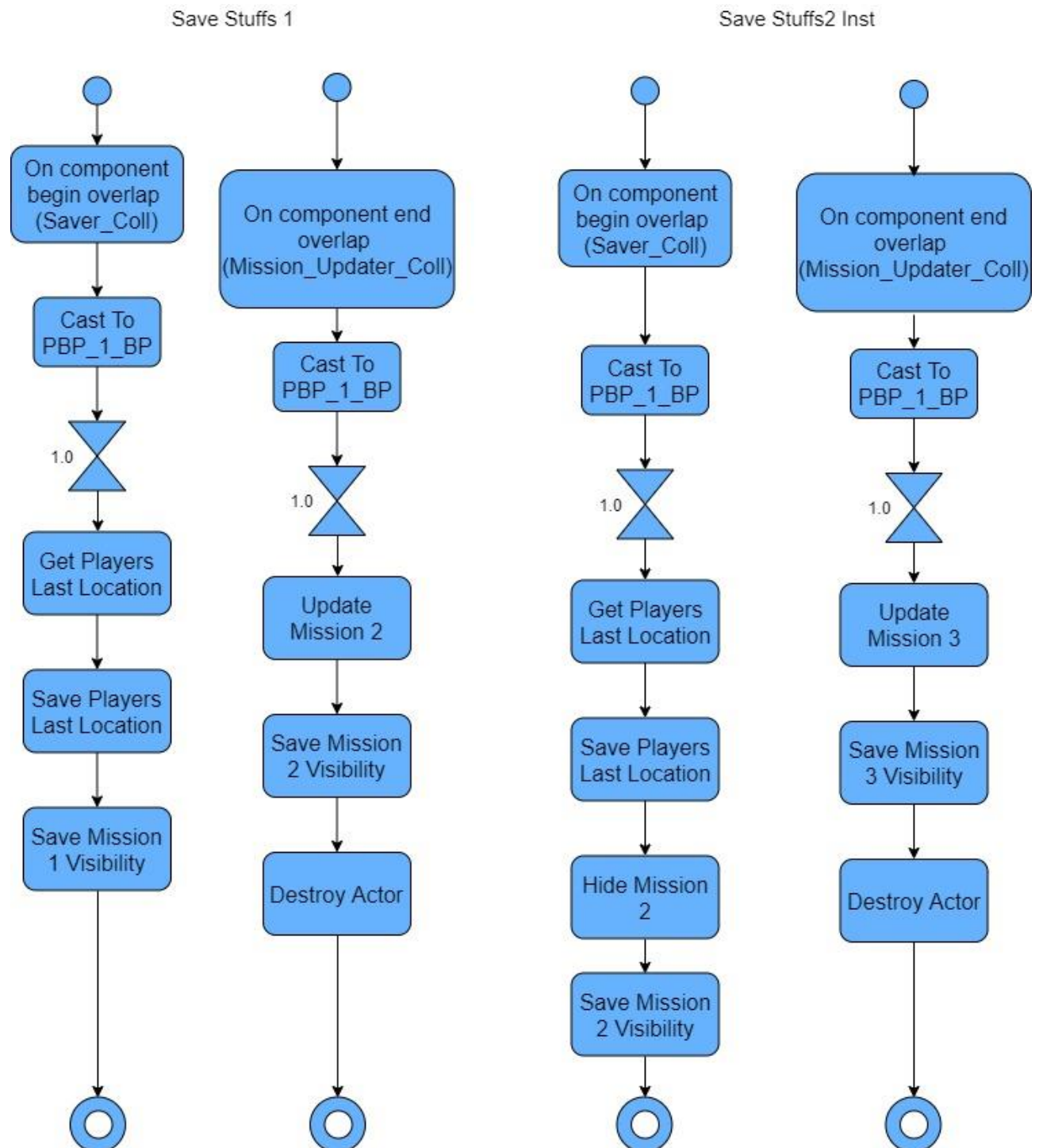


Fig: Calculating Marks

3.3.10 Keeping Track of User Progress



Save Stuffs3 Inst

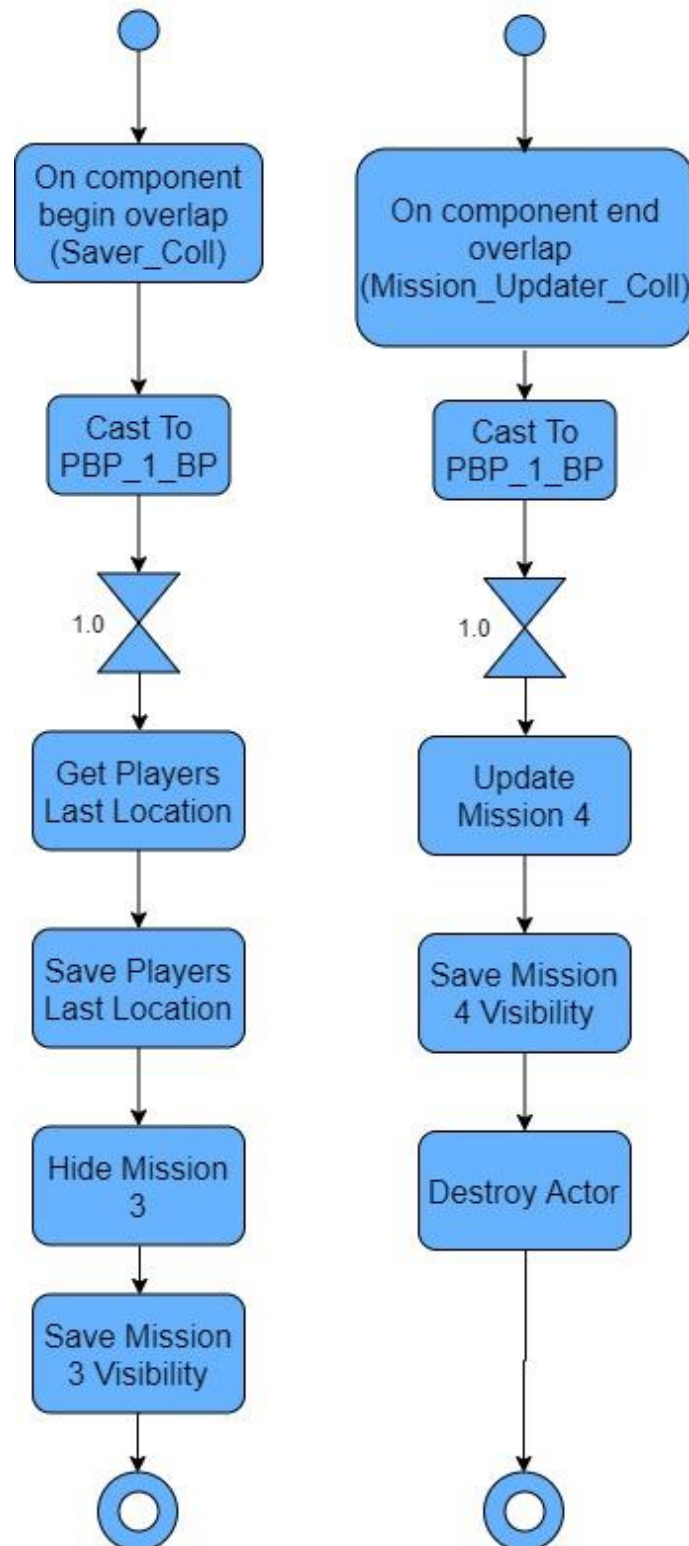


Fig: Keeping track of user progress

3.3.11 Talking with Parents

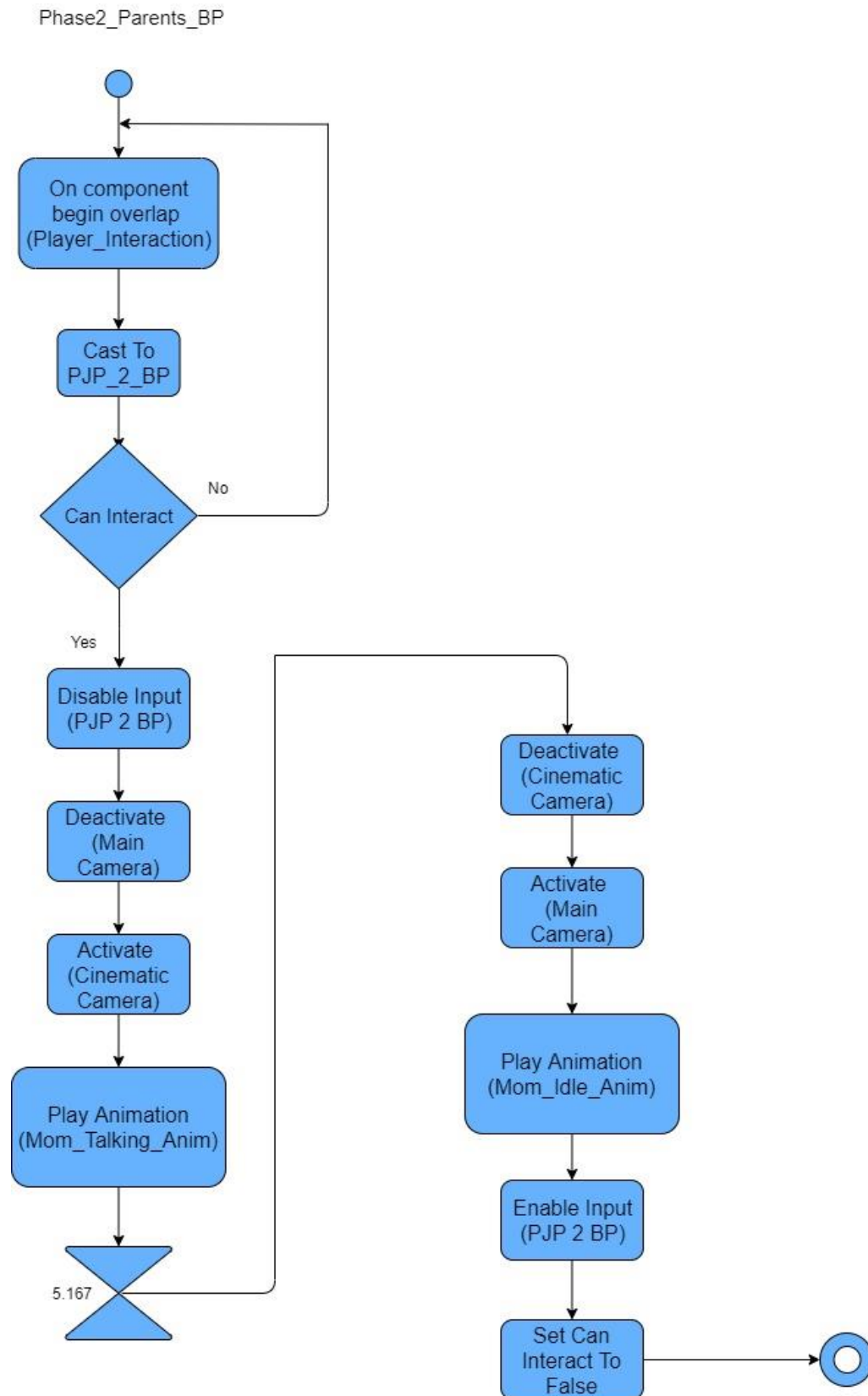


Fig: Talking with parents

3.3.12 Talking with House Master

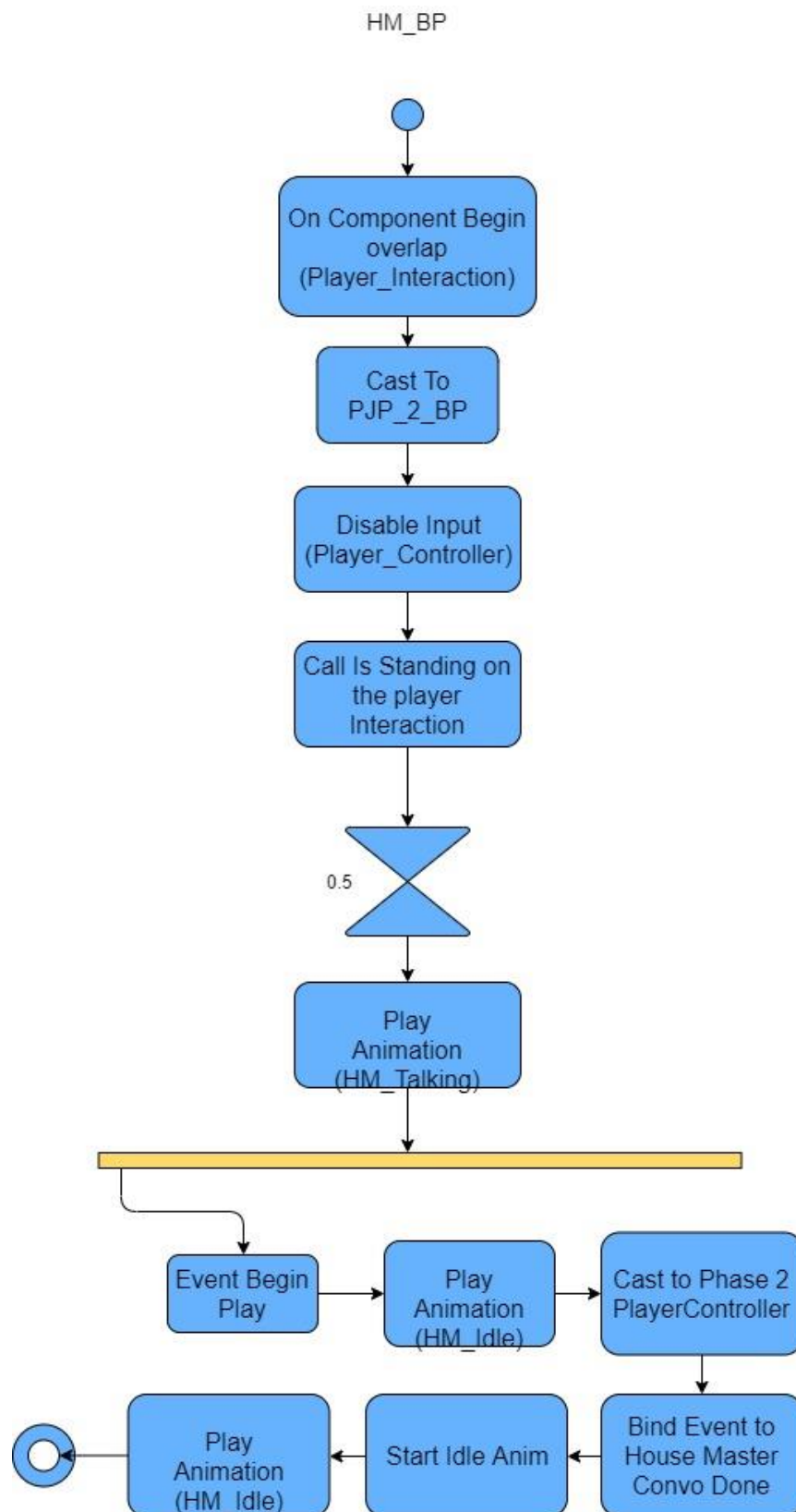


Fig: Talking with house master

3.4 System Sequence Diagram

3.4.1 Asking for Admit Card

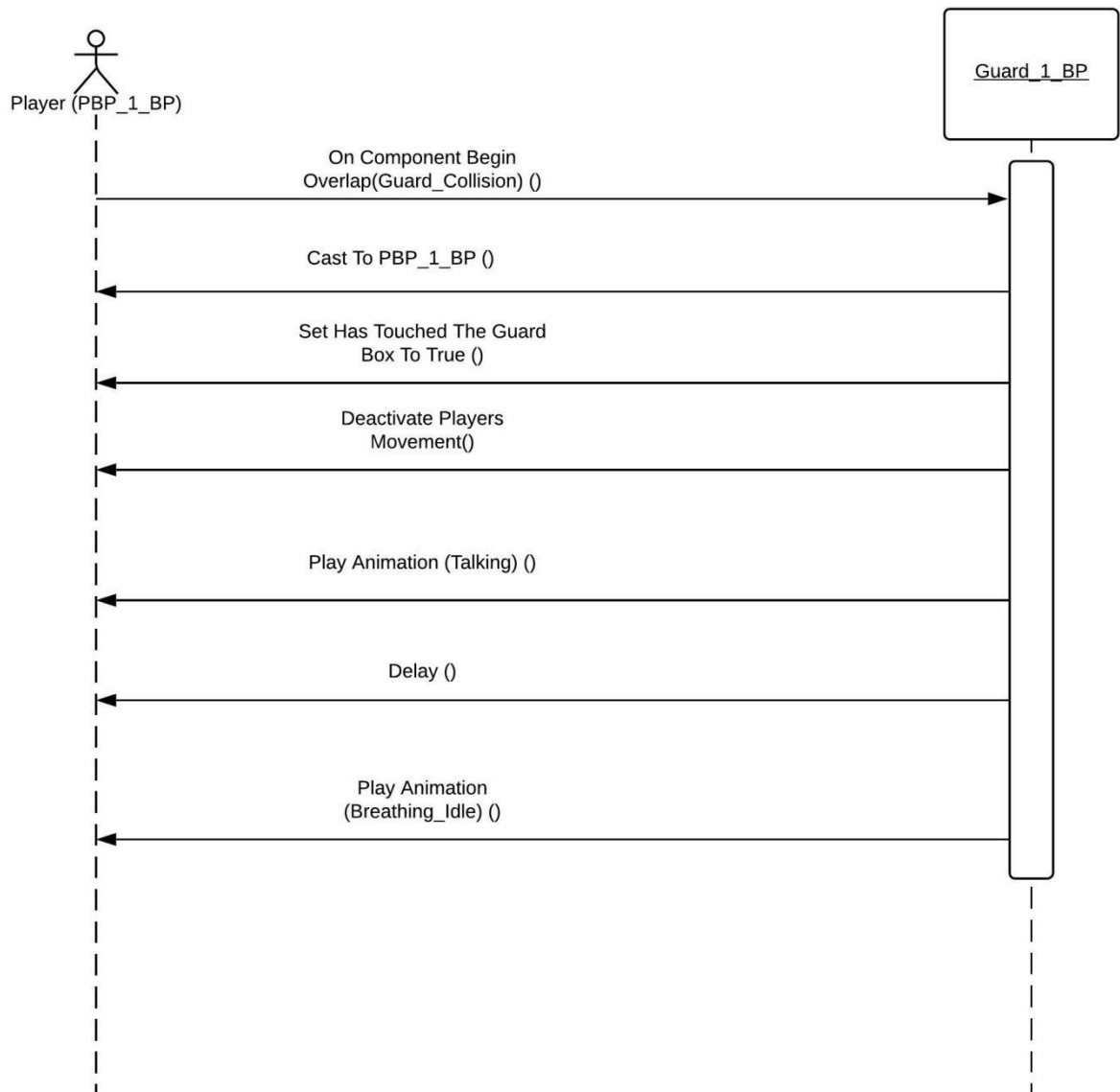


Fig: Asking for admit card

3.4.2 Showing Admit Card

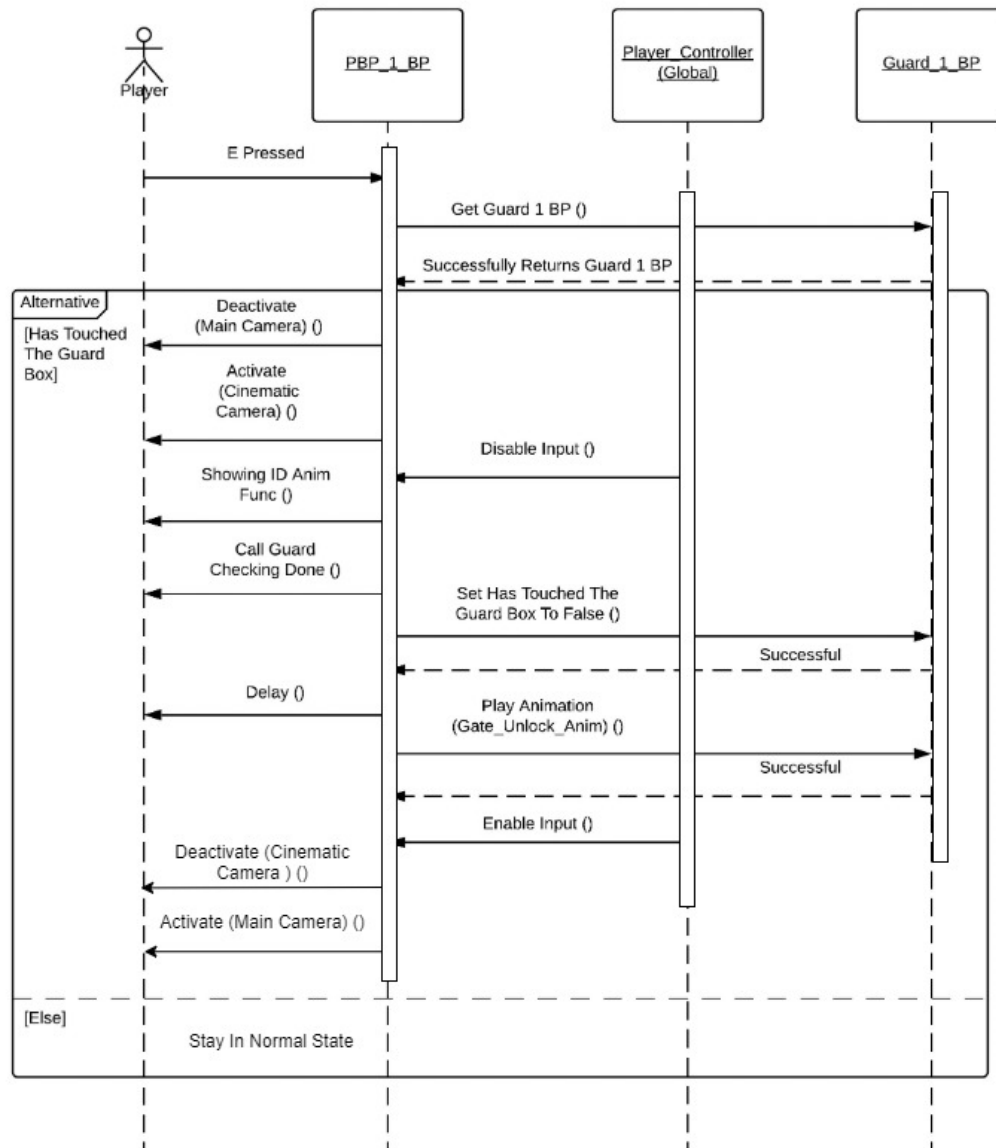


Fig: Showing admit card

3.4.3 Briefing About Exam

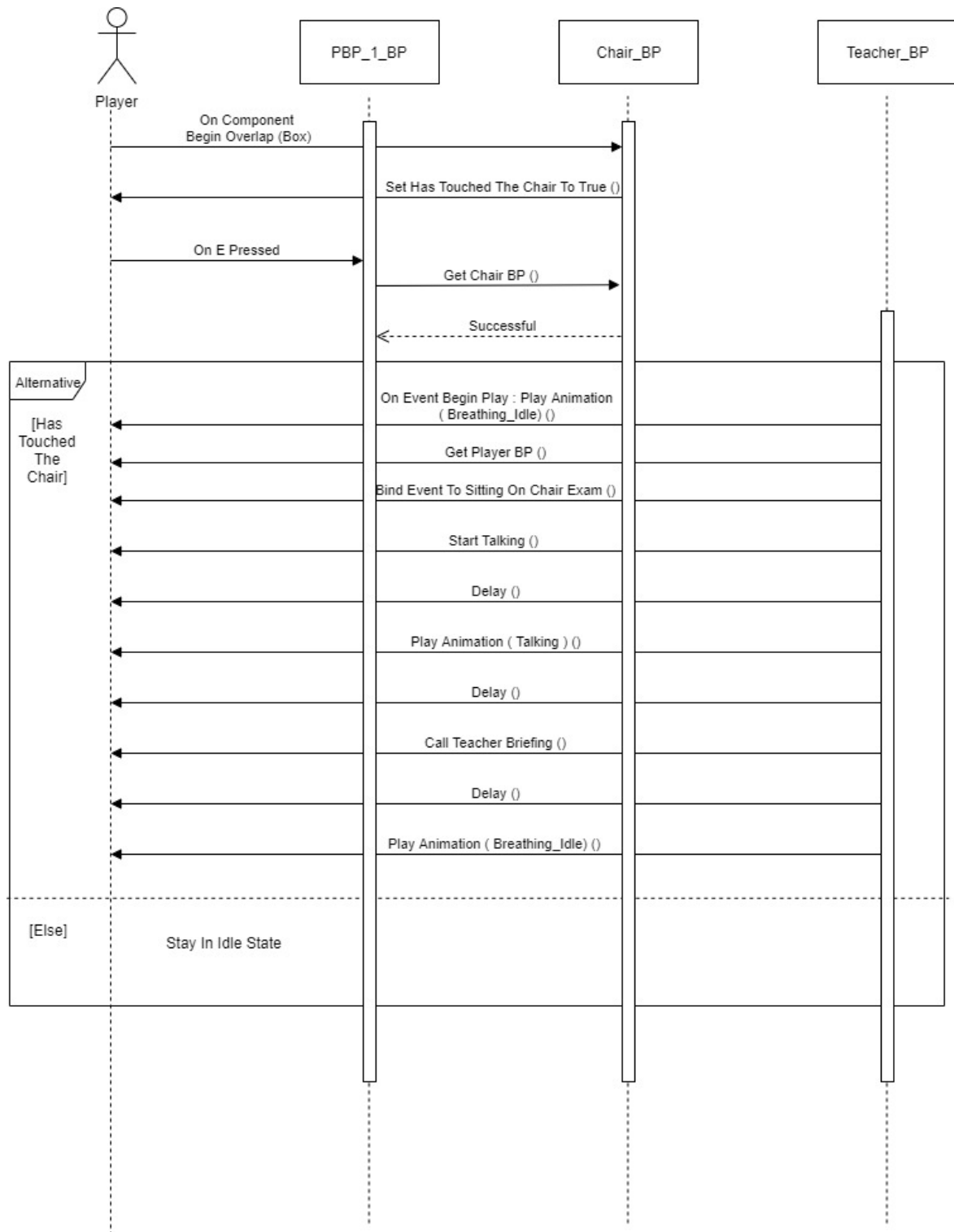


Fig: Briefing about exam

3.4.4 Providing Exam Paper

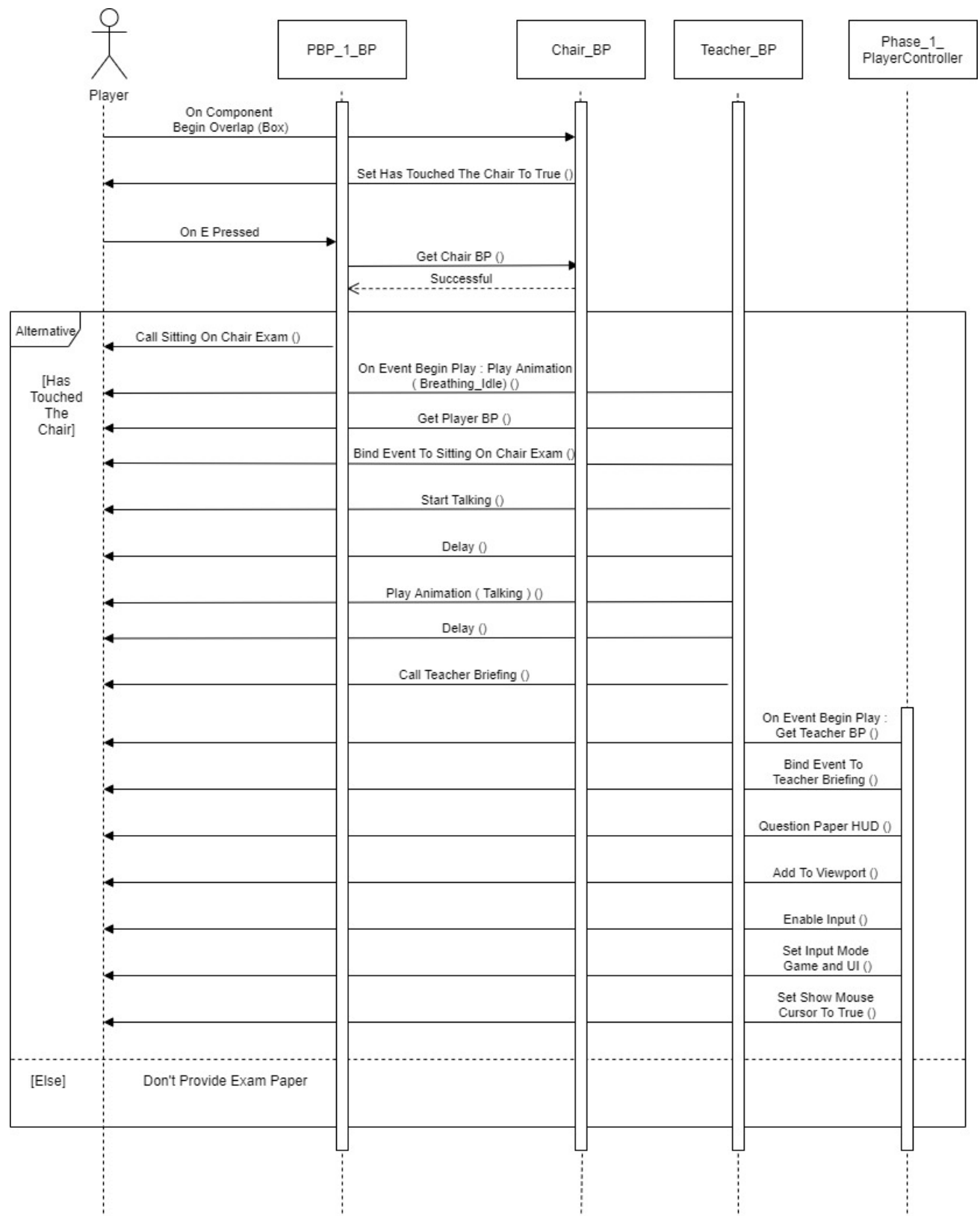


Fig: Providing exam paper

3.4.5 Interacting With UI



Fig: Interacting with UI

3.4.6 Asking Questions

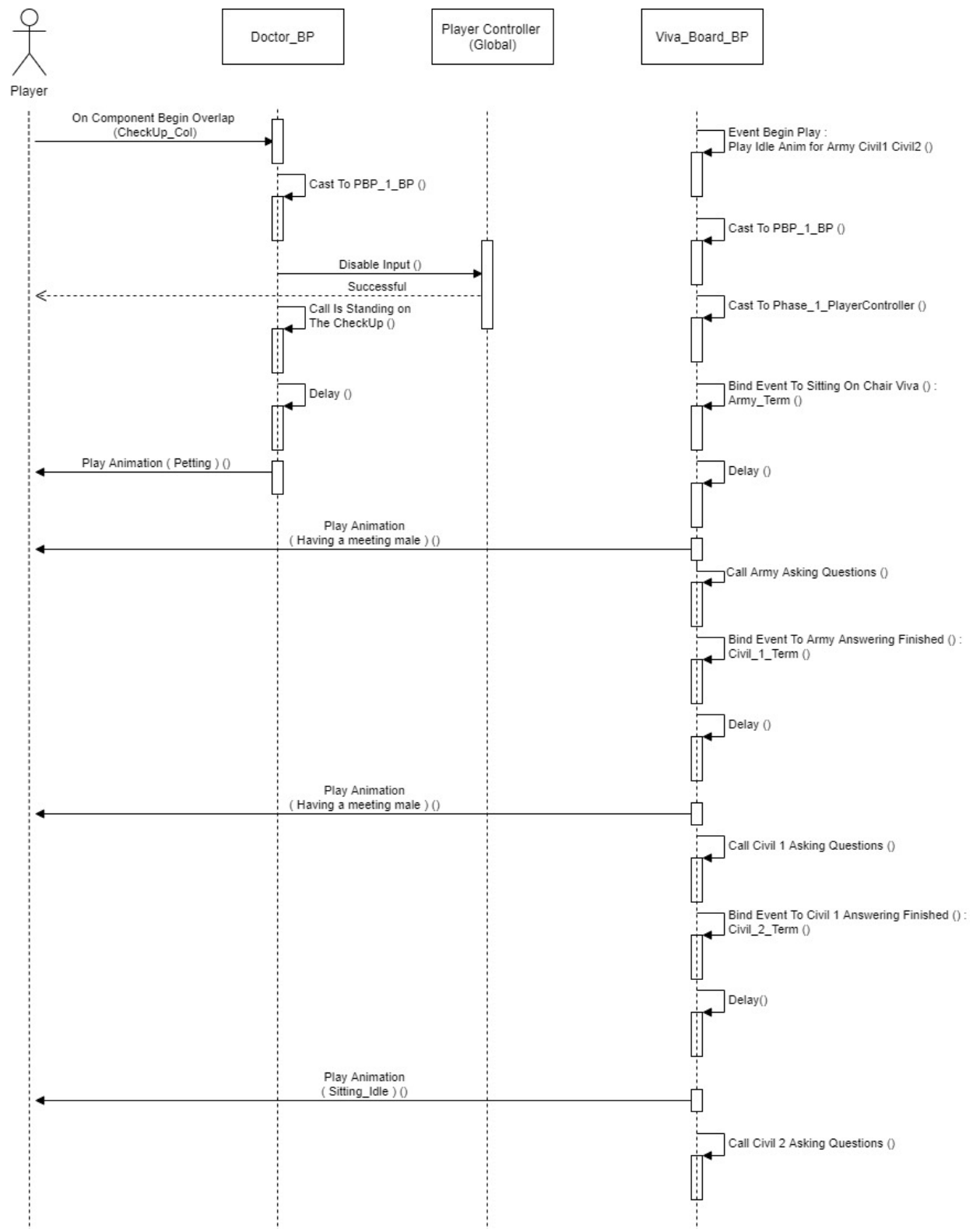


Fig: Asking questions

3.4.7 Answering Questions

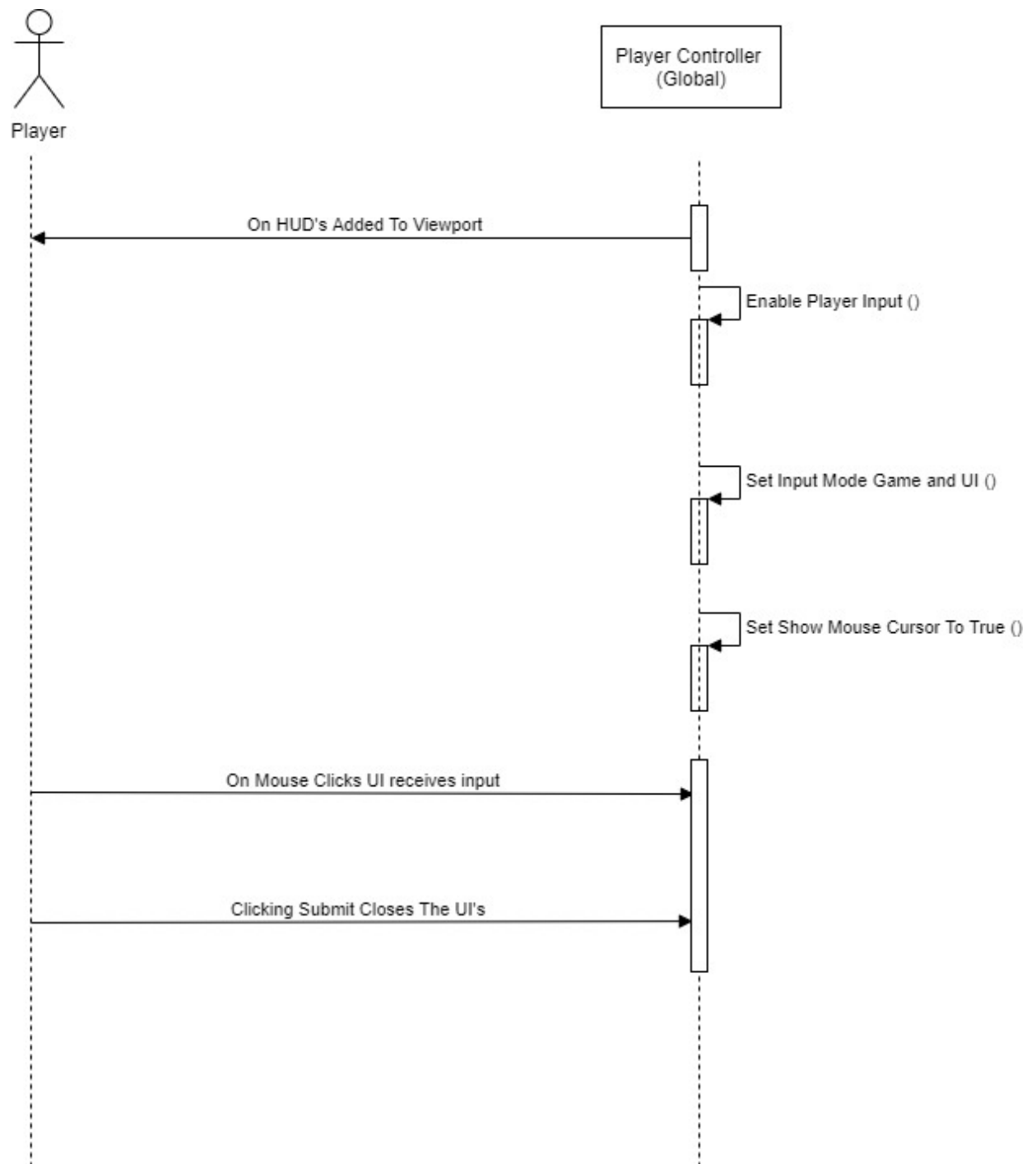


Fig: Answering questions

3.4.8 Seating on Chair

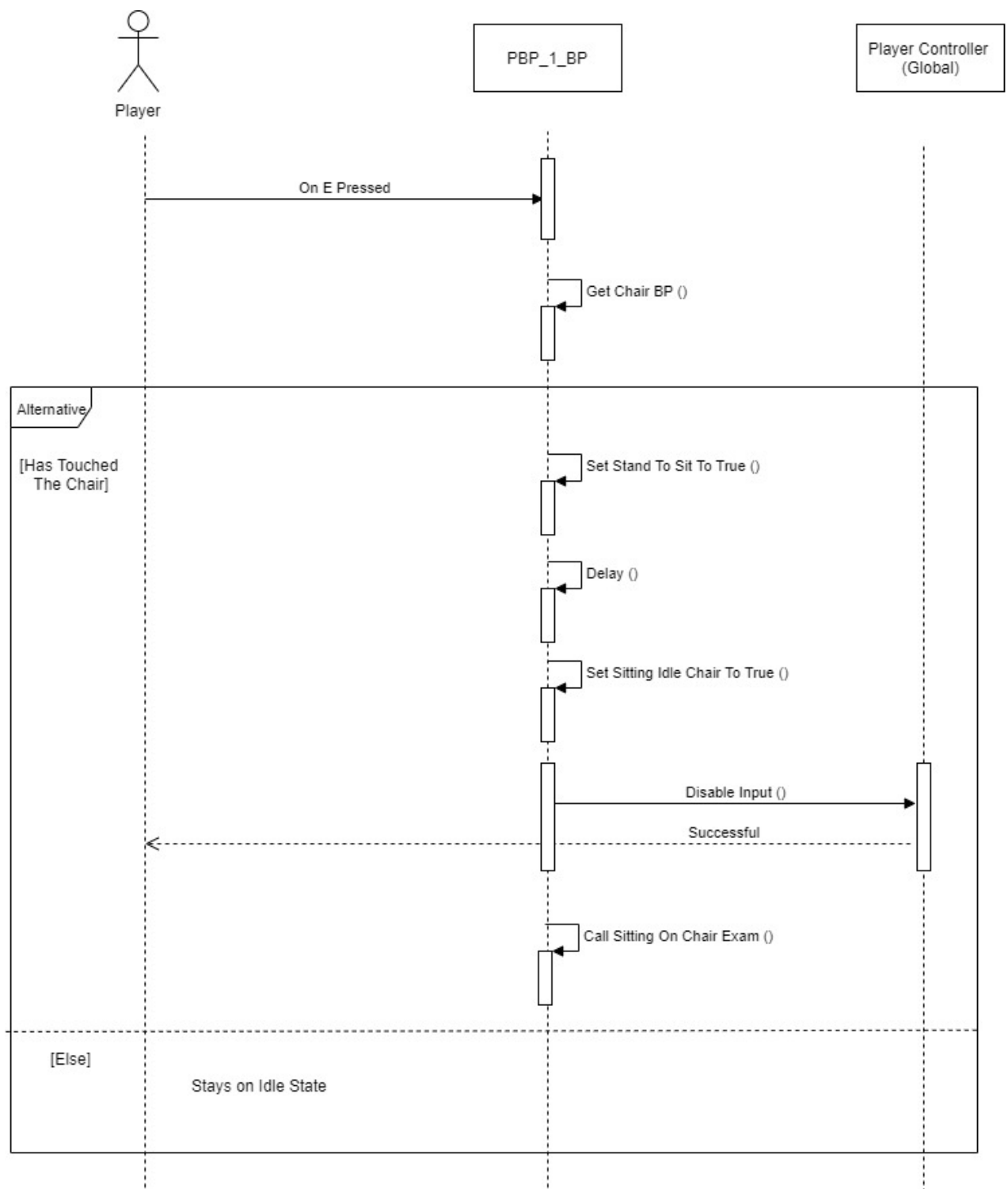


Fig: Seating on chair

3.4.9 Calculating Marks

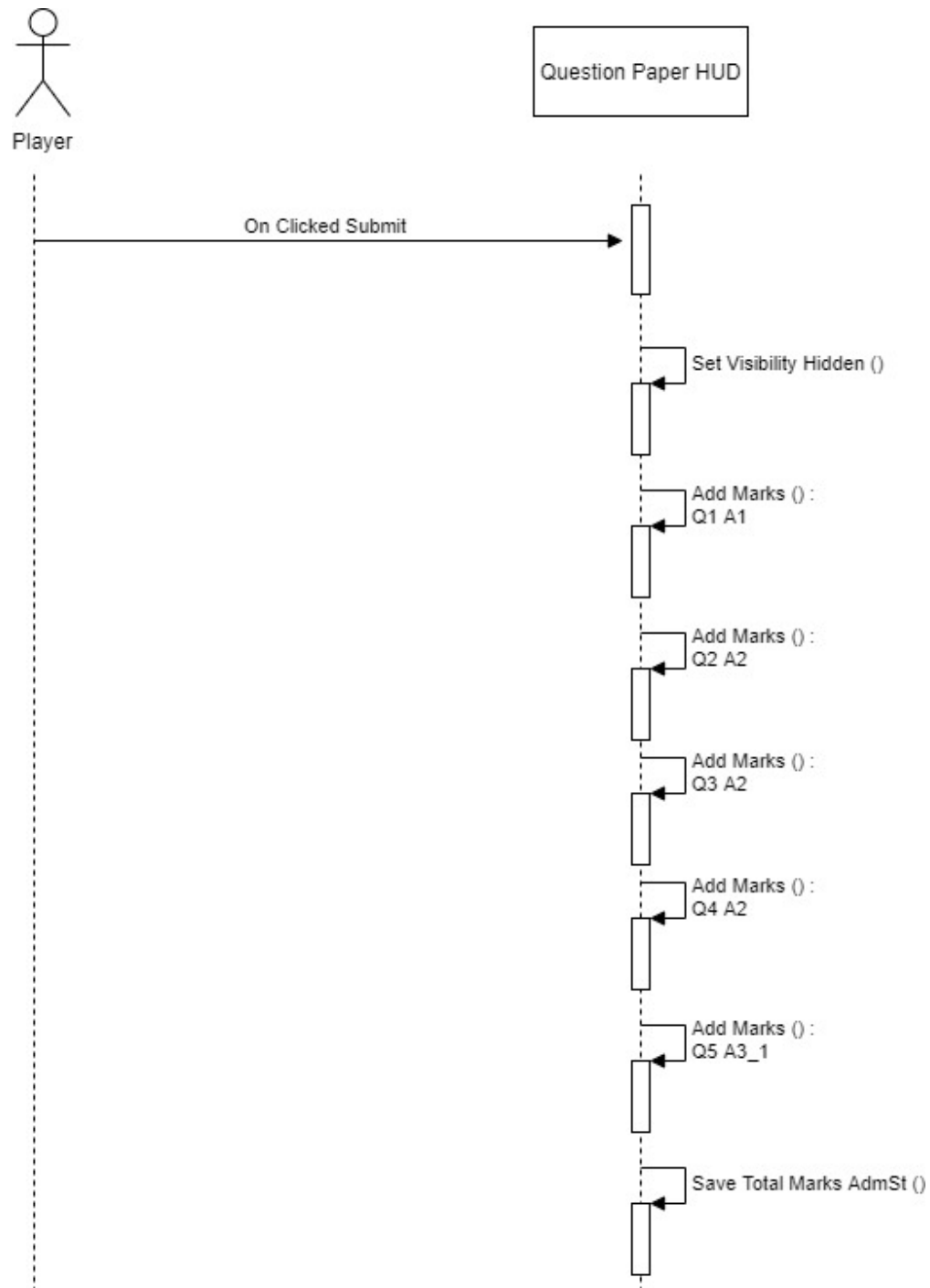


Fig: Calculating marks

3.4.10 Keeping Track of User Progress

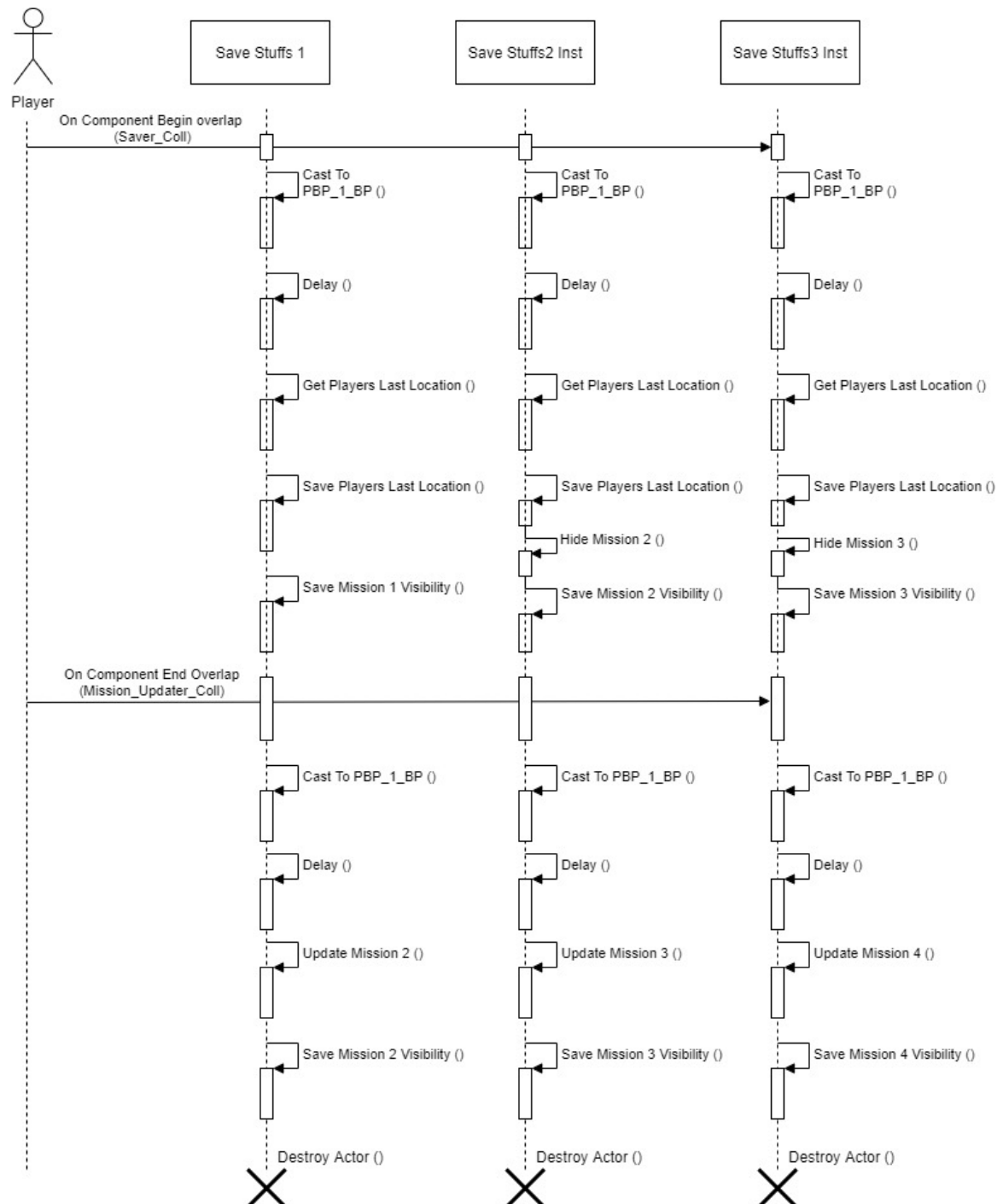


Fig: Keeping track of user progress

3.4.11 Talking with Parents

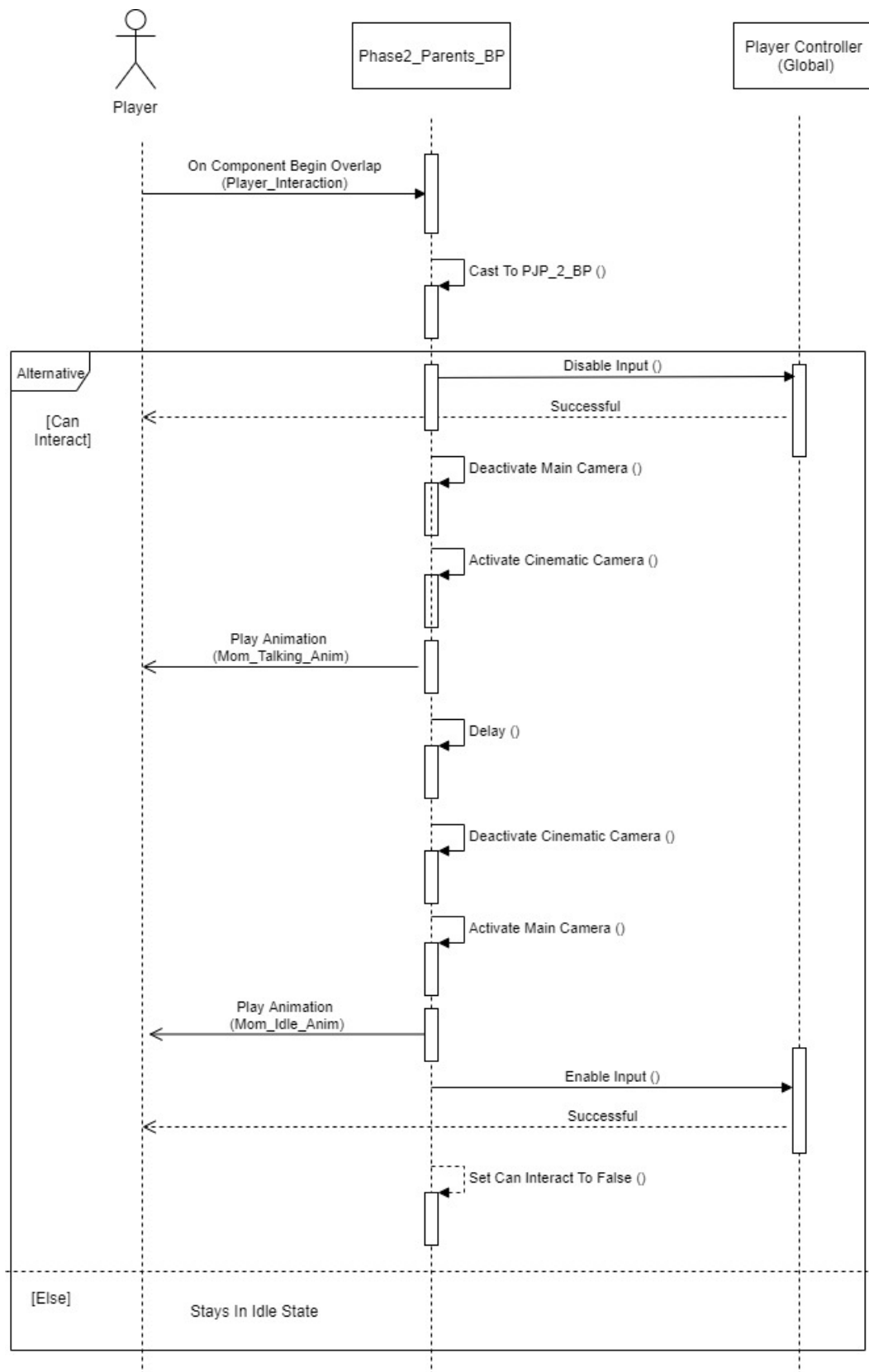


Fig: Talking with parents

3.4.12 Talking with House Master

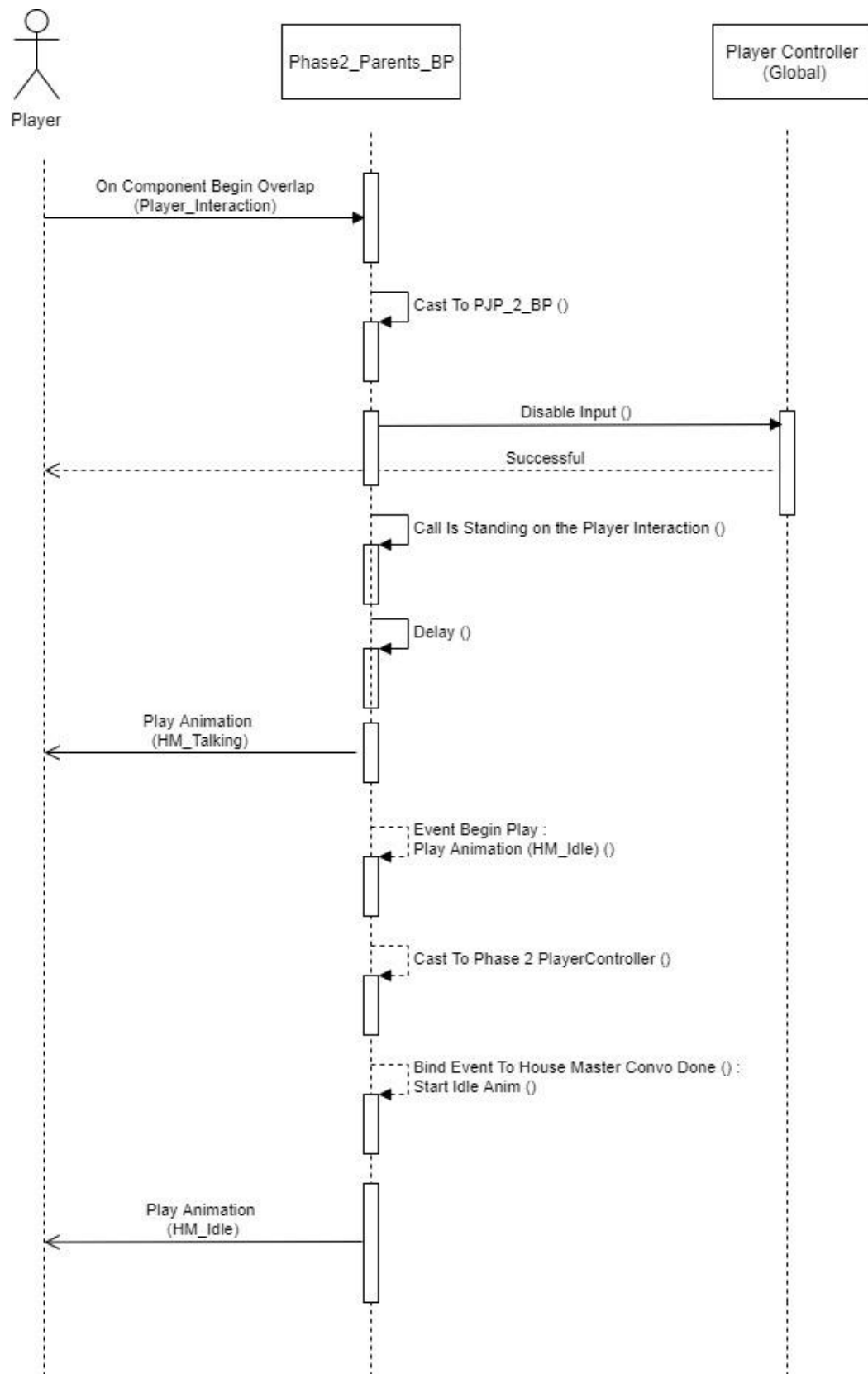


Fig: Talking with house master

4. System Design Specification

4.1 Class Diagram of Classes



Teacher_BP
+ Mesh: Skeletal Mesh
+ Teacher_Briefing() : MultiCast Delegate + Start_Talking () : Void + Get_Player_BP () : PBP 1 BP

Doctor_BP
+ CheckUp_Col: Box Collision + Doc_Notify_Col : Box Collision + Mesh: Skeletal Mesh
+ Is_Standing_On_The_CheckUp() : MultiCast Delegate + Start_Idle_Anim () : Void

Viva_Board_BP
+ Army : Skeletal Mesh + Civil_1 : Skeletal Mesh + Civil_2 : Skeletal Mesh + Viva_Notify : Box Collision + Has_Sit_For_Viva : Boolean + Phase_1_PlayerController : Phase 1 PlayerController
+ Army_Asking_Questions() : Multicast Delegate + Civil_1_Asking_Questions() : Multicast Delegate + Civil_2_Asking_Questions() : Multicast Delegate + Play_Idle_Anim_For_Army_Civil1_Civil2() : Void + Army_Term() : Void + Civil_1_Term() : Void + Civil_2_Term() : Void + Reset_Has_Sit_For_Viva() : void

Phase_1_PlayerController	
+ My_Anim_BP: PBP 1 Anim + Question_Paper_HUD: Question Paper HUD + Medical_CheckUp_HUD: Medical CheckUp HUD + My_Player_BP: PBP 1 BP + Doctor_BP: Doctor BP + Army_Asking_HUD: Army Asking HUD + Viva_Board_BP: Viva Board BP + Civil_1_Asking_HUD: Civil 1 Asking HUD + Civil_2_Asking_HUD: Civil 2 Asking HUD + Teacher_BP: Teacher BP + Admission_Result_HUD: Admission Result HUD + Phase_1_Mission_Update_HUD: Phase 1 Mission Update HUD + MySaveGame: My Save Game	
+ Teleport_Player_To_Last_Location(): void + Get_Save_Game_System(): Void + Exam_Finished(): Multicast Delegate + CheckUp_Finished(): Multicast Delegate + Army_Answering_Finished(): Multicast Delegate + Civil_1_Answering_Finished(): Multicast Delegate + Viva_Finished(): Multicast Delegate + Mission_Update_HUD_Creation(): void + Cast_To_Anim_BP(): void + Get_Doctor_BP(): void + Create_Question_Paper_HUD(): void + Create_Medical_Checkup_HUD(): void + Get_Viva_Board_BP(): void + Create_Army_Asking_HUD(): void + Create_Civil1_Asking_HUD(): void + Create_Civil2_Asking_HUD(): void + Create_Admission_Result_HUD(): void + Stand_Up_After_Viva_Finishes(): void + Get_Teacher_BP(): void + Question_Paper_HUD(): void + Get_Up_After_Exam_Finished(): void + Add_Medical_CheckUp_Question_HUD(): void + Check_Up_Finished(): void + Add_Army_Asking_Question_HUD(): void + Notify_Army_At_Viva_Board(): void + Add_Civil_1_Asking_Question_HUD(): void + Notify_Civil_1_At_Viva_Board(): void + Add_Civil_2_Asking_Question_HUD(): void + Notify_Civil_2_At_Viva_Board(): void	

Phase_2_PlayerController
+ MySaveGame : My Save Game + Phase_2_Mission_Update_HUD : Phase 2 Mission Update HUD + Phase_2_House_Master_HUD : Phase 2 House Master HUD + Phase_2_House_Master_BP : HM BP + PJP_2_BP : PJP 2 BP + PJP_2_AnimBP : PJP 2 Anim
+ Teleport_Player_To_Last_Location () : void + House_Master_Convo_Done () : Multicast Delegate + Cast_To_PJP_2_AnimBP () : Void + Phase2_Mission_Update_HUD_Creation () : void + Create_Phase2_House_Master_HUD () : void + Get_House_Master_BP () : void + House_Master_HUD () : void + Getting_Room_Info_Finished () : void

HM_BP
+ PJP_2_BP : PJP 2 BP + Player_Interaction : Box Collision + House_Master : Skeletal Mesh
+ Is_Standing_On_The_Player_Interaction () : MultiCast Delegate + Start_Idle_Anim () : Void

Phase2_Parents_BP
+ Mom: Skeletal Mesh + Dad : Skeletal Mesh + Player_Interaction : Box Collision + PJP_2_BP : PJP 2 BP + bCan_Interact? : boolean

Admission Result HUD
+ My_Save_Game : My Save Game + VMission_Update_HUD : Phase 1 Mission Update HUD + Submit : Button + Written_Exam_No_Palette : Text
+ Get_Save_Game_Sys() : My Save Game + Set_Written_Exam_No_To_Palette () : void + Get_Mission_Update_HUD () : void + Set_InputMode_ShowMouse () : void + Save_Level_Name_And_Reset_Players_Last_Location_In_Save_Game () : Name

Phase1_Mission_Update_HUD
+ bMission1_Hide: boolean + bMission2_Hide: boolean + bMission3_Hide: boolean + bMission4_Hide: boolean + MySaveGame : My Save Game + Phase_1_PlayerController : Phase 1 PlayerController + Mission_1 : Text + Mission_2 : Text + Mission_3 : Text + Mission_4 : Text
+ Get_Save_System () : Save Game + Update_Mission1 () : void + Hide_Mission1() : void + Update_Mission2 () : void + Hide_Mission2() : void + Update_Mission3 () : void + Hide_Mission3() : void + Update_Mission4 () : void + Hide_Mission4() : void

Question Paper HUD
+ Q1 A1 : Check Box + Q2 A2 : Check Box + Q3 A2 : Check Box + Q4 A2 : Check Box + Q5 A3-1 : Check Box + Total_Marks : Float + Saver Sub Class : My Save Game + Save_Stuffs_PC : Save Stuffs 1
+ Add_Marks(CheckBox_No) : Float + Save_Total_Marks_AdmSt(Total_Marks) : void + Get_Save_Stuffs () : void + Exam_Finished () : Multicast Delegate

PJP_2_BP
+ Pitch: Float + TurnRight: Boolean + TurnLeft: Boolean + Phase2_AnimBP: PJP 2 Anim + Main_Camera: Component + Cinematic_Camera : Component + Mesh: Skeletal Mesh + Phase2_PlayerController : Phase2_PlayerController
+ Get_Guard2_BP() : Boolean , Guard2_BP + Sitting_On_Chair_Exam() : Multicast Delegate + Sitting_On_Chair_Viva() : Multicast Delegate + Guard_Checking_Done() : Multicast Delegate + Showing_ID_AnimFunc() : Void + Enable_Input_After_Getting_Room_Info() : Void

Save_Stuffs1
+ Saver_Coll : Box Collision + Mission_Updater_Coll : Box Collision + Respawn_Location : Box Collision + PBP_1_BP : PBP 1 BP + Last_Location_This : Vector + MySaveGame : My Save Game + VMission_Update_HUD : Phase 1 Mission Update HUD + Last_Rotation_This : Rotator
+ Get_Players_Last_Location () : Vector,Rotator + Save_Mission1_Visibility() : void + Save_Players_Last_Location () : void + Save_Mission2_Visibility () : void + Deactivate_Collisions () : void + Activate_Collisions () : void + Get_Mission_Update_HUD () : void

Save_Stuffs2_Inst
+ Save_Mission_3_Visibility () : void

Save_Stuffs3_Inst
+ Save_Mission_4_Visibility () : void

P2_Save_Stuffs
+ Saver_Coll : Box Collision + Mission_Updater_Coll : Box Collision + Respawn_Location : Box Collision + PJP_2_BP : PJP 2 BP + Last_Location_This : Vector + MySaveGame : My Save Game + P2_VMission_Update_HUD : Phase 1 Mission Update HUD + Last_Rotation_This : Rotator
+ Get_Players_Last_Location () : Vector,Rotator + Save_Mission1_Visibility() : void + Save_Players_Last_Location () : void + Save_Mission2_Visibility () : void + Deactivate_Collisions () : void + Activate_Collisions () : void + Get_Mission_Update_HUD () : void

P2_Save_Stuffs2_Inst
+ Save_Mission_3_Visibility () : void

Chair_BP
+ Box: Box Collision + Has_Touched_The_Chair : Boolean

MySaveGame
+ Question_Paper_Marks_Save: Float + Last_Location_Save : Vector + Mission1_Hide_Save : Boolean + Mission2_Hide_Save : Boolean + Mission3_Hide_Save : Boolean + Mission4_Hide_Save : Boolean + Last_Rotation_Save : Rotator + Open_Level_Name : Name
+ Save_Total_Marks_AdmSt(Insert_Marks) : Void

Phase2_Mission_Update_HUD
+ bMission1_Hide: boolean + bMission2_Hide: boolean + bMission3_Hide: boolean + bMission4_Hide: boolean + MySaveGame : My Save Game + Phase_1_PlayerController : Phase 1 PlayerController + Mission_1 : Text + Mission_2 : Text + Mission_3 : Text + Mission_4 : Text
+ Get_Save_System () : Save Game + Update_Mission1 () : void + Hide_Mission1() : void + Update_Mission2 () : void + Hide_Mission2() : void + Update_Mission3 () : void + Hide_Mission3() : void + Update_Mission4 () : void + Hide_Mission4() : void

4.2 Relationships among classes

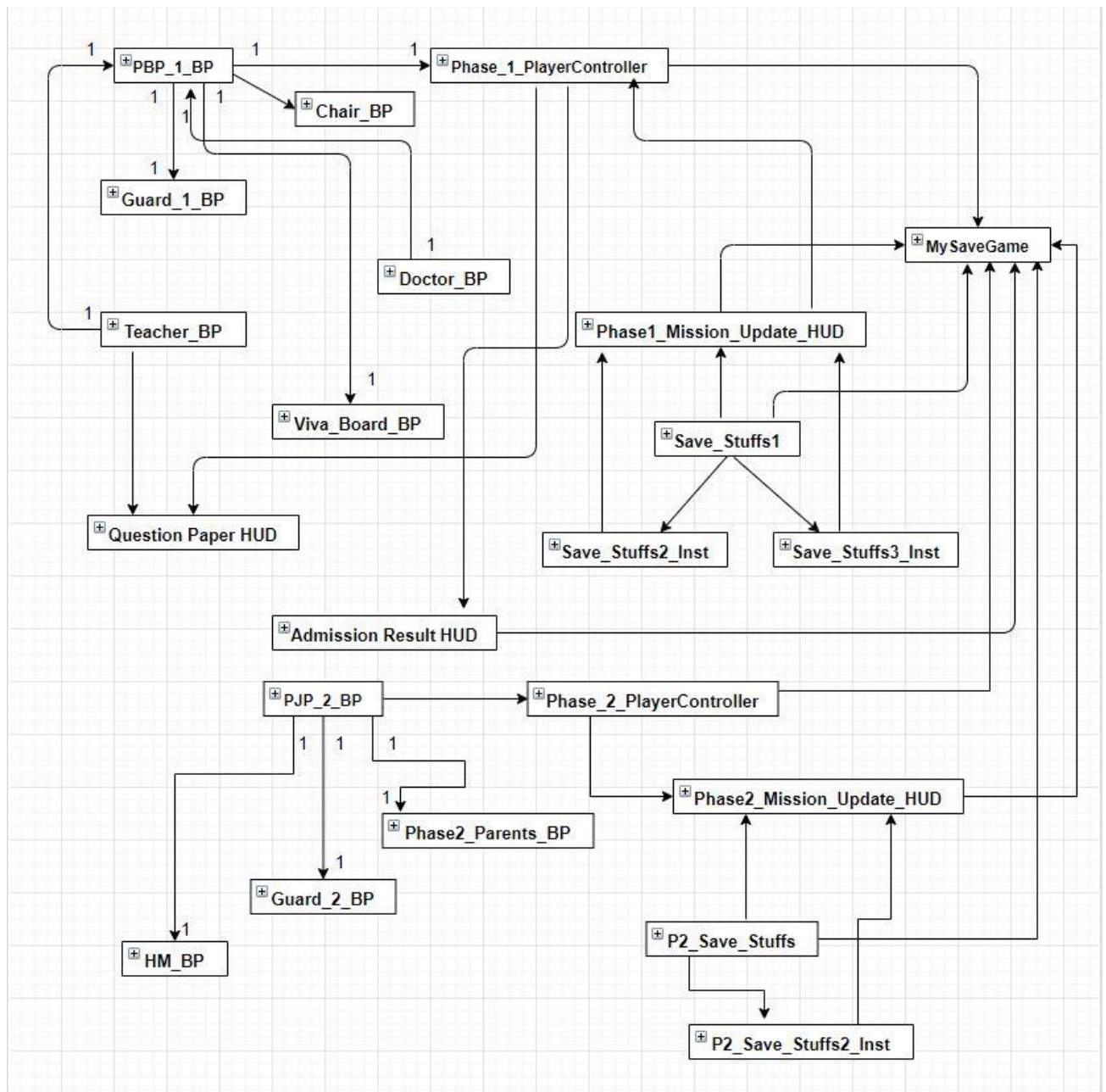
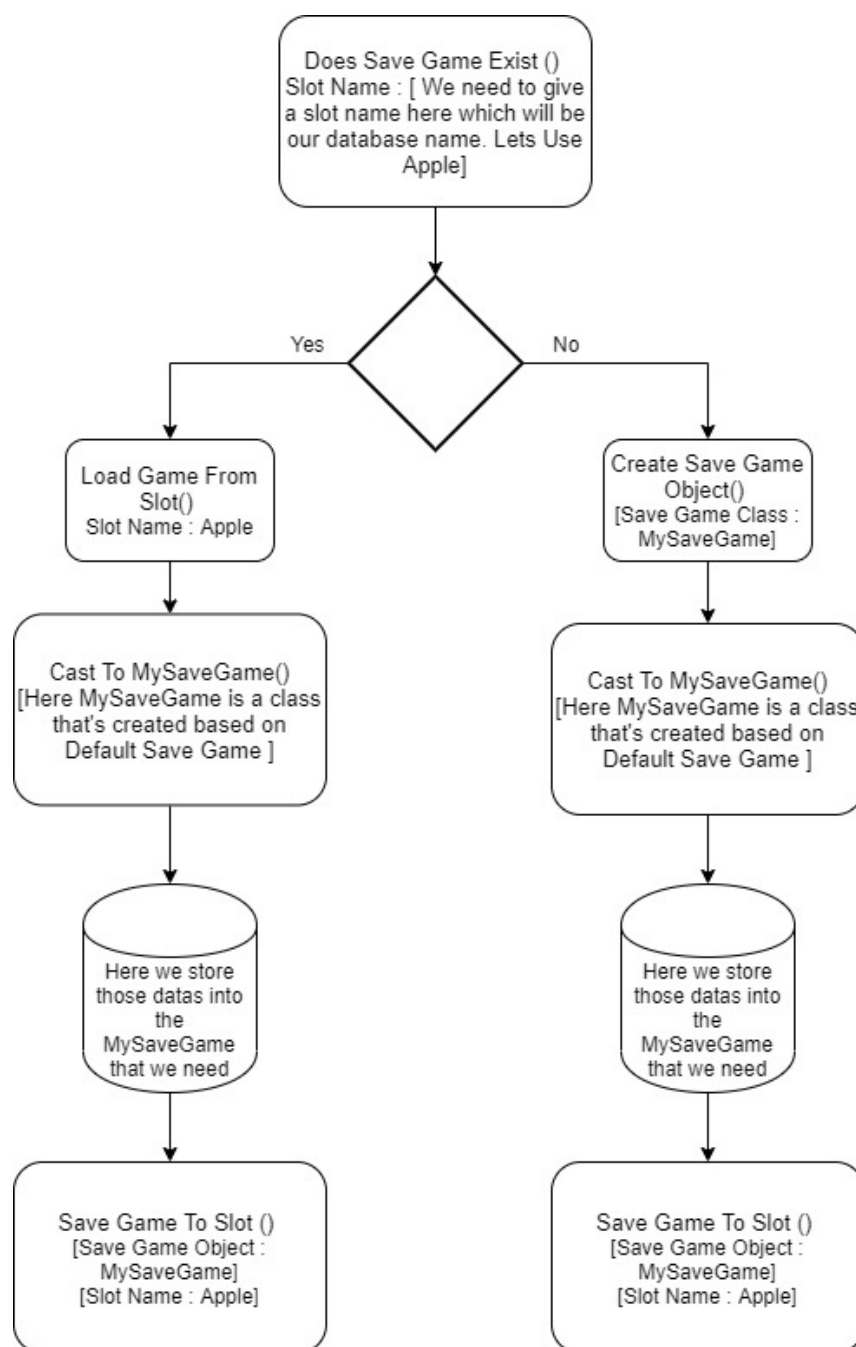


Fig: Relationship among classes

4.3 Database Design Diagram

In this project we didn't need any sort of external database system like MySQL, mongo dB, oracle etc.

We used Unreal Engine 4's advanced integrated file saving system where we can easily store and retrieve data from file system. For Demonstration purpose we can take a look at this diagram



Every time we want to save any data to our file or so-called data storage, we need to follow this process otherwise the data's won't store properly.

4.4 Development Tools & Technology

1. For Developing the game, we used Unreal Engine 4 (version 4.23.1). This game developing engine has been used in this project for programming (blueprint - visual programming), 3D World Design, Performance Optimization, Debugging and for Visual effects.
2. For Making the 3D characters and animating them Blender, Mixamo fuse and Mixamo's web-based animation system has been used.
3. For 3D Modeling, Autodesk's Maya has been used.
4. For Texturing the 3D objects Allegorithmics Substance Painter and Substance Alchemist has been used.
5. For some image processing, Photoshop has been used.

Later in future releases we will have to use Pixologics ZBrush, Marvelous Designer and Speed Tree for further development.

5. System Testing

5.1 Testing Features

5.1.1 Features to be tested

1. Firstly, I need to check whether the Mission 1 is updating or not on component begin overlap
2. Next, I need to check whether the Save Stuffs1 is actually saving my location in 3D world and storing it into the database
3. Next, I need to check after end overlap with the Mission Updater Coll in Save Stuffs class whether it updating the next mission or not.
4. After seating in the exam hall, I need to check if the Question paper UI is actually adding to the screen or not.

5. Answering the questions and submitting should store the calculated data's in the data storage system which is a major thing to test.
6. After leaving the exam room I should make sure whether the Save Stuff's Inst is actually saving my location or not and updating my next mission.
7. In Viva board I must touch the collision in chair in order to be able to take a seat which I need to make sure obviously.
8. On Phase 2 loading I need to properly reset the previous updated missions otherwise they will overlap and cause system failure.
9. 3D objects which needs to be properly optimized and their UV's needs to be unwrapped properly.
10. Animations needs to be applied to the related skeletons

5.1.2 Features not to be tested

Since this is a 3D game, every possible activity needs to be tested properly. Perfect performance makes a game more enjoyable.

5.2 Testing Strategy

A testing methodology is a layout that portrays the testing approach of the product advancement cycle. It is made to educate venture chiefs, analyzers, designers about certain key issues of the testing procedure. Test procedures depict how the item dangers of the partners are alleviated at the test-level, which sorts of testing are to be performed, and which section and leave criteria apply. They are made dependent on improvement configuration records. Framework configuration records are essentially utilized, and once in a while applied plan reports might be alluded to.

5.3 Testing Environment

A testing situation is an arrangement of programming and equipment for

the testing groups to execute experiments. As it were, it bolsters test execution with equipment, programming and system designed.

For our project the testing environments are

1. 3D objects
2. Animations and 3D characters
3. Programming with tested data's
4. Hardware configuration
5. Visual Representations
6. Visual Effects
7. Optimization
8. User Interface

In a test environment the stakeholders are the testers basically. In 3D games people gets the freedom to share their opinions and give report if any bug found.

5.4 Test Case

A Test Case is a lot of conditions or factors under which an analyzer will decide if a framework under test fulfills necessities or works effectively.

The way toward creating experiments can likewise help discover issues in the prerequisites or structure of an application.

Test situations are fairly unclear and spread a wide scope of potential outcomes. Testing is tied in with being quite certain

For example, Think of the Question Paper marks saving process

- 1.First the question paper should add to the player screen.
- 2.Next we need to check if we are properly able to answer the questions

3. Then after answering them the UI should calculate the marks

4. Finally it should store it into the database.

These are individual test cases.

6. Project Summary

6.1 GitHub Link

https://github.com/Nibir1/Life_At_BCC.git

6.2 Critical Evolution

In this project I faced some issues but I was able to solve them in time. They were major issues which could lead the project astray.

Issues like:

1. Data wasn't storing properly
2. Missions were overlapping
3. There were some issues related to players movement
4. Some missions were not updating properly
5. Player was spawning at random locations

6.3 Limitations

This project is only Phase 1&2 now. We are working on future releases. So, based on those releases the current situation may change and may not stay the same. For now, we have to rely on these portions.

We can't visualize the full curriculum of this game yet but it will give you an understanding about what we are actually making.

6.4 Future Scope

We have high hopes with this project. If we can develop this full product and bring it to the market then there may be a chance of making good game development team in Bangladesh.