CPSC 2720 [Spring]

[Prison Break]

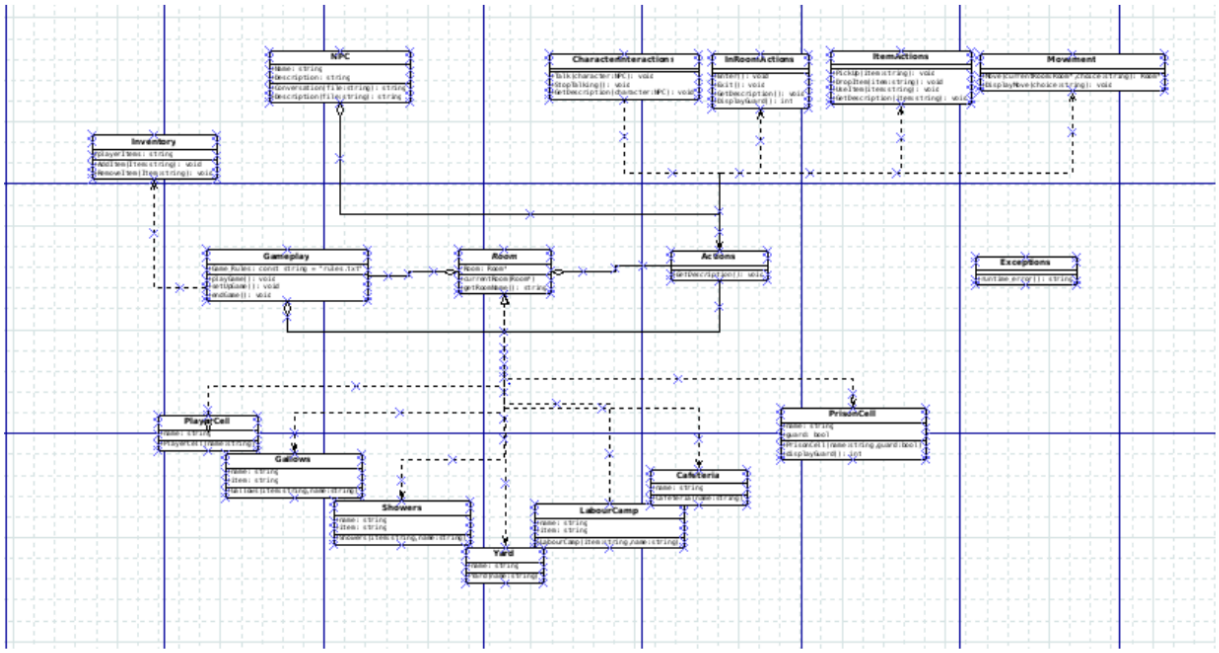
[Aria]

[Mir, Raheem Omogiate, Daniel Osman, Abdullah Todd Malcolm]

[March 6]

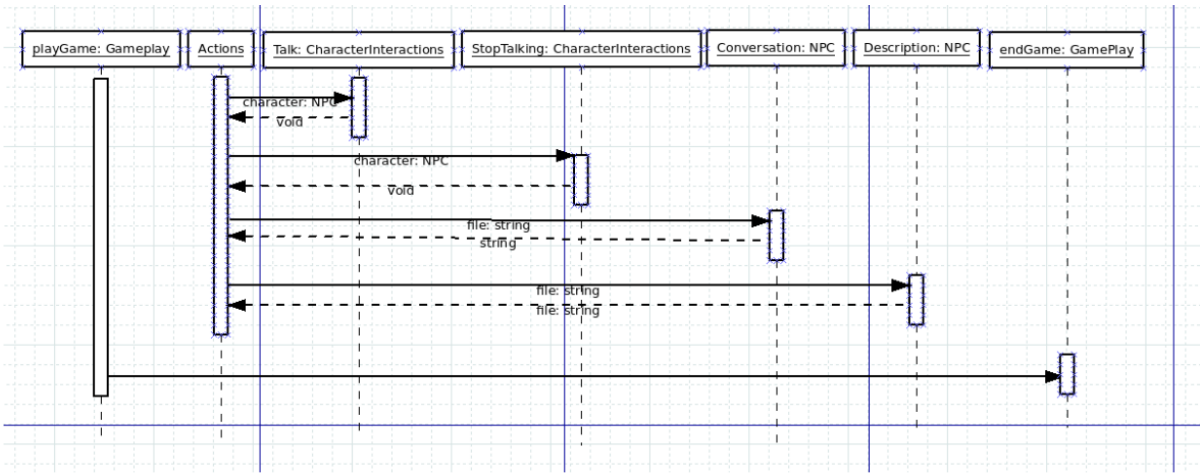
SOFTWARE DESIGN DESIGN:

– CLASS DIAGRAMS DESIGN

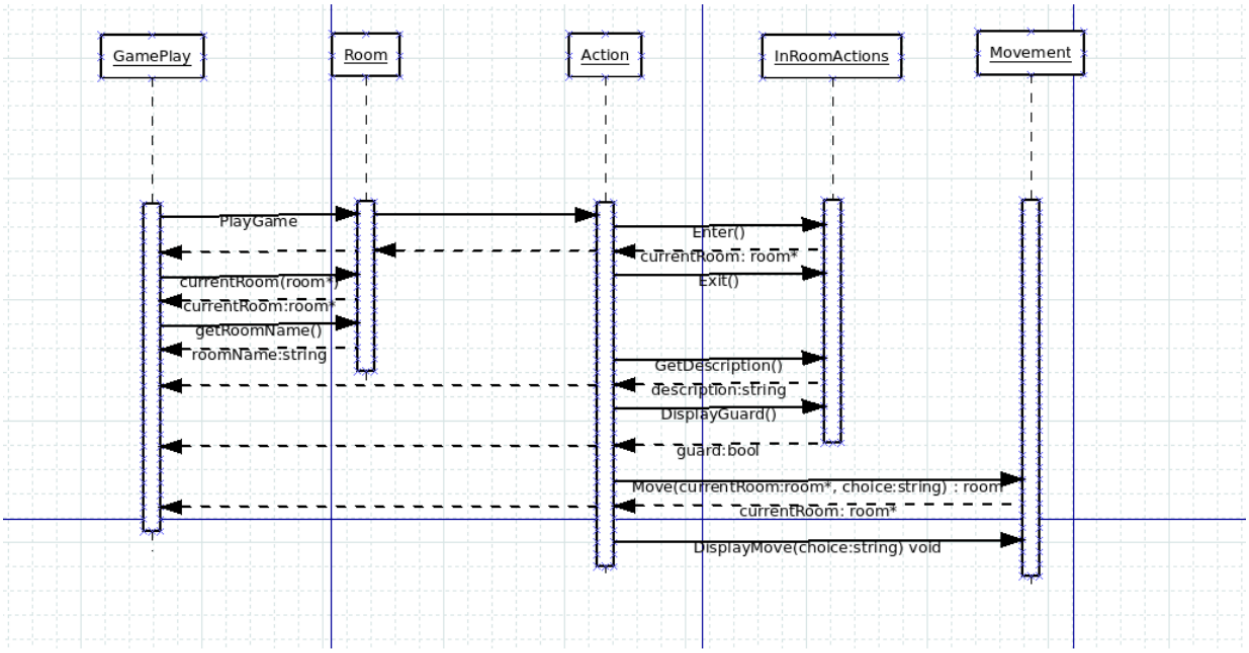



UML Diagram

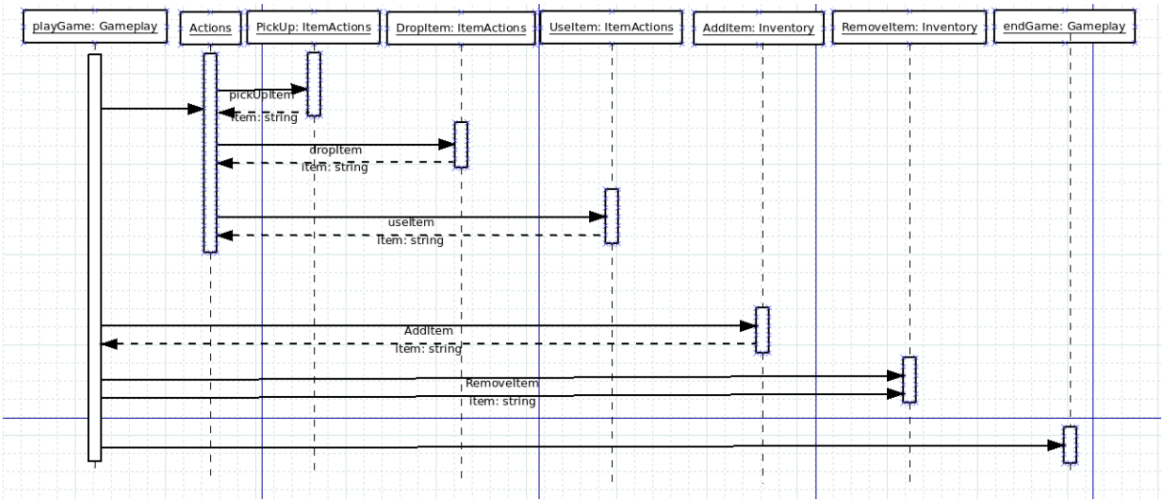
– SEQUENCE DIAGRAMS




Character Sequence Diagram




Environment Sequence Diagram




Object Sequence Diagram

CLASS DESCRIPTIONS

The goal of the game is to find a way to escape from prison after the player finds out that they are the legitimate heir to the throne. After the current king assassinated his brother to become a king. After the king had his coronation he imprisoned you and anyone with relation to you in order to keep his crown. You must escape and claim your throne! Classes: Room: this is a base class where all the rooms in the prison inherit from it. It has a function that returns the current room that we are in and it takes a pointer to class room. It also has a getRoomName function which is a virtual function. The classes that inherit from room are: playerCell: it has an attribute called name that returns the name of the room and a constructor that takes in that name. Gallows: it has two attributes name which stores the name of the room, and item which stores the item in the room, and it has a construct that takes in those attributes. Showers: it has two attributes name which stores the name of the room, and item which stores the item in the room, and it has a construct that takes in those attributes. Yard: it has an attribute called name that returns the name of the room and a constructor that takes in that name. LaborCamp: it has two attributes name which stores the name of the room, and item which stores the item in the room, and it has a construct that takes in those attributes. Cafeteria: it has an attribute called name that returns the name of the room and a constructor that takes in that name. PrisonCell: it has a name attribute called name that returns the name of the room, and a bool called guard that makes sure if there is a guard in the room. PriosnCell has a constructor that takes in name and guard, and a displayGuard function that returns true if there is a guard in the room. PlayerInventory: is a class that holds all the items that the player picks up from the rooms. It has an attribute that is protected and it is a vector called playerItems. And two methods called AddItem that adds to the inventory and dropItem that removes item form inventory. Exceptions: is a class that has a method that throws a runtime error when the player losses and makes them exist the game. NPC: this class has an attribute called name that holds the name of the non playable character that we want to interact with, it also has a string called Description that stores the NPC's description. This class has two methods Conversation and it takes in from a file and Description method that takes in from a file that has the description of the NPC. Action: this is a base class for all the actions that a player can preform in the game. Characterinteractions: this is a class that would enable us to interact with characters of the game, and it has the following methods. Talk this method enables us to talk with characters and its parameter is character of type NPC class. StopTalking

this method will stop the conversation , and GetDescription and its parameter is character of type NPC class. InRoomAction: this class holds all the actions that can be preformed in the room. It has the following methods enter that allow us to enter a room, Exit this will allow us to leave the room, GetDescription this method enable us to know whats inside the room, and DisplayGuard this method will allow us to see if there is a guard in the room. ItemActions: this class gives us all the actions that we can do with an item, and it has the following methods. PickUp this method picks up items and removes the item form the room and places it in the player inventory , dropItem this method removes the item from the player inventory to room, useItem this method allows us to use the item, getDescription this methods lets us know what is the item that we have. Movement: this class enable us to move from one room to another, and it has the following methods. Move this method enables us to move between rooms, and DisplayMove it displays all the ways that we could move