

CMPE 202

Software Systems Engineering

Final Presentation & Demo

Avengers – The Game

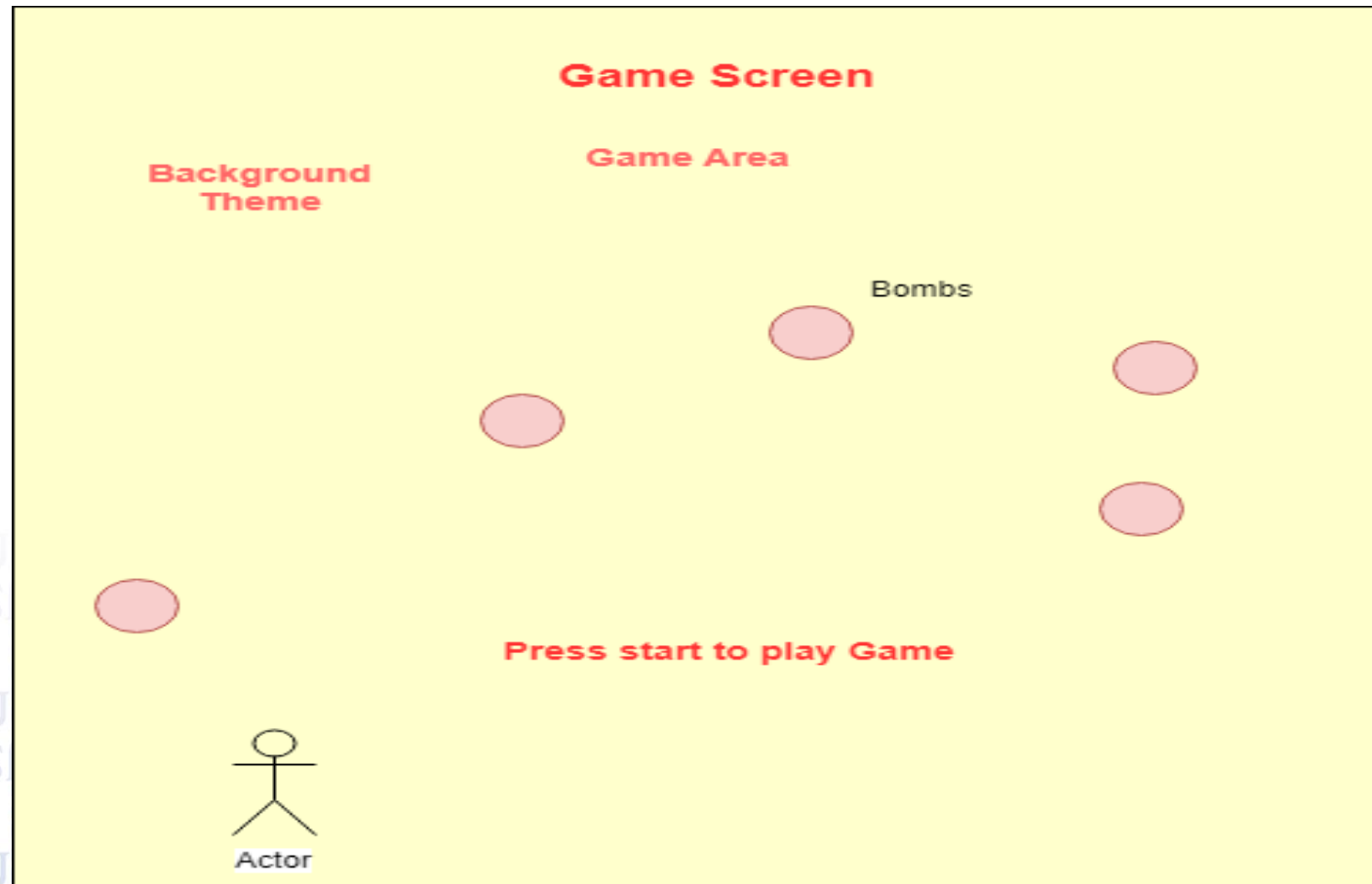


MEET THE TEAM

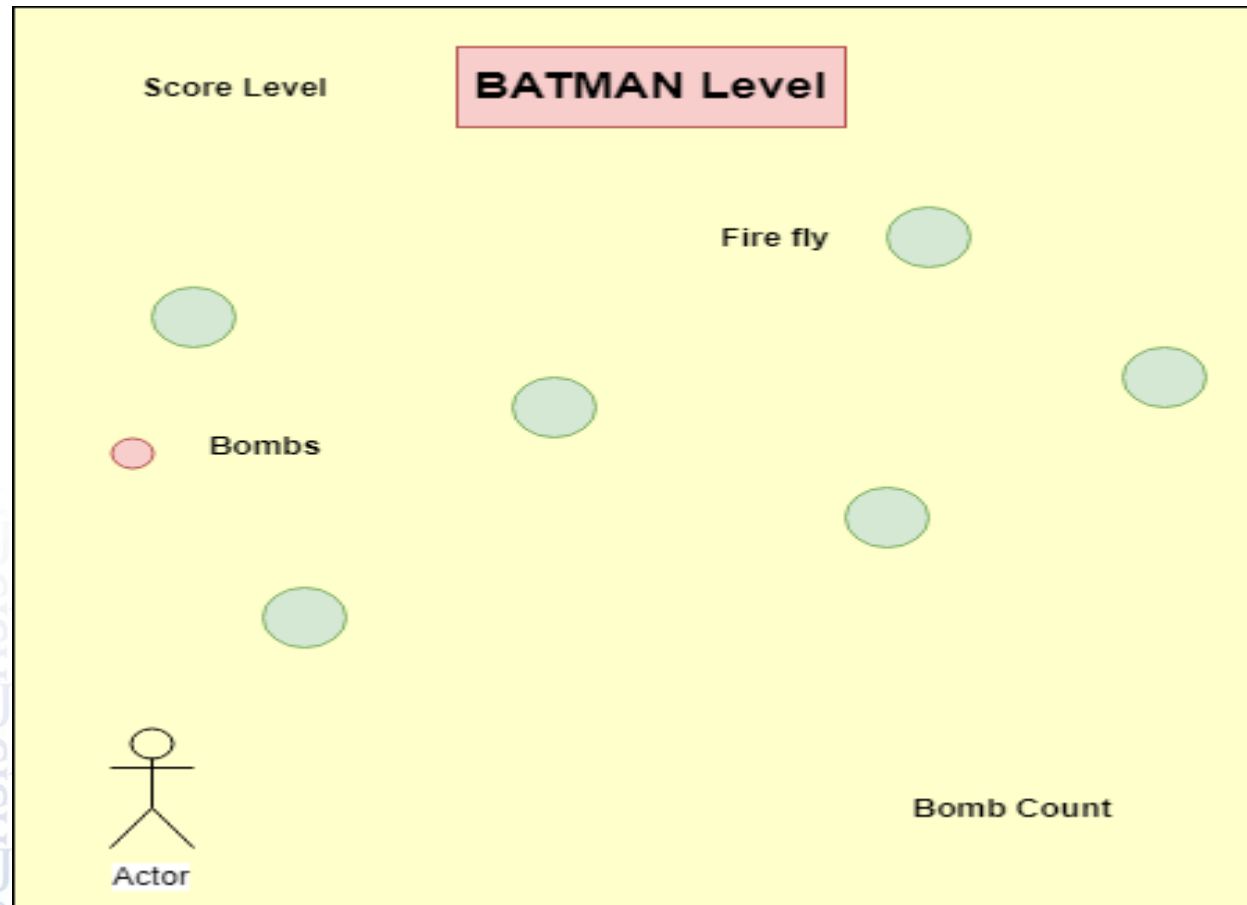
- >> Huy Vo
- >> Mangesh Tak
- >> Prateek Sharma
- >> Kalikalyan Dash
- >> Sricheta Ruj



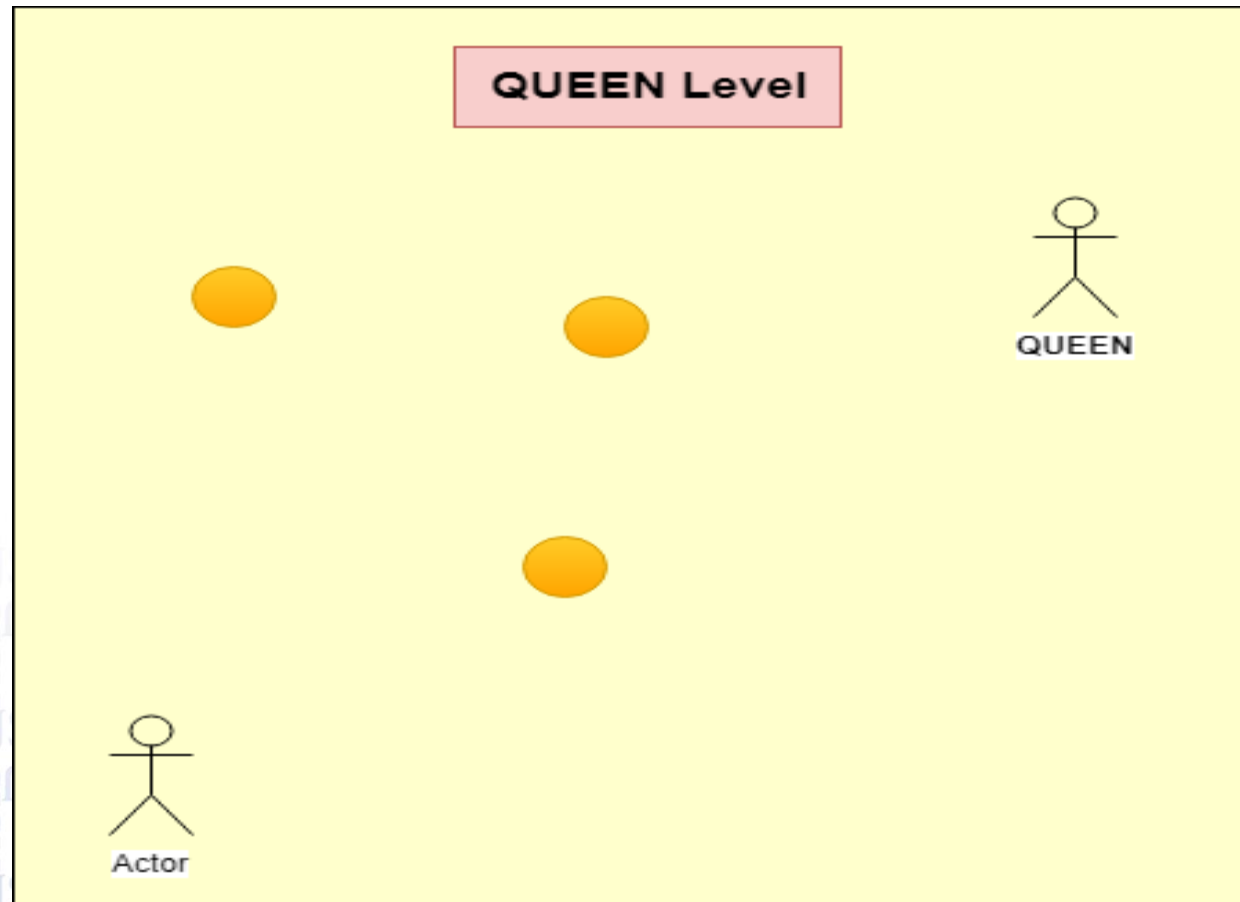
UI WireFrames



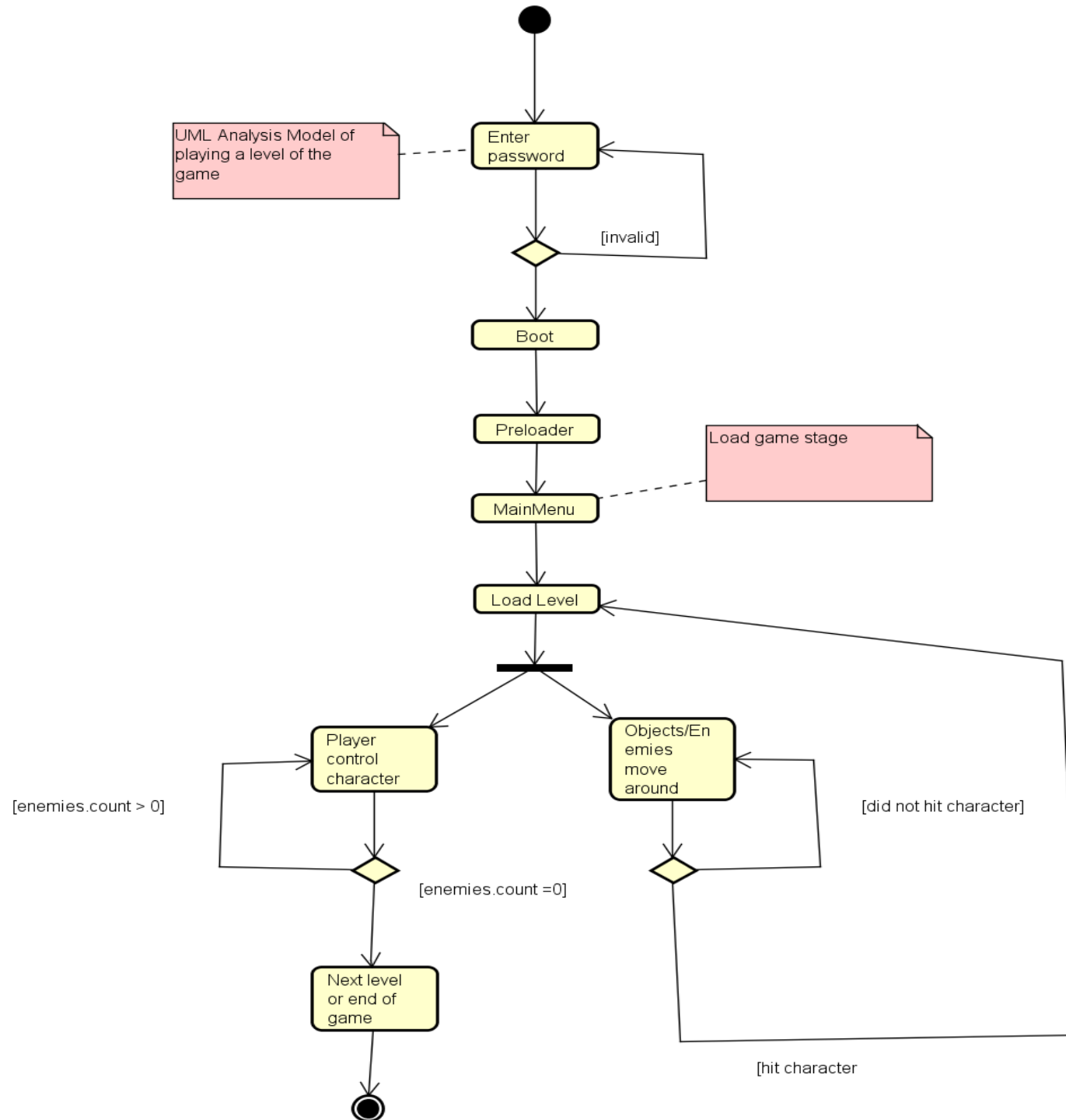
UI WireFrames



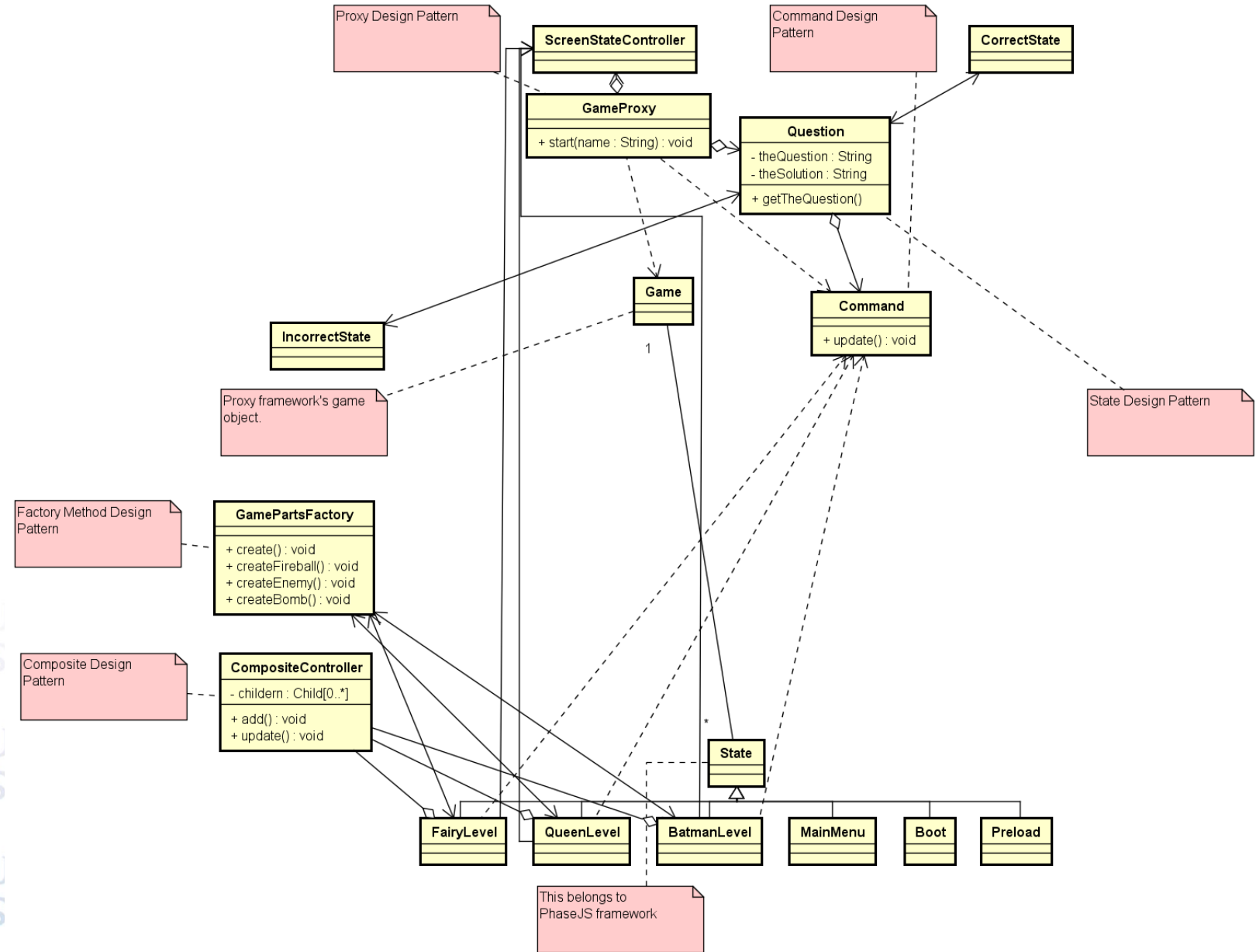
UI WireFrames



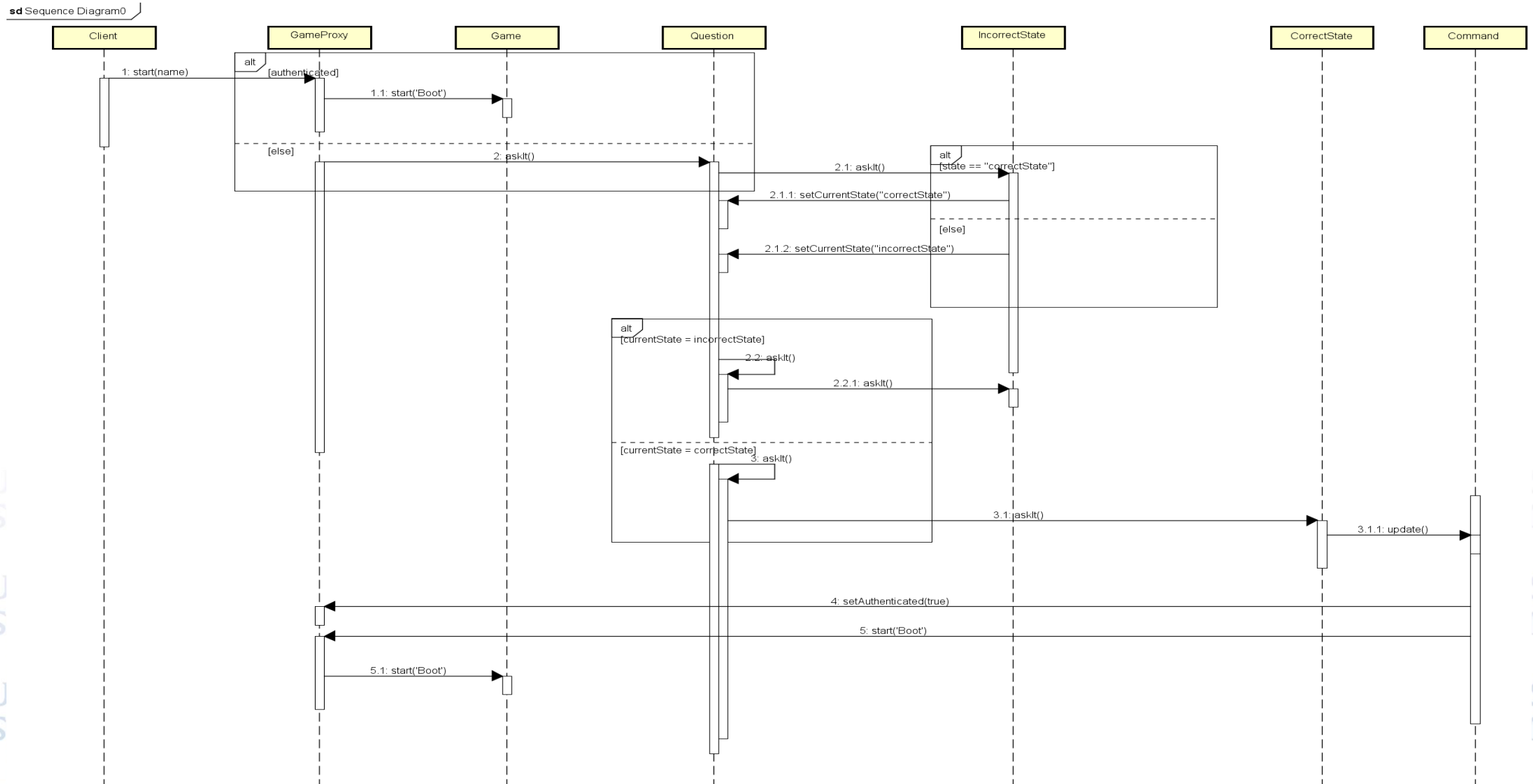
Activity Diagram



Class Diagram



Sequence Diagram



Design Patterns Implemented

>> Proxy – Initial Authentication

```
function GameProxy() {  
  
    this.game = new Phaser.Game(640 ,640 , Phaser.CANVAS , '');  
  
    this.authenticated = false;  
  
    this.game.state.add('Boot', Game.Boot);  
    this.game.state.add('Preloader', Game.Preloader);  
    this.game.state.add('MainMenu', Game.MainMenu);  
    // Load levels  
  
    this.game.state.add('Queen', Game.QueenLevel);  
    this.game.state.add('Batman', Game.BatmanLevel);  
    this.game.state.add('FairyLevel', Game.FairyLevel);  
  
    this.game.screenStateController = new ScreenStateController();  
    this.name = null;  
    this.theQuestion = new Question("type 123", "123", new Command(this, function (game) {  
  
        game.setAuthenticated(true);  
  
        game.start(game.name);  
  
    }));  
}
```

```
GameProxy.prototype.setAuthenticated = function (bool) {  
    this.authenticated = bool;  
}
```

```
GameProxy.prototype.start = function (name) {  
  
    if(this.authenticated===true)  
    {  
        this.game.state.start(name);  
    }  
    else  
    {  
        this.name = name;  
        this.theQuestion.askIt();  
    }  
}
```

Design Patterns Implemented

>> Command – Firing, Add new bombs, Update score, bombs left

```
this.compositeController.add(new Command(this, function (game) {  
    var left = game.input.keyboard.addKey(Phaser.Keyboard.A);  
  
    if( left.isDown){  
        game.player.animations.play('run');  
        game.player.scale.setTo(-1,1);  
        game.player.body.velocity.x -= game.playerSpeed ;  
        setTimeout(function(){ game.player.animations.play('idle'); }, 300);  
    }  
}));
```

```
this.compositeController.add(new Command(this, function (game) {  
    var up = game.input.keyboard.addKey(Phaser.Keyboard.W);  
  
    if(up.isDown){  
        game.player.animations.play('run');  
        game.player.scale.setTo(1,1);  
        game.player.body.velocity.y -= game.playerSpeed ;  
        setTimeout(function(){ game.player.animations.play('idle'); }, 300);  
    }  
}));
```

Design Patterns Implemented

>> Composite – Integrate all the components in each level

```
CompositeController.prototype.getChild = function (i) {  
    return this.children[i];  
};
```

```
CompositeController.prototype.hasChildren = function () {  
    return this.children.length > 0;  
};
```

```
CompositeController.prototype.remove = function (child) {  
    var length = this.children.length;  
    for (var i = 0; i < length; i++) {  
        if (this.children[i] === child) {  
            this.children.splice(i, 1);  
            return;  
        }  
    }  
};
```

Design Patterns Implemented

>> State – Switching Screen and maintaining screen

```
ScreenStateController.prototype.next = function (game) {  
    if(this.currentState === "MainMenu") {  
        game.state.start("FairyLevel")  
        this.setState("FairyLevel");  
    }  
  
    else if(this.currentState === "BeforeStart") {  
        game.state.start("MainMenu")  
        this.setState("MainMenu");  
    }  
  
    else if(this.currentState === "FairyLevel") {  
        game.state.start("Batman")  
        this.setState("Batman");  
    }  
  
    else if(this.currentState === "Batman") {  
        game.state.start("Queen")  
        this.setState("Queen");  
    }  
};
```


Design Patterns Implemented

>> Factory – Construct enemies, obstructions, bombs and bullets

```
GamePartsFactory.prototype.create = function (input) {  
    console.log("Create");  
  
    var object = null;  
  
    if (input === "enemies"){  
        console.log("create enemies");  
        object = this.level.add.group();  
        object.enableBody = true;  
        object.physicsBodyType = Phaser.Physics.ARCADE;  
    }  
  
    if (input === "bombs"){  
        object = this.level.add.group();  
        object.enableBody = true;  
        object.physicsBodyType = Phaser.Physics.ARCADE;  
        object.createMultiple(30, 'bomb');  
        object.setAll('anchor.x', 0.5);  
        object.setAll('anchor.y', 1);  
        object.setAll('outOfBoundsKill', true);  
        object.setAll('checkWorldBounds', true);  
    }  
}
```


Game Snapshots



Game Snapshots



Game Snapshots



Game Snapshots



Game Snapshots



Game Live on Heroku
<http://boringpeople.herokuapp.com>



Thank You

T.H.E E.N.D