



# Ranada Prasad Shaha University

-- an institution of Kumudini Welfare Trust of Bengal (BD) Ltd.

Department: COMPUTER SCIENCE AND ENGINEERING  
Trimester: Fall 2021

Program: Bachelor of Computer Science and Engineering  
Course Title: Software Engineering and Information System Design Lab  
Course Code: CSE 325

## ASSIGNMENT FOR FINAL EXAMINATION EVALUATION (ON THE PERIOD OF PANDEMIC COVID-19)

Stud\_Name: Md. Sakib Hossain

Stud\_ID: 18300021

Stud\_Batch: 14<sup>th</sup>

Submission Date: 16.01.2022

Course Teacher: Kingkar Prosad Ghosh

**NARAYANGANJ 2022**

## **1. Project Name:** Nintendo TB.

### **Abstract:**

1. The game is platform independent game.
2. This game can be able to run on Android, iOS, Windows & Website.
3. Tools and Technology using to build this game is Android Studio, libGDX, Java and Tiled Map Editor.
4. Implementing OOP concept here.

### **Description:**

A plumber named Mario he has limited time to reach his destination else will lose his job. His path is not straight but he knows how to overcome those situations. His job depends on time because he challenged his boss that he can do it in no time. If he able do that risky job he will get promotion with high salary.

## **2. Advantage of this project:**

Nintendo TB is challenging game. This game is made in such way that any player can win the game easily. Our purpose not only making happy our player while playing the game but also making motivated by giving them high success rate to win the game so that they can win easily.

Anyone can play this game. Controls are easily designed so that any player can able to play this game. Game environment suitable for any level of player.

**3. Programming language and Platforms:** Programming language and Platforms that are used in this game are: Java, Android Studio, LibGDX and Tiled Map Editor.

**Java:** Java is a programming language. Reason to choose Java for this project is-

1. It was designed to be easy to use and is therefore easy to write, compile, debug.
2. Java is Object-Oriented. This allows you to create modular programs and reusable code.
3. Java is platform-independent.

**Android Studio:** This is official IDE for android development. Not only we can build android we can also many more applications using this IDE. The reason to choose this IDE for my project is-

1. Faster Coding and Quick Iteration.
2. Fast and Feature-rich Emulator.
3. Boost your self-belief while coding
4. Robust Testing mechanisms
5. Powerful Development System.
6. Perfect for Teamwork.

**libGDX:** The reason why I choose libGDX to build this game is-

1. LibGDX is open source and it's free to use even for commercial games.
2. LibGDX allows you to explore each and every aspect of game development.
3. Using this you can create not only for android but also for desktop, iOS, MAC and also for website. So, if we build game using LibGDX no matter what type of operating system player have anyone can able to play this game.
4. It has strong community so that one can learn LibGDX easily.

**Tiled Map Editor:** This editor helps to create world for our game. You can create 2D based game world or 3D based game world. There various projections (Orthogonal, Isometric and Hexagonal). This is free to use and also easy to use.

#### 4. Attributes / Functions:

##### **MarioBros{**

```
create()  
dispose()  
render()  
}
```

##### **Hud{**

```
Hud()  
update()  
addScore()  
dispose()  
isTimeUp()  
}
```

##### **GameOverScreen{**

```
GameOverScreen()  
show()  
render()  
resize()  
pause()  
  
Resume()  
hide()  
dispose()  
}
```

##### **PlayScreen{**

```
spawnItem()  
handleSpawningItems()  
getAtlas()  
show()  
handleInput()  
  
render()  
gameOver()  
resizer()  
  
getMap()  
getWorld()  
pause()  
resume()  
hide()  
dispose()  
}
```

##### **Enemy{**

```
Enemy()  
defineEnemy()  
  
Update()  
hitOnHead()  
hitByEnemy()  
reverseVelocity()  
}
```

##### **Goomba{**

```
Goomba()  
update()  
defineEnemy()  
draw()  
hitOnHead()  
hitByEnemy()  
}
```

##### **ItemDef{**

```
ItemDef()  
}
```

##### **Brick{**

```
Brick()  
onHeadHit()  
}
```

##### **Item{**

```
Item()  
defineEnemy()  
update()  
draw()  
destroy()  
reverseVelocity()  
}
```

##### **Mashroom{**

```
Mashroom()  
defineEnemy()  
use()  
update()  
}
```

##### **InteractiveTileObject{**

```
onHeadHit()  
setCategoryFilter()  
getCell()  
}
```

##### **B2WorldCreator{**

```
B2WorldCreator()  
getEnemy()  
}
```

##### **Coin{**

```
Coin()  
onHeadHit()  
}
```

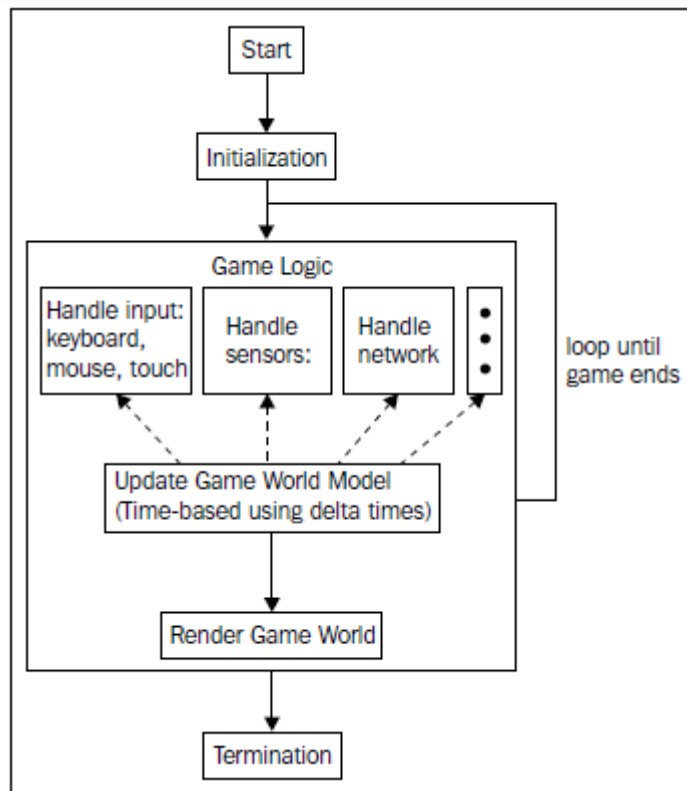
##### **WorldContactListener{**

```
beginContact()  
endContact()  
preSolve()  
postSolve()  
}
```

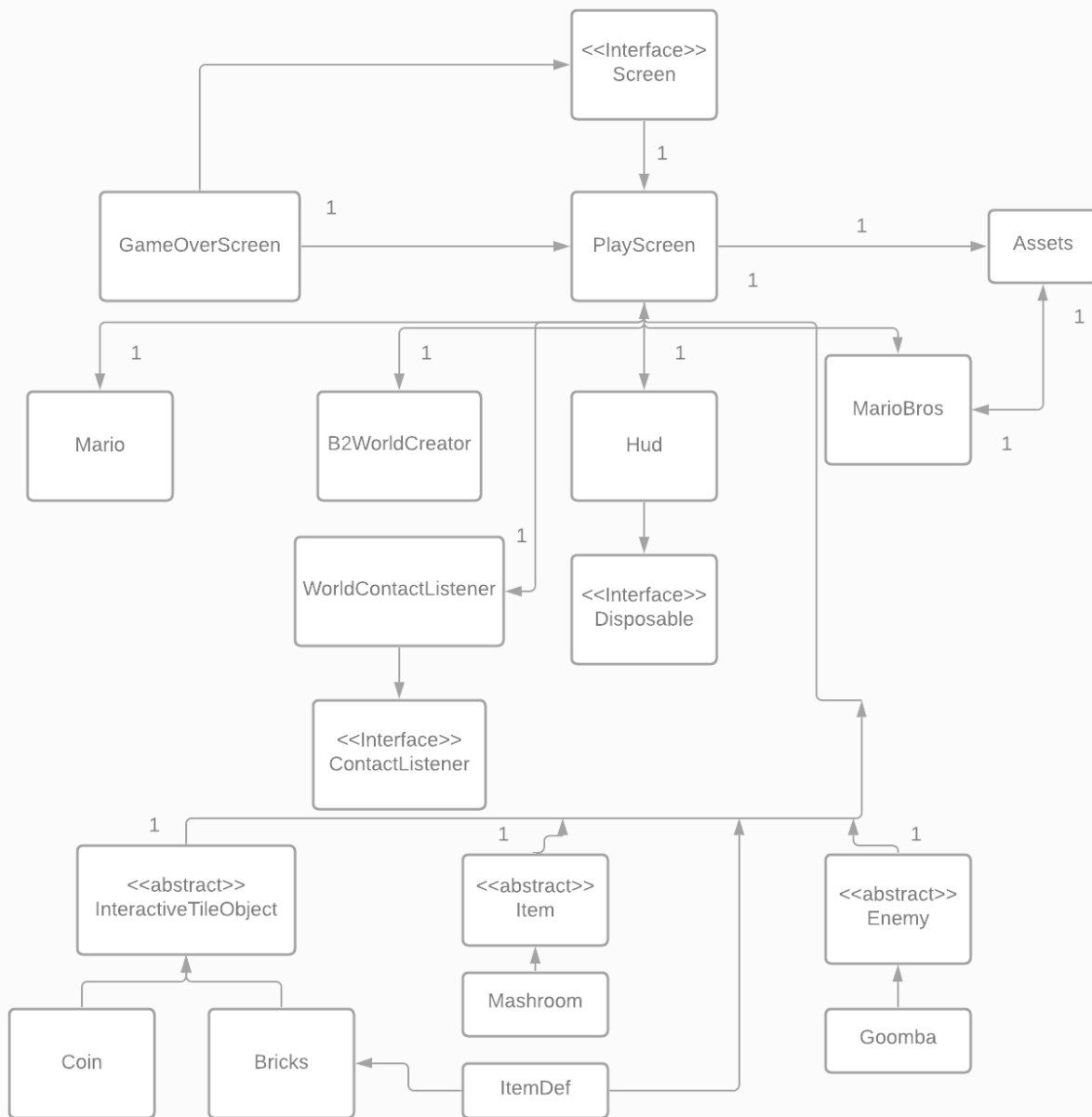
**5. Architecture:** From high level point of view, a game can be split up into two parts: Game assets and Game logic.

**Game assets** include everything that is going to be used as a kind of working material in your game, such as images, sound effect, background music and level data.

**Game logic** is responsible for keeping track of the current game state and to only allow a defined set of state transitions. These states will change a lot over time due to events triggered either by the player or by the game logic itself. To give a better idea of this take a look at the following diagram:



## Main Architecture of the game given below:



**6. Conclusion:** This game would be fun. Code optimized in such way so that our game should take less memory. So, Easy simple game everyone will have fun while playing this game.