The idea for this project came from a mini-game in the game Mario Party in which players have to knock their enemies out of a small platform. We expanded upon this idea and created a Java game with advanced physics and a plethora of extra features. We call it Bumper Ball Battle.

### Game Features

**Player-Controlled Unit**
Each player controls a sphere. Player can move the sphere around the map using keyboard controls with the goal of knocking all of the other players off of the map and being the last person standing.

**Game Physics**
We incorporated physics interaction into the game mechanism. Gravity, Elasticity, Velocity, Impulse, Particle Effects, etc. are used in calculating and implementing movements, abilities and power-ups.

**Ability**
Players use their selected Ability to enable the sphere to do special maneuvers, which are crucial to winning the game!

**Power-Up**
A special random “buff” that appears over time in a designated spot on the map that gives a player who picks it up an advantage for a limited time.

**Map**
The platform where players decide to do battle on. There are a total of 5 maps to chose from, and they all feature different terrain, objects, and some even vary in size.

**GUI**
A graphical user interface that allows players to change various settings such as number of players, abilities, ball color, map selection and key bindings.

### Future Work

**Bumper Ball Battle** is very expandable due to the language it is written in. We are planning on working on the following new features in the future:

**Online Mode**
Enables multiplayer gaming over a network. To accomplish this, we will build a server with a match-making algorithm that pits up random players against each other.

**Artificial Intelligence**
Computer-controlled “bots” in the single player and multiplayer modes of our game, that function as extra players when none are available.

**Mobile Application**
jME3.0 also supports mobile cross-platforming with Android OS. Instead of keyboard and mouse key-mapping, we will need to implement a touch listener for user inputs.

### Game Mechanics

**Players** must knock their opponents off the map to win the game. Abilities and power-ups both have cool downs – a set period of time where the ability cannot be used again upon use. When the winner is decided, the game declares the winner through on-screen text and returns back to the game setting GUI.

**The map** starts to fall apart in a spiral fashion when the timer in the top-center expires. This gives players a smaller platform to play on over time, thereby decreasing the chances of survival for every player and increasing the chances of a winner being declared.