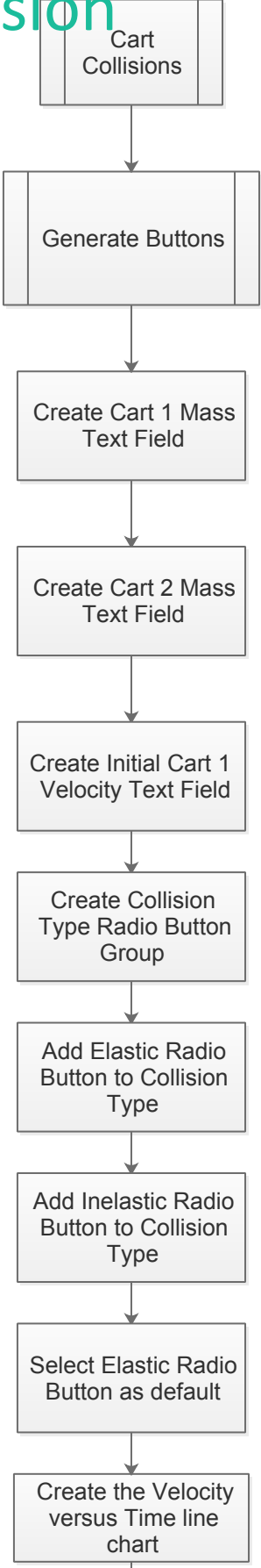


Custom Classes

Cart inherits PhysicalObject:
private float mass
public void applyForce(Vect2D force): Unused for this object and thus unimplemented.
public float getMass()

static class ClsColl:
public static Button btnStart
public static Button btnDone
public static Button btnReset
public static Button btnHelp
public static Button btnBack
private static TextField txtM1
private static TextField txtM2
private static TextField txtVel
private static RadioButton rdbtnEla
private static RadioButton rdbtnInEla
public static void doStartBtn(): The code triggered when the user presses btnStart
public static void doDoneBtn(): The code triggered when the user presses btnDone
public static void doResetBtn(): The code triggered when the user presses btnReset
public static void doHelpBtn(): The code triggered when the user presses btnHelp
public static void doBackBtn(): The code triggered when the user presses btnBack

interface ICollisions:
Interface used for storing all constants relating to elastic and inelastic collisions.

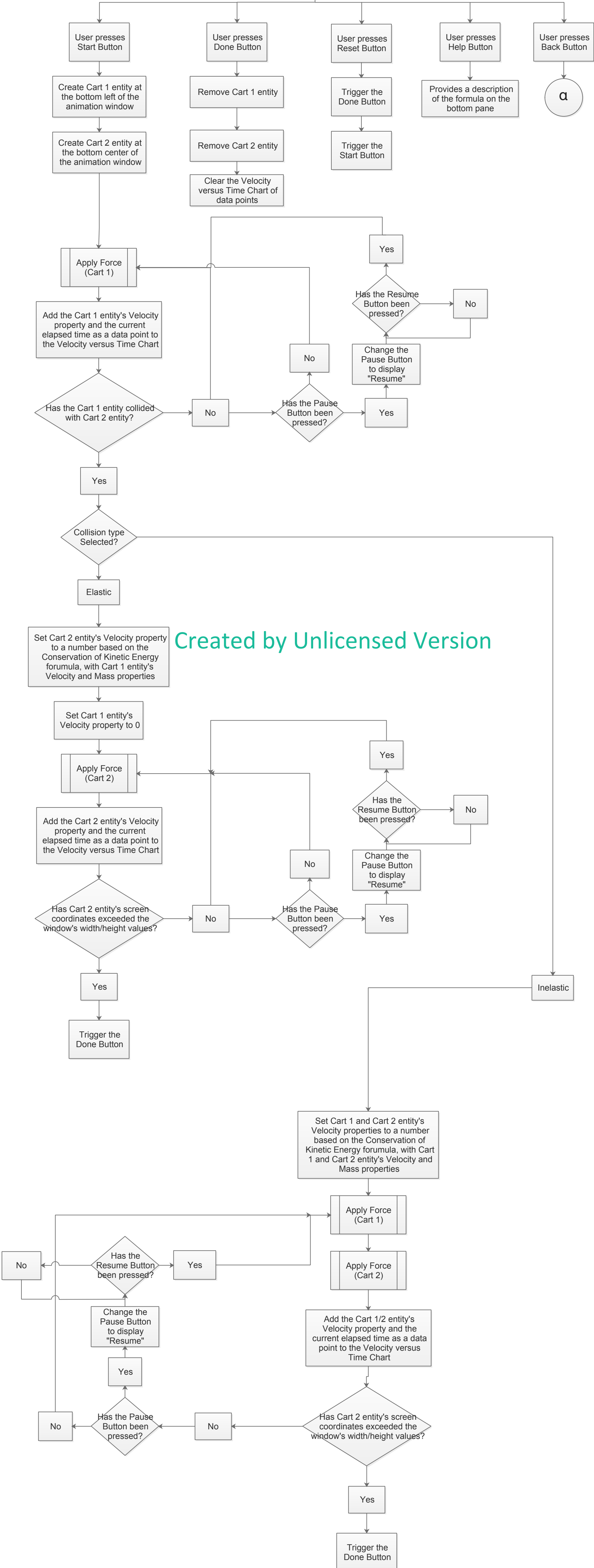
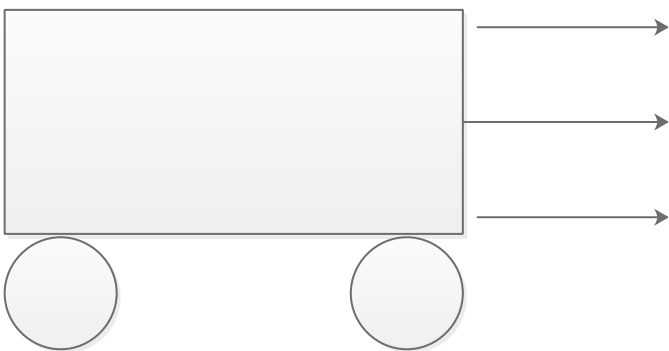


General Implementation

Collisions Formulas
Conservation of Kinetic Energy (Elastic): $m_1 * V_1 = m_2 * V_2$
Conservation of Kinetic Energy (Inelastic): $m_1 * V_1 = (m_1 + m_2) * V_2$

Keywords for Variables
Cart 1: An instance of Cart entity, to be the source of impact
Cart 2: An instance of Cart entity, to be the target of impact
 m_x : mass of x
 V_x : velocity of x

Brief Explanation
The user will be able to choose the velocity of the source of impact, and decide the type of ensuing collision between it and the target of impact.

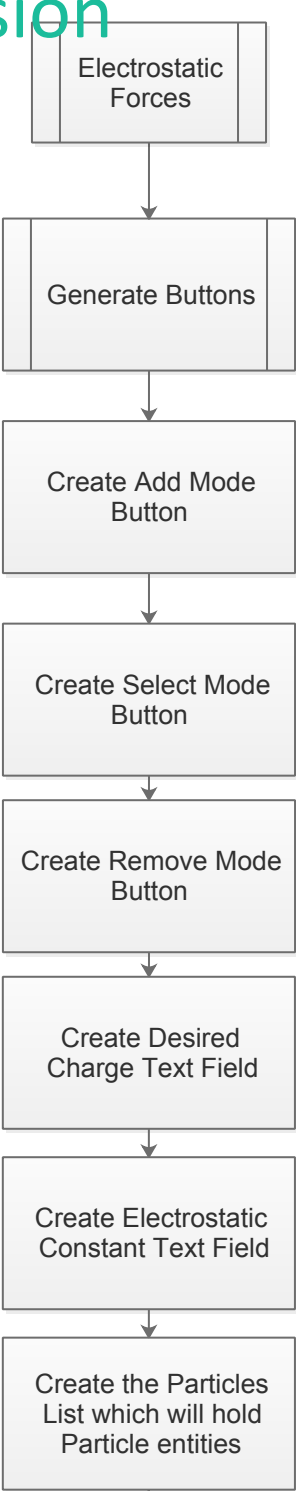


Custom Classes

Particle inherits PhysicalObject:
private bool selected
private float mass
private float charge
public Vect2D attract(Particle p2): Returns an attraction force based on the Mass property and displacement between the two Particle entities
public void applyForce(Vect2D force): Applies a force on the given Particle entity
public float getMass()
public void setCharge(float charge)
public float getCharge()
public void setSelected(bool selected)
public bool isSelected()

static class ClsEle:
public static List particles
private int mode
public static Button btnStart
public static Button btnDone
public static Button btnReset
public static Button btnHelp
public static Button btnBack
public static Button btnAdd
public static Button btnSelect
public static Button btnRemove
private static TextField txtCharge
private static TextField txtEleConst
private static RadioButton rdbtnEla
private static RadioButton rdbtnInEla
public static void doStartBtn(): The code triggered when the user presses btnStart
public static void doDoneBtn(): The code triggered when the user presses btnDone
public static void doResetBtn(): The code triggered when the user presses btnReset
public static void doHelpBtn(): The code triggered when the user presses btnHelp
public static void doBackBtn(): The code triggered when the user presses btnBack

interface IElectrostatic:
Interface used for storing all constants relating to Coulomb's law.

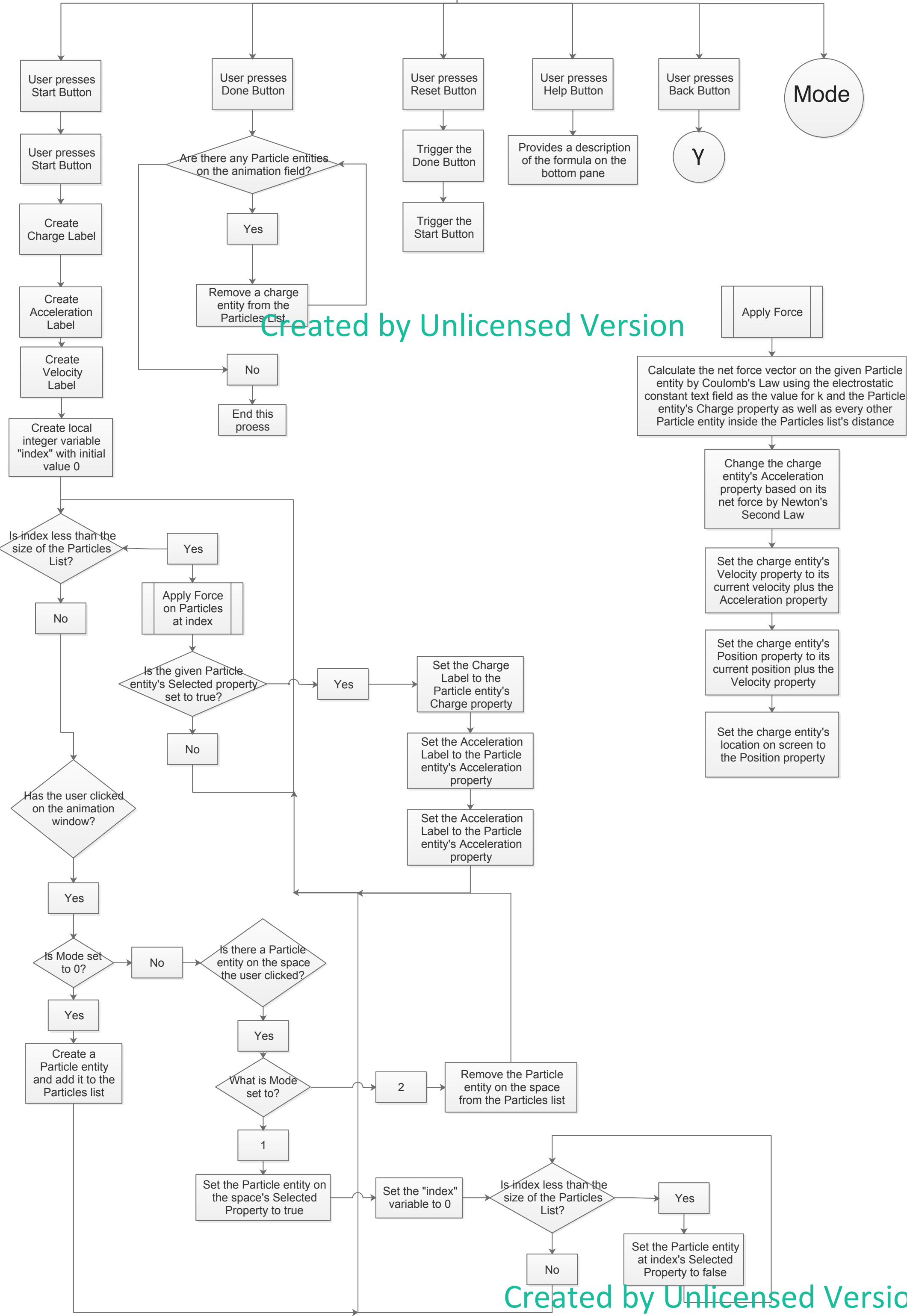
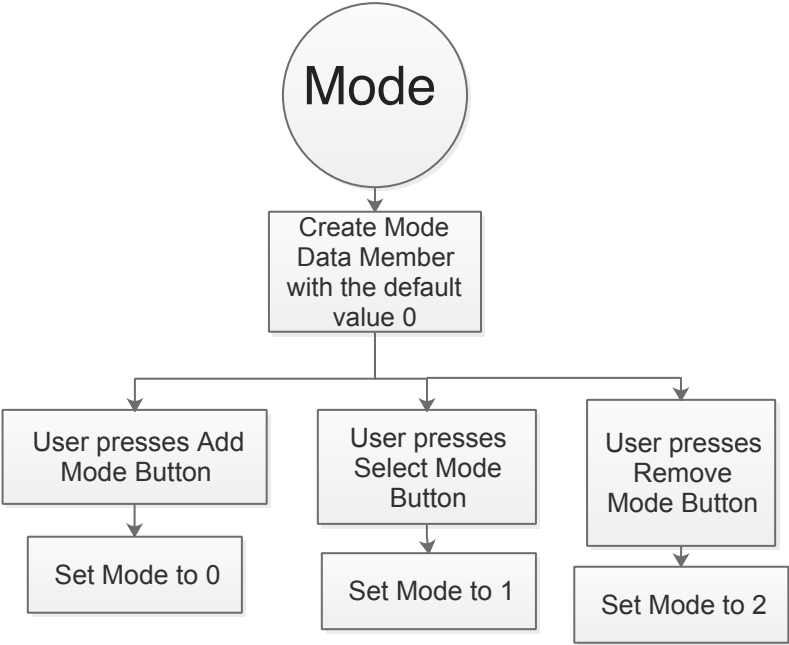


General Implementation

Electrostatic Forces:
Formulas
Coulomb's Law
 $F_e = k * q_1 * q_2 / r^2$
 $F_e = m_e * a$

Keywords for Variables
q_x: the charge of the given particle entity
r: distance between two different particle entities
m_x: mass of x
a: acceleration of the given entity

Brief Explanation
The user will be able to place charges onto an animation field and observe the electrostatic forces reacting on each other.

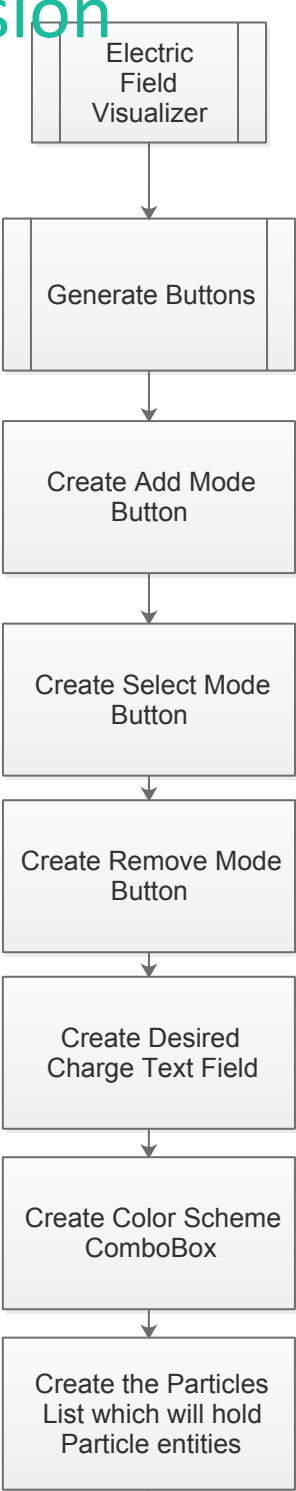


Custom Classes

EFieldPoint inherits PhysicalObject:
private bool selected
private float charge
public void applyForce(Vect2D force): Applies a force on the given EField entity
public void setCharge(float charge)
public float getCharge()
public void setSelected(bool selected)
public bool isSelected()

static class ClsFld:
public static List points
public static List charges
private int mode
public static Button btnStart
public static Button btnDone
public static Button btnReset
public static Button btnHelp
public static Button btnBack
public static Button btnAdd
public static Button btnSelect
public static Button btnRemove
private static TextField txtCharge
private static ComboBox cbxColor : user can select one of a set of color schemes for the electric field's strength
private static RadioButton rdbtnEla
private static RadioButton rdbtnInEla
public static void doStartBtn() : The code triggered when the user presses btnStart
public static void doDoneBtn() : The code triggered when the user presses btnDone
public static void doResetBtn() : The code triggered when the user presses btnReset
public static void doHelpBtn() : The code triggered when the user presses btnHelp
public static void doBackBtn() : The code triggered when the user presses btnBack

interface IField:
Interface used for storing all constants relating to the Electric Field Equation.



General Implementation

Electric Fields:
Formulas
Electric Field at a Point
 $E = k * q / r^2 * r^{\wedge}$
Sum of $E = E_1 + E_2 + E_3 + \dots$

Keywords for Variables
q: the charge of the given particle entity
r: distance between two different particle entities
E: Electric field at a given point
 r^{\wedge} : Unit vector in the direction from a given source charge to a point in space

Brief Explanation
The user will be able to place charges onto an animation field and observe the changes in the magnitude of the electric field at all points on the plane.

