Starlink Mission w/Starship

Hin Lui 301571436

IAT 265 A4

Process

Process Stage	Environment	Goal
1	Starbase	Integrate booster and spacecraft
2	Starbase	Load Starlink satellites and fuel
3	Starbase	Launch Starship (with Super Heavy booster)
4	Sky	Separate
5	Starbase	Land Super Heavy booster
6	Space	Deploy Starlink satellites
7	Sky	Re-enter Earth
8	Starbase	Land Starship

Elements (some)

Object

Rational: hold an object's properties with fields and capabilities with methods

Fields and responsibilities:

```
# PVector pos;                    // position
# Dimension dim;                  // dimension
# double scale;                   // scale
# double rotation;                // rotation angle
 + Object(PVector pos, Dimension dim); // constructor
# setShape();                     // initialize geom shapes or images
 + paint(Graphics2D g);          // do transformations and call draw()
# draw(Graphics2D g);             // actually draw the shapes
 + getPos();                     // return position
 + setPos(float x, float y);     // set to given position
 + setPos(PVector pos);          // set to given position
 + setPos(MouseEvent e);         // set to given position
```

Button

Rational: hold a button's properties with fields and capabilities with methods

Fields and responsibilities:

```
 # Button(PVector pos, PVector dim);  // constructor
  + clicked(MouseEvent e);              // if the mouse is on the object
```

Instruction

Rational: hold the instruction's properties with fields and capabilities with methods

Fields and responsibilities:

```
  - String[][] texts;              // the instructions
  - PVector default_pos;           // default position
  - Dimension default_dim;         // default dimension
  - Rectangle2D.Double body;       // body
  + Instruction();                 // constructor
 # setShape();                     // initialize body
 # draw(Graphics2D g);             // draw body and text
```

Booster

Rational: hold the booster's properties with fields and capabilities with methods

Fields and responsibilities:

```
  - boolean atPosition;            // if it is at the correct position
  - PVector targetPosition;        // the correct position
  - PVector default_pos;           // default position
  - Dimension default_dim;         // default dimension
  - Rectangle2D.Double body;       // body
  - Rectangle2D.Double fins;       // fins
  + Booster();                     // constructor
 # setShape();                     // initialize body
 # draw(Graphics2D g);             // draw body and text
  + drawPosition(Graphics2D g);    // draw a box at the correct position
  + atPosition();                  // return atPosition
  + getTargetPosition();           // return targetPosition
```

+ setAtPosition(boolean b); // set atPosition to b
+ move(float x, float y); // move by given distance

Main Process

1. The user will use the mouse to drag the booster and spacecraft to the correct position.
2. The user will use the mouse to drag the satellite and fuel to the correct position.
3. The user will use the mouse to press a button and the starship will start moving.
4. The user will use the mouse to turn the engines of the booster off and turn the engines of the spacecraft on to separate by clicking.
5. The user will use the keyboard (WASD or arrows) to adjust the landing position of the booster exactly at the tower for the arms to catch it.
6. The user will use the mouse to open the "door" of the spacecraft and drag the satellite out.
7. The user will use the mouse to turn the engines of the spacecraft off by clicking so the spacecraft will re-enter the Earth by gravity.
8. The user will use the keyboard to adjust the landing position of the spacecraft exactly at the tower for the arms to catch it.