User Manual

Version: 1.0
Platforms Supported: Windows NT, 98

General Overview
For a few seasons now, the characters of the popular series South Park have “accidentally” killed Kenny in almost every episode. Well, now it’s Kenny’s turn. Only Kenny doesn’t opt for revenge. After an unfortunate incident when Kenny mistook plutonium for a glass of Crazy Bob’s Green Chill-Aide, he gained supertoon powers capable of amazing feats. With all power concentrated in his left arm, Kenny helps the quiet town of South Park fight against evil do-ers like Mecha-Streisand and the evil Christmas Poo. The time is now. The time is Kenny Time!

Game flow
At game start, a bunch of Mr. Hanky will be placed all over the screen and Kenny placed in the lower left corner. User can merrily control Kenny to walk around the town of South Park collecting Mr. Hanky and depositing in the Christmas Poo collector. Mecha-Streisand, an evil mechanoid, flies around the scene terrorizing our hero. Kenny, using his arm, can contact Mecha-Streisand and kill her instantly. Mecha-Streisand has the power to kill Kenny as well. If Kenny can fight off Mecha-Streisand three times, Kenny saves the day and wins the game! If Mecha-Streisand gets to Kenny first, well, just like old times, Kenny dies and loses the game.

<table>
<thead>
<tr>
<th>Win</th>
<th>Kenny kills Mecha-Streisand</th>
<th>3 times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lose</td>
<td>Mecha-Streisand kills Kenny</td>
<td>1 time</td>
</tr>
</tbody>
</table>

This Version
This release will implement the first level of Kenny Time.
Sound feedback is included.
Kenny is our hero in this action packed game. You can control him via the provided User Interface. For a quick game play, you can use the keyboard.

Kenny can be moved entirely with your right hand. That’s right, you can drink coffee, eat Twinkies or use your cell phone while enjoying a rousing game if *Kenny Time!*

Movement Control:

*Number pad on keyboard*

<table>
<thead>
<tr>
<th>Num Lock</th>
<th>/</th>
<th>*</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 UP LEFT</td>
<td>8 UP</td>
<td>9 UP RIGHT</td>
<td>+ ARM UP</td>
</tr>
<tr>
<td>4 LEFT</td>
<td>5 CHOP</td>
<td>6 RIGHT</td>
<td></td>
</tr>
<tr>
<td>1 DOWN LEFT</td>
<td>2 DOWN</td>
<td>3 DOWN RIGHT</td>
<td>Enter</td>
</tr>
<tr>
<td>0</td>
<td>.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key Stroke</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Kenny moves to user’s RIGHT</td>
</tr>
<tr>
<td>4</td>
<td>Kenny moves to user’s LEFT</td>
</tr>
<tr>
<td>8</td>
<td>Kenny moves UP the screen</td>
</tr>
<tr>
<td>2</td>
<td>Kenny moves DOWN the screen</td>
</tr>
<tr>
<td>7</td>
<td>Kenny moves UP LEFT</td>
</tr>
<tr>
<td>9</td>
<td>Kenny moves UP RIGHT</td>
</tr>
<tr>
<td>1</td>
<td>Kenny moves DOWN LEFT</td>
</tr>
<tr>
<td>3</td>
<td>Kenny moves DOWN RIGHT</td>
</tr>
<tr>
<td>ARM</td>
<td>+ Raises Kenny’s mutant arm</td>
</tr>
<tr>
<td>5</td>
<td>Chops Kenny’s mutant arm</td>
</tr>
<tr>
<td>---</td>
<td>-------------------------</td>
</tr>
<tr>
<td>*</td>
<td>Extends Kenny’s mutant arm</td>
</tr>
<tr>
<td>-</td>
<td>Retracts Kenny’s mutant arm</td>
</tr>
</tbody>
</table>

Graphical User Interface:
Kenny’s movement can also be controlled via UI and the mouse.

Interaction:

<table>
<thead>
<tr>
<th>Character</th>
<th>Interaction</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mr. Hanky</td>
<td>Any part of Kenny colliding with any part of Mr. Hanky</td>
<td>Mr. Hanky is placed in Kenny’s hand</td>
</tr>
<tr>
<td>Mecha-Streisand</td>
<td>Any part of Mecha-Streisand colliding with Kenny’s left arm</td>
<td>Mecha-Streisand dies.</td>
</tr>
<tr>
<td>Poo Collector</td>
<td>Any part of Kenny colliding with any part of Poo Collector</td>
<td>Mr. Hanky is removed from Kenny’s hand</td>
</tr>
</tbody>
</table>

**Level 1 Characters:**

**Mr Hanky:**

Mr. Hanky, as cute as he is, needs to be placed back into his box.

Movement Control:
None. Mr Hanky is randomly placed in Kenny’s World.
Interaction:

<table>
<thead>
<tr>
<th>Character</th>
<th>Interaction</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kenny</td>
<td>Any part of Kenny colliding with any part of Mr. Hanky</td>
<td>Mr. Hanky is placed in Kenny’s hand</td>
</tr>
<tr>
<td>Mecha-Streisand</td>
<td>Any part of Mecha-Streisand colliding with any part of Mr. Hanky</td>
<td>Mr. Hanky does nothing. Mecha-Streisand bounces.</td>
</tr>
</tbody>
</table>

Poo Collector:

![Poo Collector Image]

This box is for used for Mr. Hanky collection. This is where he lives.

Movement Control:
None. 2 Poo Collectors are placed randomly in Kenny’s world.

Interaction:

<table>
<thead>
<tr>
<th>Character</th>
<th>Interaction</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kenny</td>
<td>Any part of Kenny colliding with any part of Poo Collector</td>
<td>Poo Collector “Collects” Mr. Hanky thus removing him/them from Kenny’s Hand.</td>
</tr>
<tr>
<td>Mecha-Streisand</td>
<td>Any part of Mecha-Streisand colliding with any part of Mr. Hanky</td>
<td>Poo Collector does nothing. Mecha-Streisand bounces.</td>
</tr>
</tbody>
</table>
Mecha-Stresiand:

Evil Mecha-Streisand. Flying around terrorizing South Park.

Movement Control:
None. Mecha-Streisand flies around randomly in Kenny's world.

Interaction:

<table>
<thead>
<tr>
<th>Character</th>
<th>Interaction</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kenny: body &amp; right arm</td>
<td>Any part of Mecha-Streisand colliding with Kenny's body (all except his mutant left arm)</td>
<td>Kenny dies.</td>
</tr>
<tr>
<td>Kenny: mutant left arm</td>
<td>Any part of Mecha-Streisand colliding with Kenny's left arm.</td>
<td>Mecha-Streisand dies.</td>
</tr>
<tr>
<td>Mr. Hanky</td>
<td>Any part of Mecha-Streisand colliding with any part of Mr. Hanky</td>
<td>Mecha-Streisand bounces off flying in another direction</td>
</tr>
<tr>
<td>Poo Collector</td>
<td>Any part of Mecha-Streisand colliding with any part of Poo Collector</td>
<td>Mecha-Streisand bounces off flying in another direction</td>
</tr>
</tbody>
</table>

World Constraints:
Kenny's world is confined by the dimensions of the World View located on the right side of the game window. Kenny, nor any of the characters cannot exit the scene.

The white box represents a “zoom box”. This zoom box moves along with Kenny illustrating Kenny’s position in his world.
Using Zoom Box:

<table>
<thead>
<tr>
<th>Mouse Click</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mouse Click</td>
<td>Action</td>
</tr>
<tr>
<td>Middle Button Down</td>
<td>Centers Zoom Box</td>
</tr>
<tr>
<td>Middle Button Down Drag: RIGHT</td>
<td>Zoom OUT from center point</td>
</tr>
<tr>
<td>Middle Button Down Drag: LEFT</td>
<td>Zoom IN from center point</td>
</tr>
<tr>
<td>Right Button Down Drag</td>
<td>Relocates Zoom Box</td>
</tr>
</tbody>
</table>
System Design

A complete breakdown and discussion of every class in this project is not yet available. What has been provided are some highlights of the key classes, data structures and events that make up Kenny Time!

Important Classes:

Character Classes:

\[ \text{T<charactername>} \]

- TKenny: Kenny Object
- TMechaStreisand: Mecha-Streisand Object
- TMrHanky: Mr. Hanky Object
- TPooCollector: Poo Collector Object

These classes cover the following for each character:
- Character Representation
- Character Special movement
  - Any movement other than relocation in the game world (e.g. Kenny’s arm)
- Character Type
  - This is a SpriteType that identifies the type of character
    - HANKY
    - MECHA
    - KENNY

TGraphicsObject:
Each Character class inherits from TGraphicsObject. This class addresses the Translation, Rotation and Scale of each object as well as the various attributes. Added to the class is a set of methods used for Movement.

Variables:
- fDirection

Each Graphics object now has a direction. In the case of Kenny, a direction is set each time he is moved. Since Kenny is completely controlled by the user, this attribute is rarely used. In the case of Mecha-Streisand, this allows the freedom of “Bounce”. The concept of bouncing in this rudimentary game is basically defined as:

Set Object’s direction to the opposite of current direction. If object’s direction is RIGHT, set direction to LEFT. If object’s direction is UpLeft, set direction to DownRight and so on …
Methods:
setRandomDirection()
   This sets a GraphicsObject’s direction to some random
direction.
Bounce(Direction d)
   Given Direction d, set the GraphicObject’s direction to the
   “opposite” (see fDirection for defininition)
Bounce()
   Set the Object’s direction to the opposite of current direction.
Move(Direction d)
   Take GraphicObject’s current location and move the
   appropriate direction by MOVE_INTERVAL.
   MOVE_INTERVAL for this version is 3 units.
MoveRight(), MoveLeft(), MoveUp() ... MoveDownLeft()
   Move the character by MOVE_INTERVAL (3) units.

Direction
Direction is an enum that represents the direction an object is heading.
The following Directions are described:
   | RIGHT | UP   | UpRight | UpLeft |
   | LEFT  | DOWN | DownRight| DownLeft|

Important Data Structures
The most important data structures in Kenny Time! exist in TmyGame.

TmyGame
This class has separate lists for a collection of characters.
   fAllObjects
      Granted, a poor choice for a variable name, however, this is
      actually used for generic purposes. This list is always drawn.
      Upon the games end, a new instance of the winning
      character is inserted into this list to be drawn. A RESET will
      delete this list. This list comes in handy for this temporary
      reason. A few lines of code will pop an object in there to be
      drawn and is easy to remove.
   fMrHankyCollection
      This holds the Mr. Hanky collection. Each hanky is placed
      on the board, if there is a collision, it is removed and another
      attempt made. To avoid objects landing in the same spot,
      any object placed will go through the same routine. Mr.
      Hanky was granted its own data structure to ease any future
      modification of the game. For example, with a few lines, you
      can move the entire Mr. Hanky collection to make it more
difficult for the Kenny object to catch one.
fStationaryObjects
This list holds the Poo Collector right now. This list was created to hold all things stationary. If an object isn’t intending to move in the game, now or in future versions, this is where it should be placed.

fHeroObject
This is Kenny. Kenny is drawn along with all other lists. There is no intention of having more than one Kenny in *Kenny Time!*, therefore, having one pointer to our hero was sufficient. No list structure was needed.

fcurMehcaStreisand
This is the active Mecha-Streisand. One could foresee a dead Mecha-Streisand cease moving and still be drawn while a moving Mecha-Streisand was roaming around. This pointer would point to the current moving Mecha-Streisand. Version 1.0 of *Kenny Time!* did not address this issue. The variable name still exists and points to the current moving Mecha-Streisand … this is the *only* Mecha-Streisand on the board and any given time.

TKenny

fMrHanky
Holds Kenny’s collection of Mr. Hanky.

fHankyHeight
Indicates at what Y the new Mr. Hanky should be added. In this game, they are stacked on top of each other. When a Mr. Hanky is added to the list, this value in incremented by one. When Kenny’s Hanky list is removed (e.g. a collision with a Poo Collector) the height is set back to the starting location for Mr. Hanky (HANKY_Y)

TKenny has a special data structure that holds his collection of Mr. Hanky. As Kenny collides with Mr. Hanky on the game board, this list grows. If he collides with a Poo Collector, this list is erased.
Basic Events

Kenny Collects Mr. Hanky.  
Kenny deposits Mr. Hanky.

Mecha-Streisand kills Kenny.
Kenny kills Mecha-Streisand with mutant left arm.

Limitations:
As great and charming as *Kenny Time!* is, it’s not without it’s limitations. One limitation of *Kenny Time!* that definitely needs to be addressed for any future versions is **Performance**. Currently there are 10 Mr. Hanky placed on the game board. While 10 is a nice round number, this decision was not arbitrary. Any more than ten and the game slows to a crawl. The many lists being traversed to check for collisions and just general drawing routines slow down the game. A cool side effect of this behavior lies in the removal of Mr. Hanky. As Kenny deposits more and more Mr. Hanky into the Poo Collector, Mecha-Streisand gets faster and faster. This gives the game that little edge to keep you entertained.

For the next version:
The following list suggests a few potential improvements to *Kenny Time!* By no means is this an exhaustive list. It merely points out some key issues to make the game more marketable.

- Switch to a left handed keyboard control.
  - *Kenny Time* currently accommodates the right-handed individual. The next release should have a ‘one-button’ option to switch the controls to a left-handed version.
- Multiple Mecha-Streisand characters
Presenting Kenny with more than one evil object to deal with would give the game more of a challenge to play.

ISSUE: a prototype was made during the development of the this project and revealed that game play “slowed down” quite a bit as more Mecha-Streisand characters were added. To accomplish this feature, some performance issues with the code base will need to be addressed.

- Introduce a level system
  o Version 1.0 implements only one level of *Kenny Time!* It is highly likely that in the future versions of this project, introducing a level system would prove to be more marketable.
  o After Kenny kills Mecha-Streisand 3 times, Kenny will enter a new level with different goals and evil characters to deal with.

- Showing Score
  o Though this seemed an easy task, it would benefit the user to actually view how many times he/she has killed Mecha-Streisand. This feature got cut due to time constraints. We felt shipping on time was more important than a visual representation of the score.

- Have Evil objects “chase” Kenny
  o In version 1.0, the evil characters (Mecha-Streisand in level 1) randomly fly around the world. Have these objects hone in on Kenny base on his location, would add that extra challenge to the game. As it is, random is difficult in it’s own right, but there are more giggles and fun to be had when the evil character is hunting you.

<table>
<thead>
<tr>
<th>Date</th>
<th>Editor</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/11/2001</td>
<td><a href="mailto:jeanc@westside.com">jeanc@westside.com</a></td>
<td>Initial Documentation</td>
</tr>
</tbody>
</table>