**Caged In**

*team-error-undefined*

*Ashton, Kevin, Miguel*

**Table of Contents:**

Game:

QWO Game v1

Source Packages

version5

**MainCagedIn.java** (run this)

FraMain.Java

PanDisplay.Java

PanGame.Java

PanInGameOptions.Java

Main.Java

MainMenu.Java

PanStats.Java

Yeti.Java

Scratch1:

QWO Game v1

Source Packages

SoundScratch

**MainGame.java** (run this)

FraMain.Java

PanGame.Java

PanMain.Java

PanMainMenu.Java

PanStats.Java

Yeti.Java

Scratch2:

QWO Game v1

Source Packages

OptionsPanelScratch

**MainCagedIn.java** (run this)

FraMain.Java

PanDisplay.Java

PanGame.Java

PanInGameOptions.Java

Main.Java

MainMenu.Java

PanStats.Java

Yeti..Java

LoadImages

OptionsPanelScratch2

**MainCagedIn**.java (run this)

FraMain.Java

PanDisplay.Java

PanGame.Java

PanInGameOptions.Java

Main.Java

MainMenu.Java

PanStats.Java

Yeti..Java

LoadImages.Java

Scratch3:

QWO Game v1

Source Packages

CountdownScratch

FraMain.java

**Main.java** (run this)

PanUI.java

2.**Starting Release Schedule**

|  |  |
| --- | --- |
| **Release Name** | **New incremental features of this release** |
| **0.01** | **load background** |
| **0.21** | **Load image of character** |
| **0.50** | **animation** |
| **0.67** | **character movement (horizontal)** |
| **0.69** | **character movement (vertical)** |
| **0.70** | **Add main menu with start and exit button** |
| **0.72** | **load test platform** |
| **0.91** | **add collision detection** |
| **1.0** | **begin level design** |
| **1.21** | **Add pits of death** |
| **1.89** | **add death** |
| **1.95** | **add attack function** |
| **2.0** | **add attack animation** |
| **2.05** | **add stats screen** |
| **2.1** | **add save feature** |
| **2.7** | **add lives** |
| **2.92** | **power ups** |
| **3.0** | **add enemies** |
| **3.25** | **add enemy deaths** |
| **3.5** | **add coins that contribute to score** |
| **4.0** | **Co-op** |

|  |  |
| --- | --- |
| **Release Name** | **New incremental features of this release**  **Bold is not complete** |
| **0.01** | load background |
| **0.21** | Load image of character |
| **0.50** | animation |
| **0.67** | character movement (horizontal) |
| **0.69** | character movement (vertical) |
| **0.70** | Add main menu with start and exit button |
| **0.72** | begin level design + implementation |
| **0.82** | background music |
| **0.91** | **Add pits of death** |
| **1.0** | **add save feature** |
| **1.21** | **add death** |
| **1.89** | **add Leaderboards** |
| **1.95** | add lives |
| **2.0** | **load test platform (for character to jump on)** |
| **2.05** | add collision detection |
| **2.07** | attack sound |
| **2.1** | **add attack function** |
| **2.7** | **add attack animation** |
| **3.0** | **add enemies (nicholas cages)** |
| **3.25** | **add enemy deaths** |
| **3.5** | **power ups** |
| **4.0** | **add coins that contribute to score** |
| **5.0** | **Co-op** |

**Program overview/introduction:**

You are the yeti, the poor unfortunate soul stuck in Nicholas Cages basement. This game documents your escape from the hollywood superstars dungeon. Good luck.

Our game is a combination of the classic mario games and the popular app jetpack joyride. The goal of the game when it is finished is to reach the end of the map while passing through narrow paths, dodging enemies and trying not to die. if you lose all of you lives the game will close and you will have to start again.

**Scratches:**

1. **Sound scratch:** 
   * This is the scratch where we learned how to implement sound.
2. **Options scratch:**
   * Our attempt at an options panel to control features of the game.
3. **Countdown logic scratch**:
   * This scratch was where we constructed the timer to countdown with images.

**Disclaimer/ list of known bugs:**

* The character should not be able to move outside of the screen, however by pressing 2 keys at a time the barrier can be bypassed.
* The game will only end once the yeti dies.
* Journals are in a separate file named “Journals.txt”

**Notes to future programmers:**

* Do not believe everything you read. Even if a source says that their code should work perfectly, it may be more broken than a game by Bethesda.

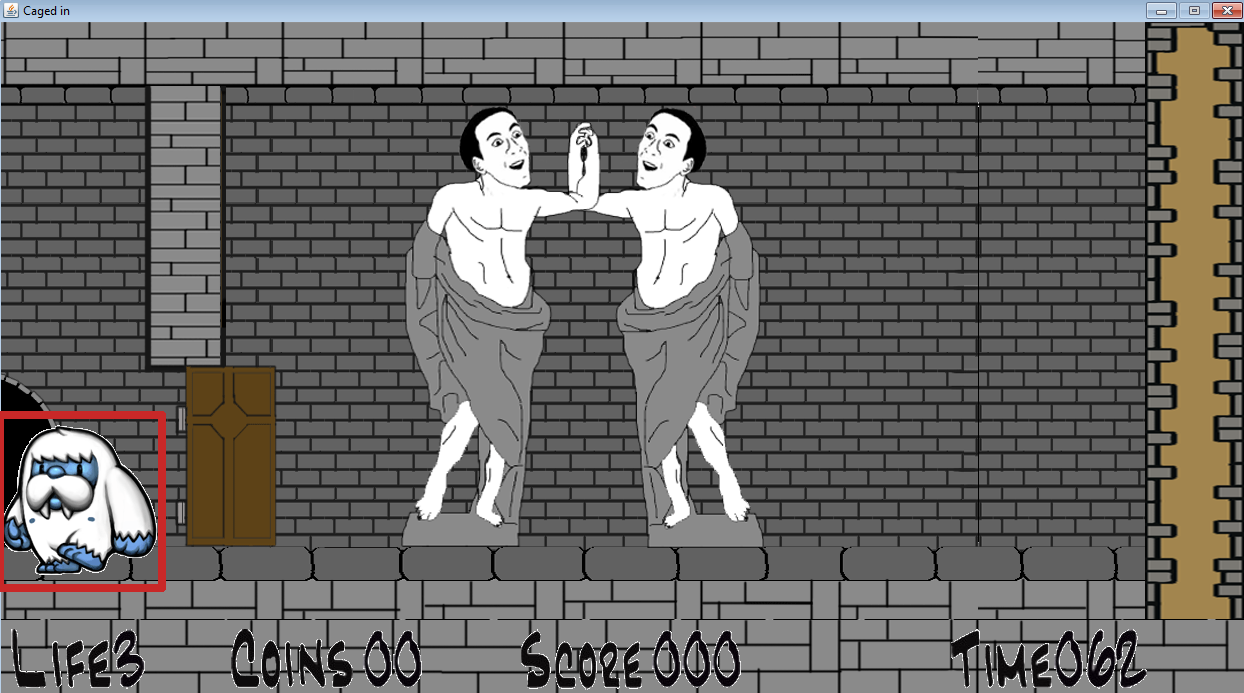
**Caged in - User Manual:**

1. Upon opening the game located: quogame/version5/CagedIn.java the PanMainMenu will appear. In this menu, you will be prompted with buttons to: Start Game or End Game.



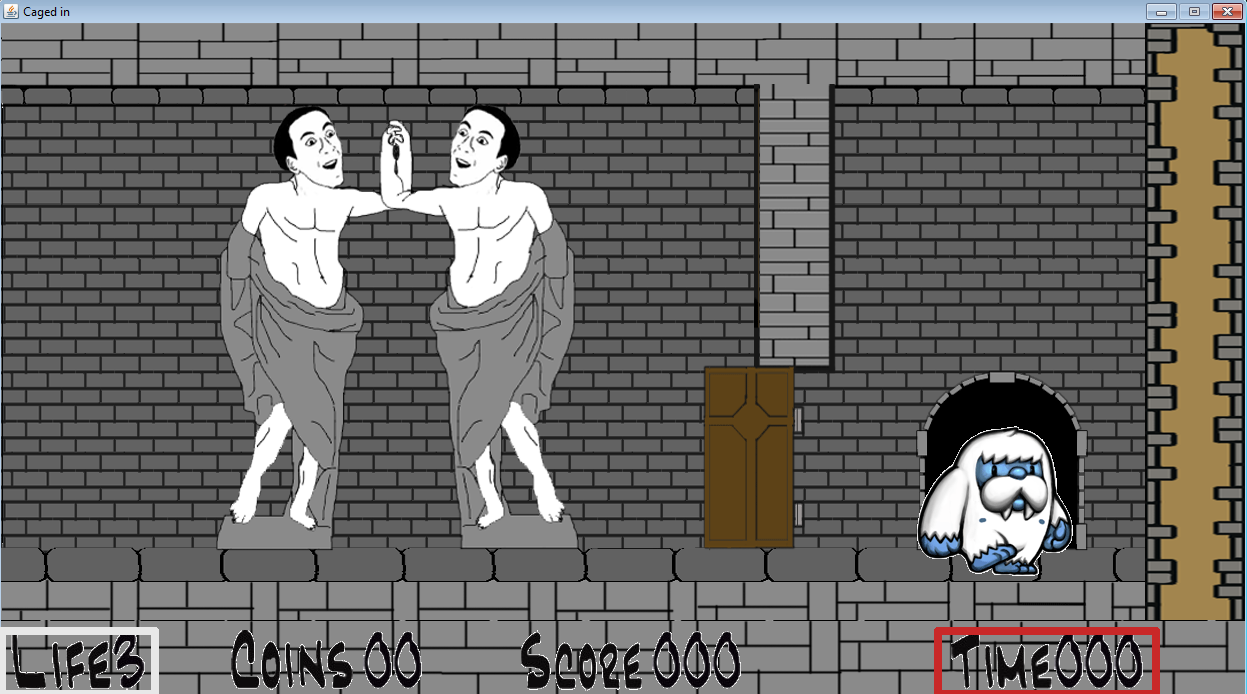
* 1. If you wish to end the game you may wish to do so now by pressing the ‘End Game’ button as indicated with the black arrow in the photo. In addition to this you can also click the red ‘X’ button at the top left side of the screen.+
  2. To start the game press the ‘Start Game’ button as indicated with the red arrow.

1. This is Your Character:



* 1. The Background will move with your character.
  2. To Jump press ‘W’, walk right with ‘D’ , walk left with ‘A’, and while jumping, press ‘S’ to move to the ground

1. This is the End:



* 1. Make it to the end before the timer gets to ‘000’ or else you will lose your life
  2. If you lose all of your lives it is game over… so do your best!

8.

sources:

sound:

Don Vo

animation:

Chris Imperatore

Shane Ploj

9. Notes to future programmers

Do not believe everything you read. Even if a source says that their code should work perfectly, it may be more broken than a game by Bethesda.

**Lessons:**

Ashton - My favourite “technique” was drawing backgrounds based on a variable and the additive widths of past rooms.

public void paint(Graphics g) {

super.paint(g);

Graphics2D g2d = (Graphics2D) g;

try {

nBgXEND -= 10;

if (nBgXEND >= 0) {

nBgX -= 10;

}

////Room 1

if (nNumRooms >= 1) {

g2d.drawImage(iRooms[nLevel[0]], nBgX, 0, null);

}

////Room 2

if (nNumRooms >= 2) {

g2d.drawImage(iRooms[nLevel[1]], nBgX + iRooms[nLevel[0]].getWidth(this), 0, null);

}

////Room 3

if (nNumRooms >= 3) {

g2d.drawImage(iRooms[nLevel[2]], nBgX + (iRooms[nLevel[0]].getWidth(this)) \* 2, 0, null);

}

...ECT.

} catch (Exception e) {

e.printStackTrace();

}

Where nBgXEND acts as a count down so that the background stops moving, and nBGX is the variable that controls the location of the background

**Miguel & Kevin - Sound**

**these are the imports we used**

import java.applet.Applet;

import java.applet.AudioClip;

…

**this is where we declare the needed variables**

File AttackSound1 = new File("AttackSound1.wav");

AudioClip Attack1;

…

**this is where “Attack1” becomes active**

try {

Attack1 = Applet.newAudioClip(AttackSound1.toURL());

} catch (Exception e) {

}

…

**this is the action listener we used to play the sound**

public void keyPressed2(KeyEvent k2) {

int code2 = k2.getKeyCode();

if (code2 == KeyEvent.VK\_Z && canAttack == true) {

System.out.println("attacking");

Attack1.play();

canAttack = false;

}

}