

# COMP4021

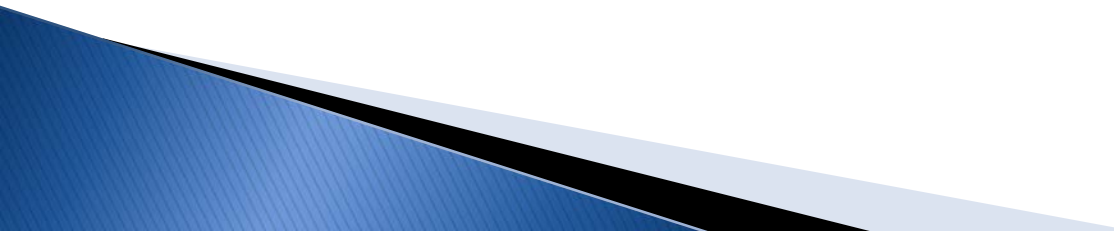
## Internet Computing

### Assignment 2

### *Dynamic SVG Game*

*Handout date: July 1, 2017 (Sat)*

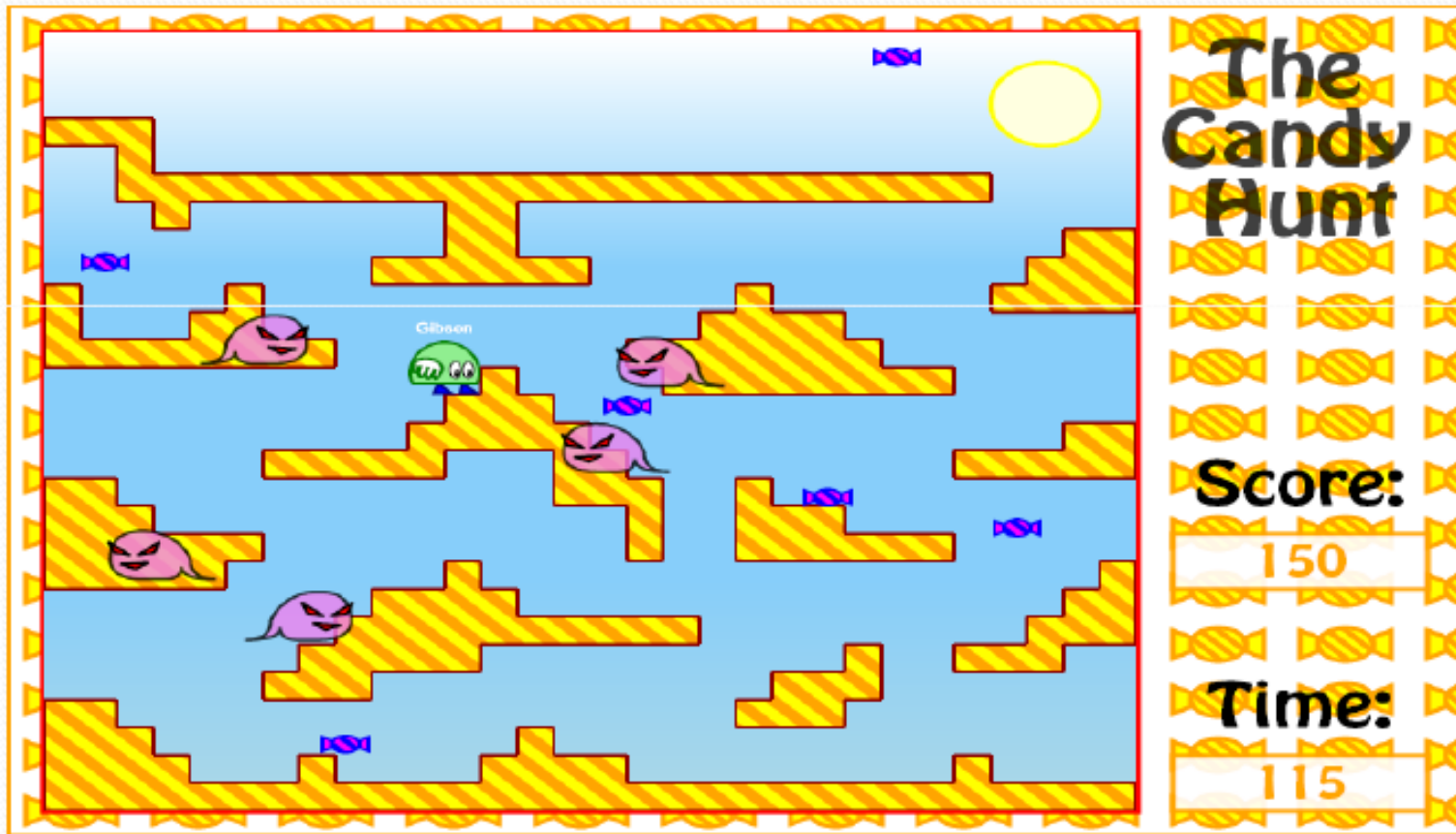
# Basic Idea of the Game 1 / 2

- ▶ After the game starts, the 'time left' will start from an appropriate number, such as 100 seconds
  - ▶ The 'time left' will be reduced by 1 every second
  - ▶ The player needs to reach a particular exit point (i.e. an exit) before the time becomes zero
  - ▶ If the time runs out before the player reaches the exit point, the player dies
  - ▶ If the player reaches the exit before the time runs out, the time remaining is added to the player's score
- 

# Basic Idea of the Game 2 / 2

- ▶ Platforms
  - To get to the exit point, the player has to walk/jump on several platforms
- ▶ Good things
  - There are good things near the platforms; when the player gets one he/she increases the score
  - The player has to collect all the good things before he/she can finish the level i.e. get them before going to the exit
- ▶ Monsters
  - Monsters appear in random places; the player dies if it touches one
  - The player can shoot the monsters to get more score

# Example Screen Shot



# Example of Commercial Games



Chack'n Pop, from 1983



Manic miner, from the 1980's

- You can get ideas for your platform game arrangement by looking at other platform games available on the Internet

# Two Ways to Play

- ▶ As soon as the page loads, the player gets two choices:
  1. Play in normal 'no zoom' mode
    - The entire game is played in 'normal' mode (no zoom)
  2. Play in 'zoom' mode (200% zoom)
    - The entire game is played in 'zoom' mode
    - The whole game works exactly the same as for no zoom mode, with the same platforms, monsters, etc, but because of the zoom everything is much harder (i.e. can't see the monsters until they are very close)
    - Therefore, in 'zoom' mode all scores will be greater compared to the 'no zoom' mode, as it is much harder
- ▶ The whole game is played in the selected mode until the player dies (i.e. touches a monster or runs out of time)

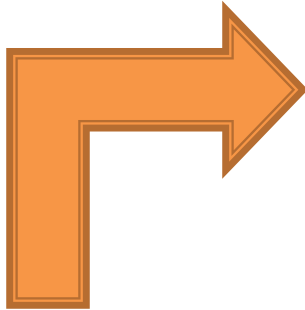
# Example Start-up Screen



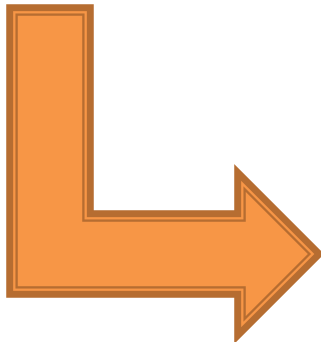
*Include your Name and last four digits of your student (e.g. xxxx1234) when in the start-up screen*



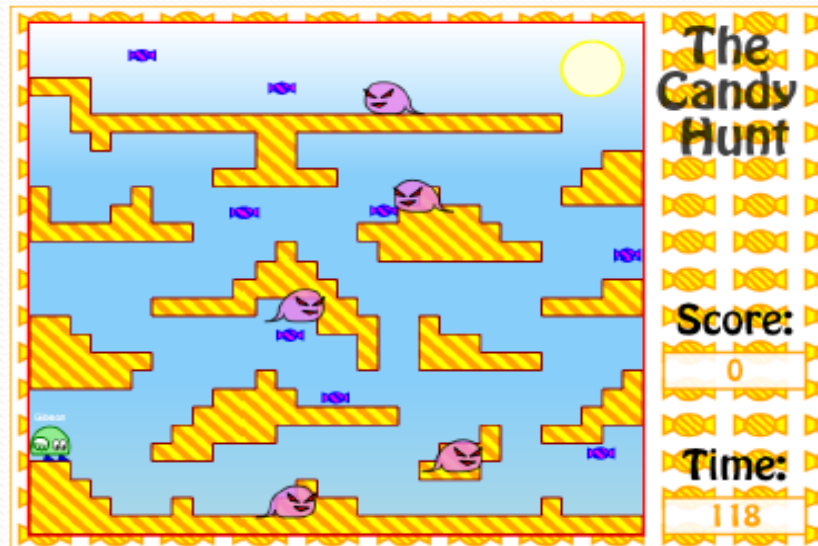
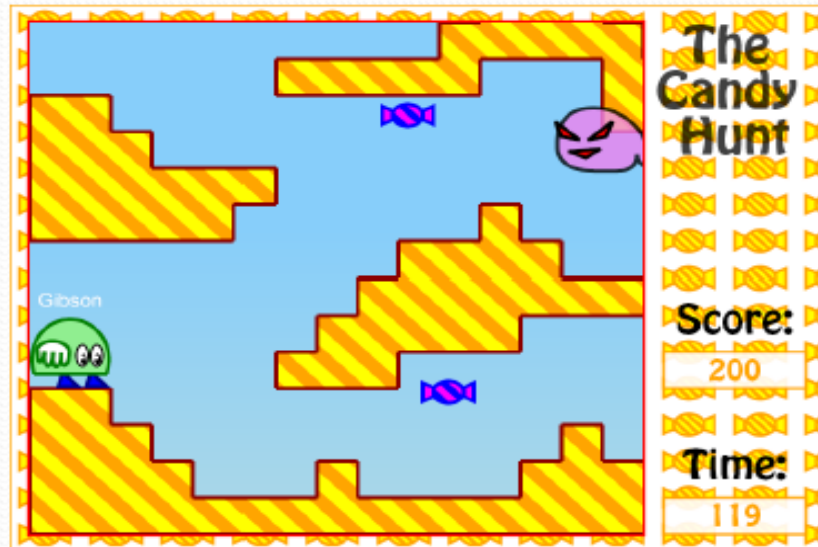
- Zoom mode (200% zoom)



- After user chooses which mode to play the game in



- Normal mode





# Zoom Mode = Relatively More Score

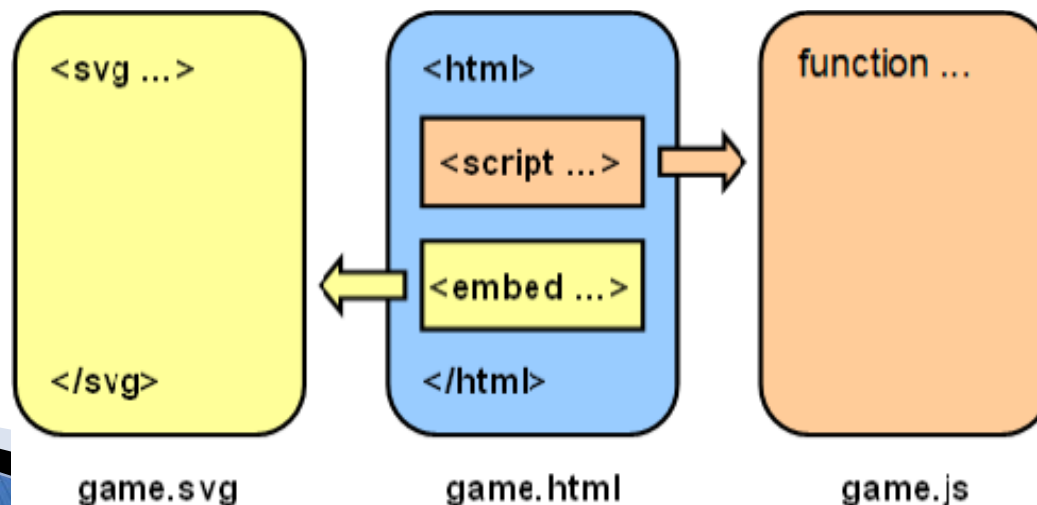
- ▶ In zoom mode all scores are worth significantly more than in normal mode i.e.
  - shoot a monster, get perhaps twice as much as you would in normal mode
  - finish a level, get perhaps twice as much as you would in normal mode

# Summary of Keys

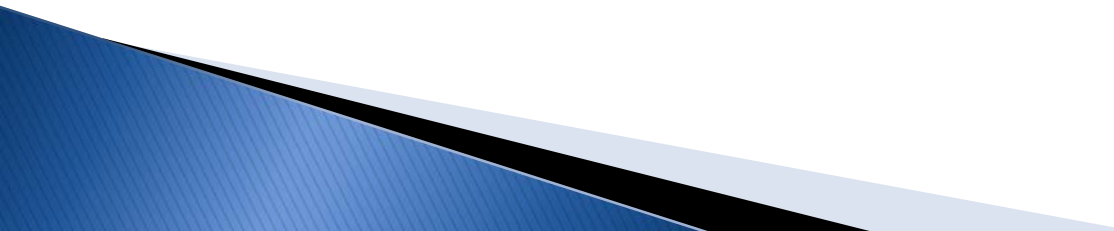
- ▶ Please use these keys in your game:
  - w – jump
  - a – left
  - d – right
  - space – shoot
  - c – cheat mode on/off

# File Arrangement

- ▶ Use the same file arrangement from the labs arrangement from the labs
- ▶ To start the game, open game.html in Firefox 26.0 (we will mark the assignment using exactly the same environment in lab, i.e. the Portable Firefox in L:\apps\COMP4021\ )
- ▶ There is no html display in this project, all display is SVG (you can use prompt() for name input)



# Overview (1 / 2)

- ▶ Game Engine
  - ▶ Starting Screen 4%
  - ▶ Handling of Player 8%
  - ▶ Handling of Monsters 10%
  - ▶ Handling of Good Things 5%
  - ▶ Platforms 4%
  - ▶ Transmission Portal 4%
  - ▶ Shooting 6%
- 

# Overview (2 / 2)

▶ Sound	5%
▶ Time Remaining	8%
▶ Level Handling	8%
▶ Game Quality	8%
▶ Score Update and Display	4%
▶ Cheat Mode	10%
▶ End of Game	8%
▶ Handling High Score Table	8%

Maximum Mark = 100%

# Game Engine/ Basic Requirements

- ▶ Game engine/ basic requirements

- No logical/ procedural errors

- Appropriate collision detection

- Correct jumping/walking behavior

- Deductions for any use of bitmap images in the game

- Zoom mode implemented correctly compared to normal mode

- Everything stored in the DOM (except cookie information)

- ...and so on

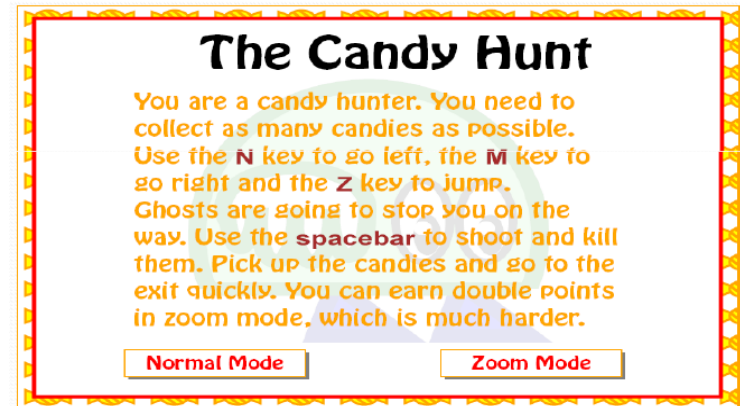
- ▶ Marks will be deducted for any problems encountered

# Size of the Game

- ▶ The game area (where the game is taking place) has a size of 800 pixels by 600 pixels at least
- ▶ This is the same as the game we have in the lab
- ▶ If you want to change the size from the lab code (optional) you need to:
  - Adjust the size in the svg element at the top of the svg file
  - Adjust the size of the game area group in the svg file
  - Adjust the value stored in the constant `SCREEN_WIDTH` and `SCREEN_HEIGHT` in the JavaScript file

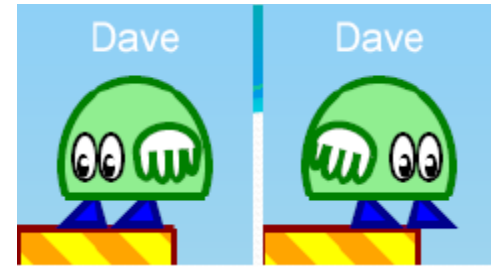


# Starting Screen 4%



- ▶ When the SVG starts you need to give the player some information
  - Include the **title of the game, your name, and student ID**
  - Give a general introduction to the game and tell the player what he/she needs to do
  - Say what keys the user needs to press to play the game (left/right/jump/shoot)
  - This is all shown in the same 'screen', in SVG
  - On this screen the user can choose either 'normal' or 'zoom' mode to start the game
  - You can add anything else appropriate

# Handling of Player 8%



- ▶ +2 marks – ‘Flip’ player when move left/ right
- ▶ +2 marks – The player can jump/ move left/ move right/ shoot on any platform
- ▶ +2 marks – The player name is appropriately shown at the top of the player (as shown above) , with ‘Anonymous’ used as the name if the user enters an empty string
- ▶ +2 marks – The player dies if it touches any monster

# Handling of Monsters 10%



- ▶ There must be at least 6 monsters
- ▶ The monsters can all look the same, if you want
  - +2 marks – Some appropriate animation of monsters (using any SVG animation command(s) except animateColor )
  - +2 marks – The monsters appear at random places at the start of the game but must not be very close to the player
  - +2 marks – The monsters move smoothly from one random location to another random location during the game
  - +2 marks – ‘Flip’ monster when move left/ right
  - +2 marks – There has exactly one special monster which can shoot bullet. There has at most one bullet in game window from the monster at a time.

# Handling of Good Things 5%



- ▶ There must be at least 8 good things in the game
  - +1 marks – The good things are generated at random places at the start of the game
  - +1 marks – The good things cannot appear within a platform, i.e. they should not overlap with any platforms
  - +1 marks – The player collects the good things by touching them. The collected good things are deleted from the DOM
  - +2 marks – The player needs to collect all good things before he/she can go to the next level

# Disappearing Platforms 4%

- ▶ +2 marks – There are three disappearing platforms. If the player stays on the disappearing platform after a certain period of time (i.e. 0.5 second), the disappearing platform will disappear and the player will fall down
- ▶ +2 marks – Good visual impact showing the disappearing platform is going to disappear (i.e. changing the opacity or the color)

After the platform has disappeared, it does not come back again.



When standing on a sliding platform, the platform gradually disappears, then the player falls down.

# Transmission Portal 4%



- ▶ +2 marks – There should have two portals appeared on the screen (shape and location is freely defined).
- ▶ +2 marks – When the player enters into one portal, it will teleport to the position of another portal.

# Shooting 6%



- ▶ +2 marks – A player gets 8 bullets at the start of the game for each level and the number of remaining bullets is appropriately shown and updated in the GUI
- ▶ +2 marks – When facing left, the player shoots to the left (bullet is removed from DOM appropriately when it is off the screen on the left)
- ▶ +2 marks – When facing right, the player shoots to the right (bullet is removed from DOM appropriately when it is off the screen on the right)



# Sound 5%

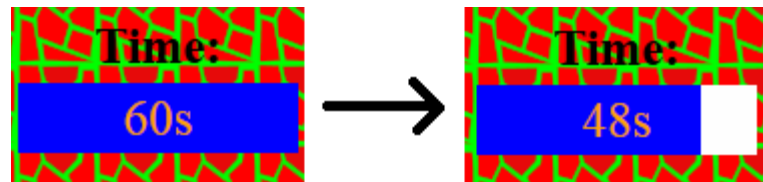
## ► Use of sound

- +1 mark – Appropriate sound when the player shoots
- +1 mark – Appropriate sound when the player reaches the exit point
- +1 mark – Appropriate sound when the player dies (touches monster or runs out of time)
- +1 mark – Appropriate sound when a monster dies (is shot by the player)
- +1 mark – Appropriate continuous music during the game

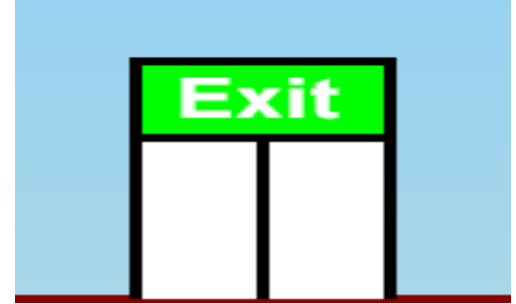
# Time Remaining 8%



- ▶ The player needs to reach the exit point within a certain period of time, e.g. 60 seconds
- ▶ The player will die if the player cannot reach the exit point within that time
- ▶ The time is count down in two ways: (1) by number and (2) by time bar. The time bar is just a rectangle which keep on reduce its width with the ratio according to the remaining time and the width becomes zero when the remaining time becomes zero
- ▶ +4 marks - Time count down number is updated and displayed appropriately every second (perhaps using a setInterval() )
- ▶ +4 marks – Time bar is updated and displayed appropriately in every second or even faster/smoothen (perhaps using a setInterval() or svg animations)



# Level Handling 8%



- ▶ The game starts at with a fixed initial time duration, e.g. 60 seconds.
- ▶ The player should collect all good things before the player can reach the exit point.
- ▶ When the player reaches the exit point, the score from the remaining time is added **and the High Score Table will then be displayed**
  - +1 mark – Appropriate visual impact of the exit
  - +2 marks – **The High Score Table will then be displayed when the player reaches the exit point**
  - +2 marks – Correct time duration
  - +3 marks – The game is restarted after the display of the High Score Table


# Game Quality 8%

- ▶ How playable the game is
  - +0/4/8 marks – generally poor/ok/good game
    - ✓ To get any marks in this section, your game must use a different theme/images compared to the theme/images given in the labs
    - ✓ You should not make your game too difficult, e.g., the monsters should be roughly uniformly distributed in the game area, and the platforms should be created in some "reasonable" ways (not too narrow/high for the player to reach)

# Score Update and Display 4%

- ▶ Score is updated at the end of the game. Add  $X$  points for each second of remaining time, where you choose an appropriate value of  $X$ .
- ▶ Score is updated when a monster is shot – add  $Y$  points when this happens, you choose an appropriate value of  $Y$
- ▶ Score is updated when a good thing is touched – add  $Z$  points when this happens, you choose the value of  $Z$
- ▶ If the game is played in zoom mode, the marks are significantly more than those mentioned above. We double  $X$  and  $Z$  and triple  $Y$ .
- ▶ +1 mark for each of the above

# Cheat Mode 10%

- +3 mark – In cheat mode, everything is the same as usual, but player will not die when colliding with a monster. And also the player will have infinite bullets in cheat mode (everything else works the same as usual)
- +1 marks – Set the opacity of the player to 50% → 
- +2 mark – user can press 'c' to enter cheat mode, which works appropriately. If user presses 'c' in the cheat mode, it will still keep the player in the cheat mode
- +2 mark – user can press 'v' to leave cheat mode, which works appropriately. If user presses 'v' outside the cheat mode, then nothing happens
- +2 mark – In cheat mode, the player can always jump no matter whether the player is on a platform.
  - ▶ The player can turn on and off cheat mode whenever he/she wants to (i.e. this feature is useful for debugging your game while you build it)

# End of Game 8%

- ▶ If the player cannot reach the exit point during the required period of time, or touches a monster, or touches the bullets shot by the special monster, the player will die
  - +2 marks – Score+name are inserted into a top 10 high score table at correct place, if score is high enough
  - +2 marks – High scores saved/ updated appropriately in the cookie(s)
  - +2 marks – Show players score and high score, see next slide.
  - +2 mark – Show a ‘Start again?’ button, if the player clicks on it the game begins again, and the user is asked for his/her name as usual, with the previously entered name used as the default text in the window (i.e. using prompt())



# Handling High Score Display 8%

- ▶ +1 mark – Appropriate title is shown above the high score table i.e. ‘High Score Table’
- ▶ +4 mark – Top 10 scores are shown in descending order, with highest at the top, lowest at the bottom, this will include the player’s score if it was high enough. If the current player is within the top 10 scores, mark him/her with a different color
- ▶ +3 marks – cookies are used appropriately to store everything

Example high score display from a commercial game



ELITE RAPTORX PILOTS	
DredPirateRob	00200000
Rod	00100000
MountainKing	00080000
Marcuz	00070000
SWeroluk	00060000
ForotHead	00050000
Mgndth9	00040000
SilverNinja	00030000
GreatYak	00020000
BeerMan	00010000

# Example High Score Table

The image shows a 'Game Over' screen for a game titled 'The Candy Hunt'. The screen has a pink background with a yellow border decorated with candy icons. At the top, it says 'Game Over' in large white letters. Below this is a 'High Score' table with two columns: scores and player names. To the right of the table, there are fields for 'Score:' and 'Time:'. The 'Score:' field shows '0' and the 'Time:' field shows '114'. At the bottom center, there is a 'Start Again' button. The background also features a small pink cloud, a pink candy, and a pink circle.

**Game Over**

**High Score**

25800	Superman
10700	Batman
9200	X Man
6550	X Man
6550	X Man
1200	X Man
800	Superman
700	Rockman
600	Superman
450	X Man

**Start Again**

**The Candy Hunt**

**Score:**  
0

**Time:**  
114

# HTML5 Canvas

- ▶ You are not required to use HTML5 Canvas to finish this assignment but you are welcome to use it if you want to
- ▶ We will not provide any materials about HTML5 Canvas because this is required
- ▶ You can get some tutorials about HTML5 Canvas from the following websites
  - [https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Canvas\\_tutorial](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Canvas_tutorial)
  - [http://www.w3schools.com/html/html5\\_canvas.asp](http://www.w3schools.com/html/html5_canvas.asp)
  - <http://www.sitepoint.com/html5-canvas-tutorial-introduction/>

# Submission

- ▶ Your submission should work fine with the Portable Firefox used in the labs (L:\apps\COMP4021\)
  - ▶ You need to submit all the used files and sounds
  - ▶ Please generate a single file using ZIP
  - ▶ If you want to write any message to the marker, write them in a file called "readme.txt"
  - ▶ The deadline is **July 9 (Sun) 23:59:59**
- 