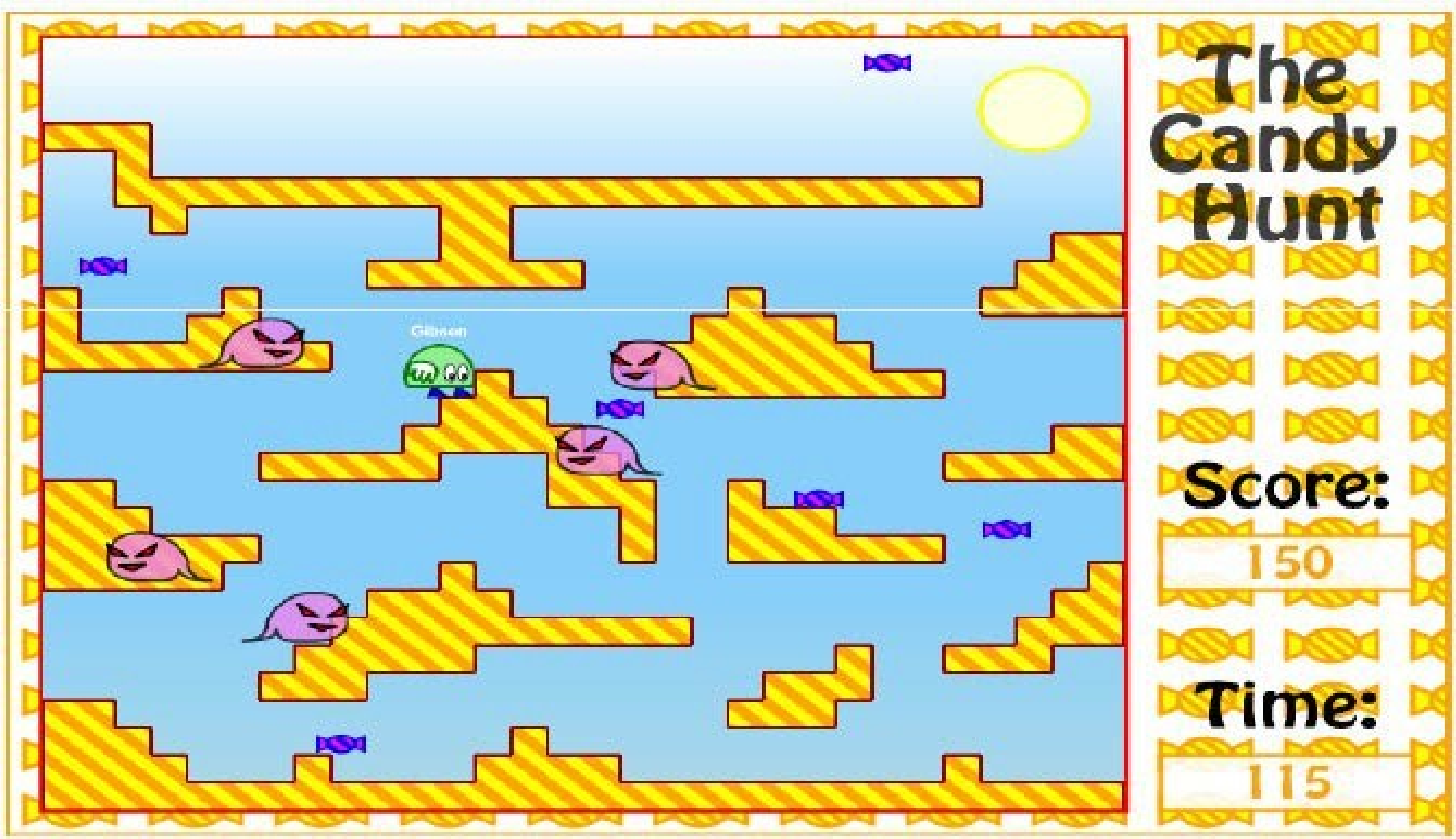
# COMP4021
# Internet Computing

Assignment on *Dynamic SVG Game*

# Example Screen Shot

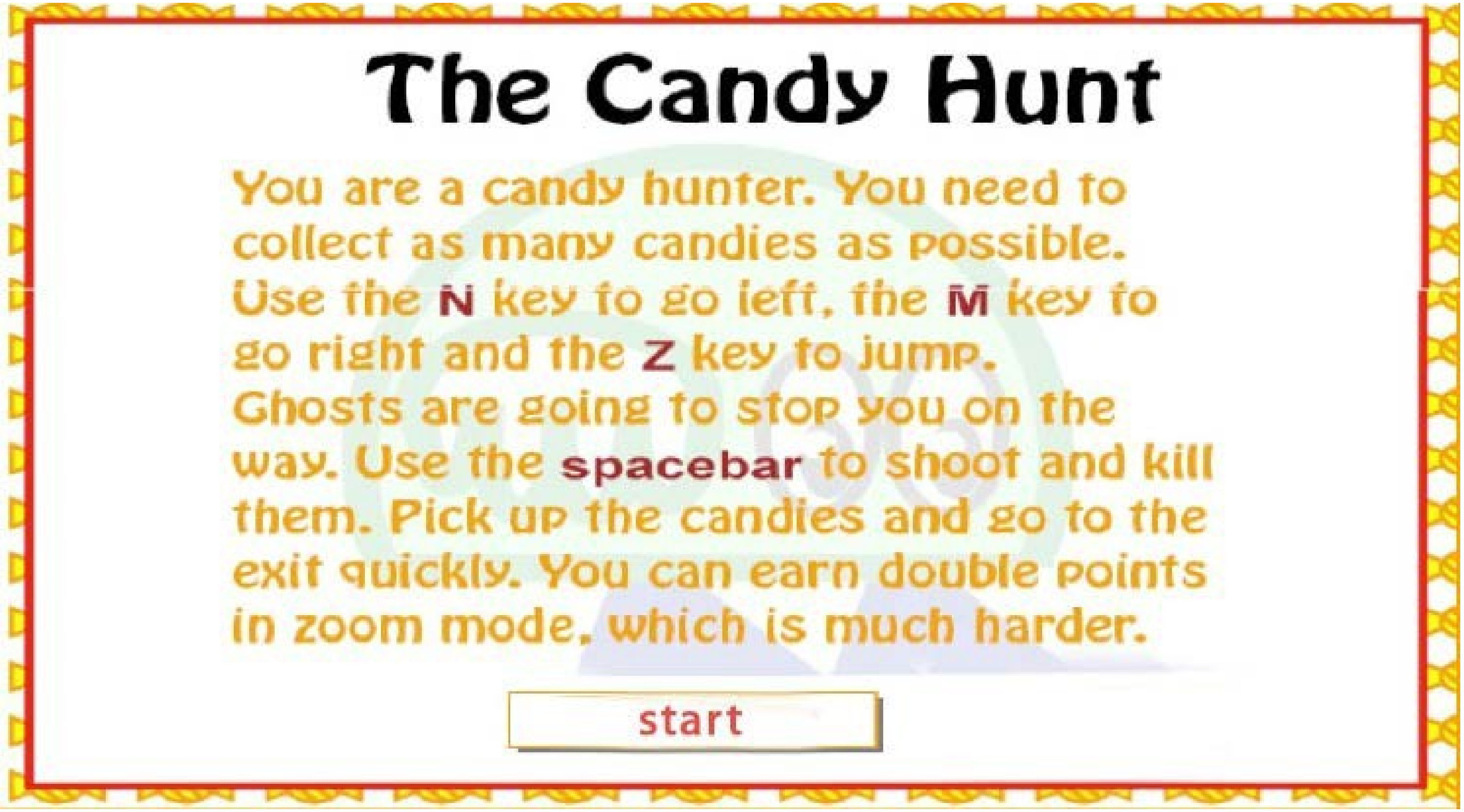


The Candy Hunt

Score:
150

Time:
115

# Example Start-up Screen

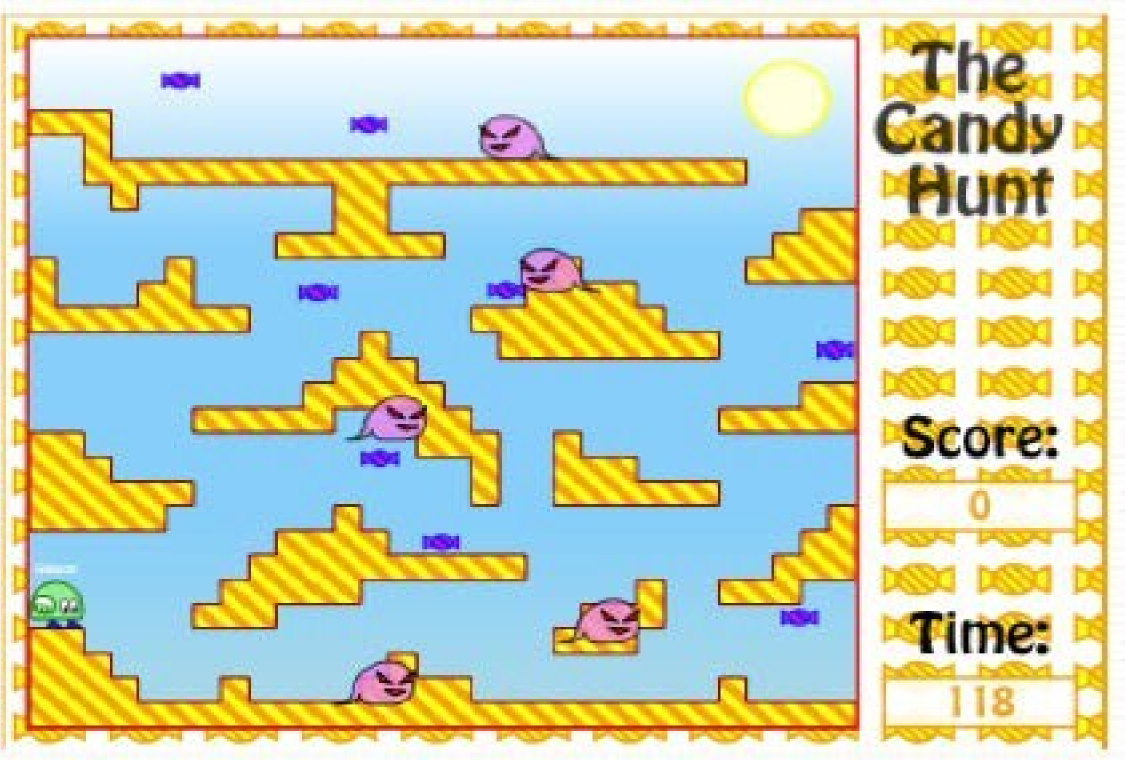


The Candy Hunt

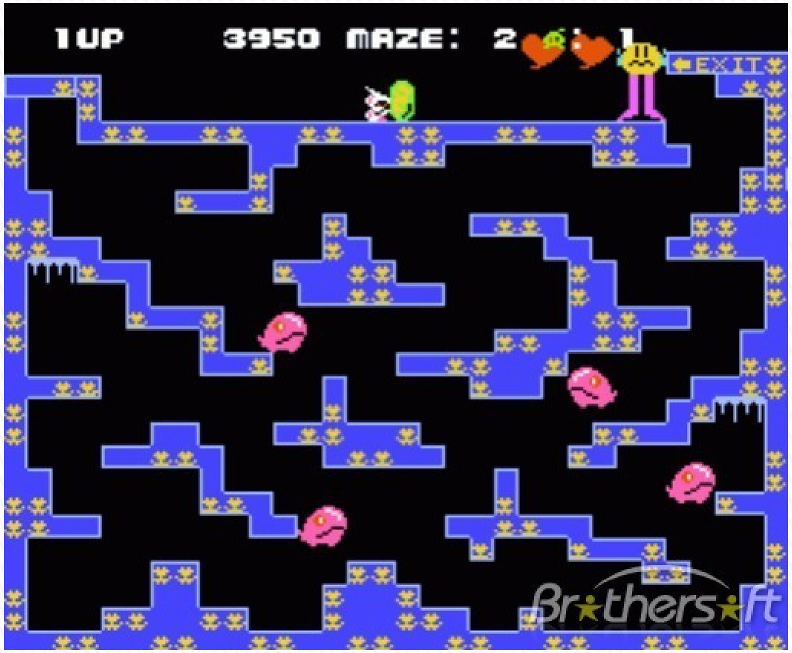
You are a candy hunter. You need to collect as many candies as possible. Use the N key to go left, the M key to go right and the Z key to jump. Ghosts are going to stop you on the way. Use the spacebar to shoot and kill them. Pick up the candies and go to the exit quickly. You can earn double points in zoom mode, which is much harder.

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

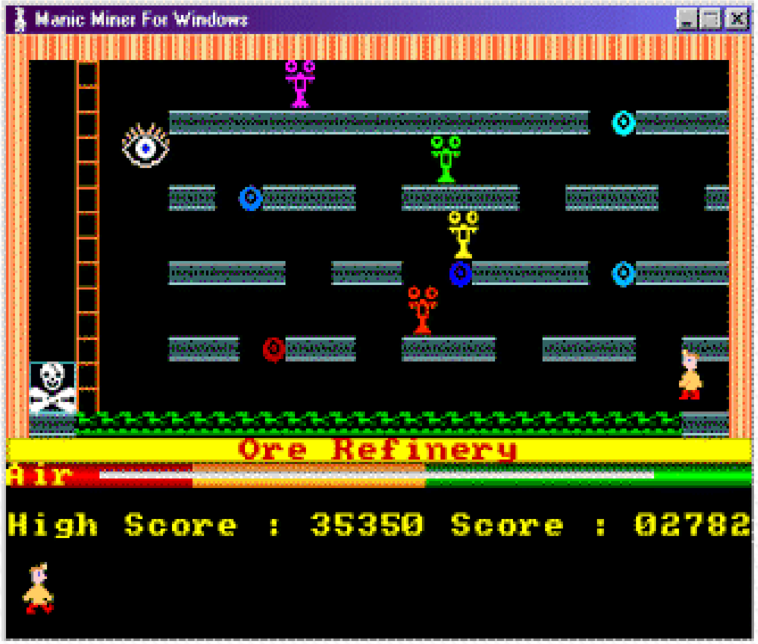
- After user clicks the start button



# Example of Commercial Games



Chack'n Pop, from 1983
1980's



Manic miner, from the

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

# Basic Idea of the Game 1/2

- After the game starts, the ‘time left’ will start from 60 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 remaining time is added to the player’s score

# Basic Idea of the Game 2/2

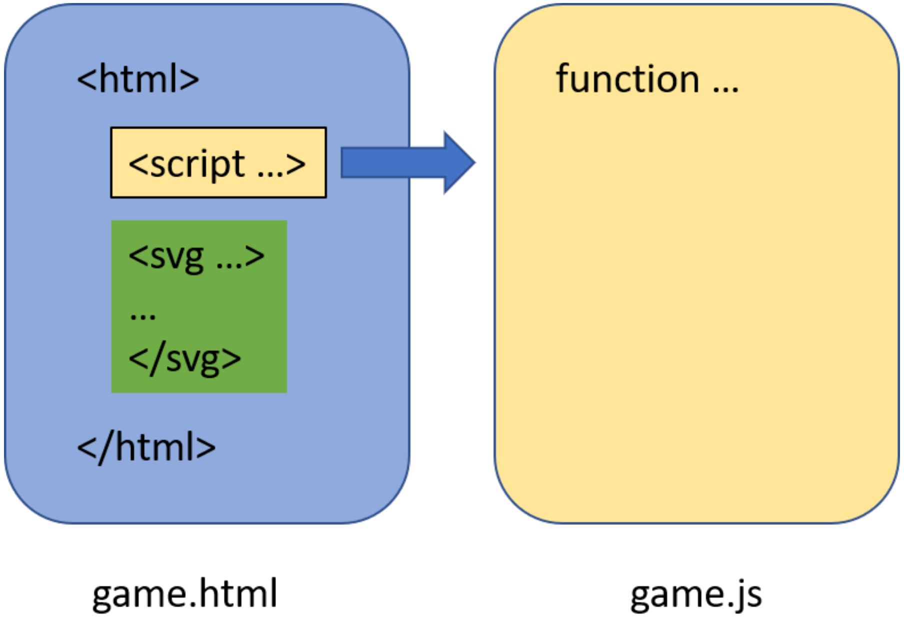
- Platforms
  - To get to the exit point, the player has to walk/jump on several platforms
- Good things
  - There are good things near the platforms; when the player gets one he/she increases the score
  - The player has to collect all the good things before he/she can finish the level i.e. get them 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 scores

# Summary of Keys

- Use these keys in your game (different  from the settings used in the labs):
  - w – jump
  - a – left
  - d – right
  - h – shoot

# File Arrangement

- Use the same file arrangement as in the labs
- Use SVG to implement all the interfaces (you can use prompt() for name input)



# Run the game

- Open the game’s HTML webpage directly

# Overview (1/2)

| | |
|---|---|
| Starting Screen | 7% |
| Handling of Player | 10% |
| Handling of Monsters | 12% |
| Handling of Good Things | 6% |
| Platforms | 9% |
| Transmission Portal | 9% |
| Shooting | 4% |
| Sound | 9% |
| | 5% |

# Overview (2/2)

| | |
|---|---|
| Time Remaining | 4% |
| Level Handling | 6% |
| Game Quality | 8% |
| Score Update and Display | 4% |
| End of Game | 8% |
| Handling High Score Table | 8% |

Total 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

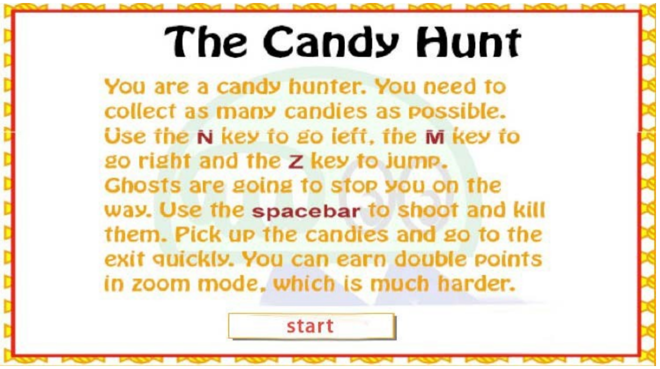
  Everything stored in the DOM

  …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.
- This is the same as the game we have in the lab.
- You can change the background pattern as you like, but the platforms (except the color) should be consistent with the game we have in the lab.

# Starting Screen 7%

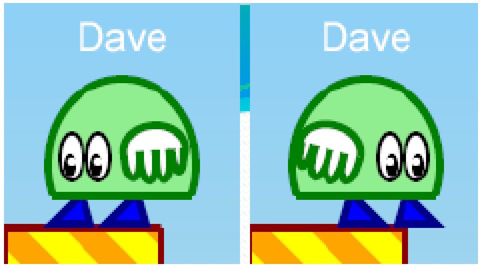


The Candy Hunt

You are a candy hunter. You need to collect as many candies as possible. Use the **N** key to go left, the **M** key to go right and the **Z** key to jump. Ghosts are going to stop you on the way. Use the **spacebar** to shoot and kill them. Pick up the candies and go to the exit quickly. You can earn double points in zoom mode, which is much harder.

- When the SVG starts you need to give the player some information
  - +2 Include the title of the game and your name. Give a general introduction to the game and tell the player what he/she needs to do
  - +2 Say what keys the user needs to press to play the game (left/right/jump/shoot)
  - +3 Display the start button and game only starts after clicking the **start** button
  - You can add anything else appropriate

# Handling of Player 10%



- +3 marks – ‘Flip’ player when move left/ right
- +3 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 or is shot by a bullet.

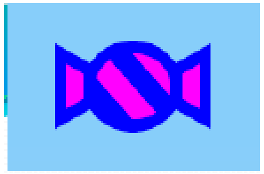
Please use the SVG File we provided

# Handling of Monsters 12%

- 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)-
  - +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
  - +3 marks – ‘Flip’ monster when move left/ right
  - +3 marks – Exactly one special monster can shoot bullet; at any time, at most one bullet can be shot in the same game window.

Please use the SVG File we provided

# Handling of Good Things 6%

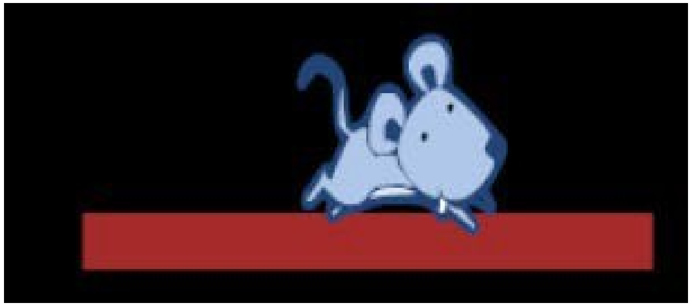


- There must be at least 8 good things in the game
  - +1 marks – The good things are generated at random places at the start of the game
  - +1 marks – The good things cannot appear within a platform, i.e. they should not overlap with any platforms
  - +2 marks – The player collects the good things 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

Please use the SVG File we provided

# Vertical Platforms 4%

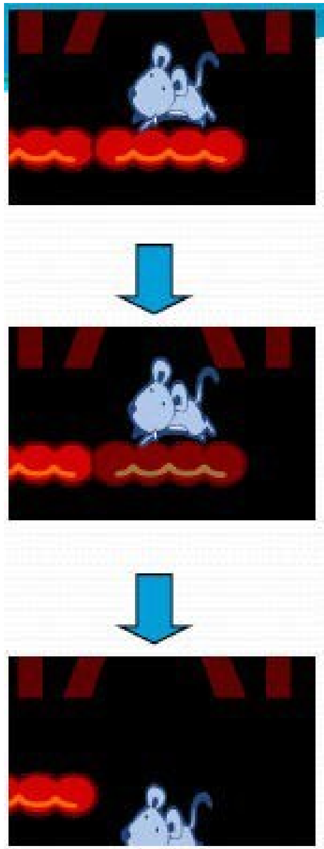
- +4 marks – There are one ‘vertical’ platform (platforms that move up and down in the y axis)



# Disappearing Platforms 5%

- +3 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 effect 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 player enters into one portal, it will appear at the position of another portal. (Positions of portals can be defined at will, e.g. upper-left and upper-right corners of the game interface)

# Shooting 9%

- +3 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
- +3 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)
- +3 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)
- The SVG of the bullet is the same as the game we have in the lab

# Sound 5%

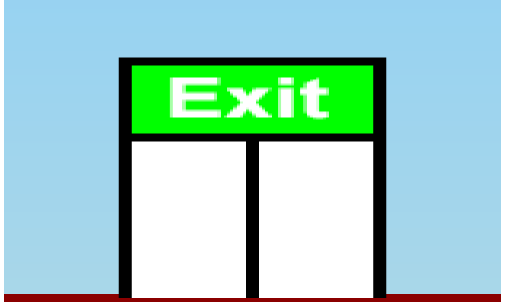
- Use of sound
  - +1 mark – Appropriate sound when the player shoots (shoot.mp3)
  - +1 mark – Appropriate sound when the player reaches the exit point before the time runs out and successfully collects all the good things (win.mp3)
  - +1 mark – Appropriate sound when the player dies (touches monster or runs out of time) (lose.mp3)
  - +1 mark – Appropriate sound when a monster dies (is shot by the player) (kill.mp3)
- +1 mark – Appropriate continuous music during the game (bgm.mp3)

# Time Remaining 4%

- The player needs to reach the exit point within the specified time (See the "Basic Idea" slide)
- The player will die if the player cannot reach the exit point within that time
- +4 marks - Time count down is updated and displayed appropriately every second (perhaps using a setInterval() )

Time Remaining:
55 sec.

# Level Handling 6%



- When the game first begins, it is level 1.
- When the player reaches the exit point, the score from the remaining time is added and the game moves to the next level, which is harder (see other slides)
- Don’t forget you have to collect all good things before you can finish the current level.
- The upper limit of the level is 2.
  - +1 mark – Appropriate appearance of the exit (looks like an exit)
  - +2 marks – The current level is shown in the GUI, and is updated appropriately. It is incremented by one each time the player finishes a level and moves to the next level.
  - +3 marks – The game is correctly re-started when the next level is started (i.e. score continues and is not reset to zero, etc.)

# Game Quality 8%

- How playable the game is
  - +8 marks – The game gets harder in level 2. This is achieved  by adding four monsters per subsequent level. I.e. If level 1 starts with 6 monsters, then level 2 will start with 10 monsters (the player always starts with 8 bullets, whatever the level is)

# Score Update and Display 4%

- +1 Score is updated at the end of each level. Add L * 100 points for passing level L.
- +1 And also add X points for each second of remaining time, where you choose an appropriate value of X.
- +1 Score is updated when a monster is shot – add Y points when this happens, you choose an appropriate value of Y
- +1 Score is updated when a good thing is touched – add Z points when this happens, you choose the value of Z

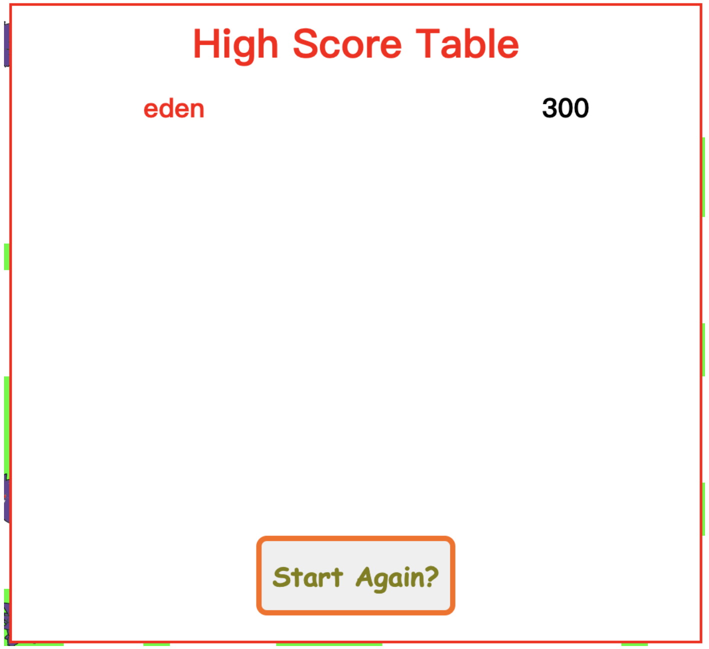
# End of Game 8%

- 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
  - +4 marks – Score & name are shown in a score table
  - +4 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 – The score of the player is correctly shown in the table
- +3 marks – Show a ‘Start again?’ button, if the player clicks on it the game begins again

Example high score display



High Score Table

| | |
|---|---|
| eden | 300 |

# Submission(1/2)

- Make your project run on Chrome; if it runs on other browsers but not Chrome, 10 points will be deducted
- You need to submit all the needed files and sounds
- If you want to write any message to the marker, write them in a file called ” readme.txt “

# Submission(2/2)

- Deadline: See Canvas comp 4021 -> Assignments
  - If you submit after the due date, your score will be penalized by 20% for each day after the due date.
  - Submissions are rejected 2 days after the due date.

- Format of submissions
  - Put all the files into a single zip file.
  - Filename: StudentID_Name_svg_comp4021.zip