



# Hidden Image Game

Version: 1.0.0 / Date: 06-15-16

Acknowledgement to the NSF and Kean University for funding research which aided in the production of this game.

Game Development Team:

Lead Developer and Designer - Stephanie Eordanidis

Assistant Programmer - Maitri Chakraborty

Technical Document written by: Stephanie Eordanidis

GDD Template Written by: Benjamin "HeadClot" Stanley

Special thanks to Alec Markarian

Check out my webpage for more projects and creations

<http://eve.kean.edu/~sleordan/Portfolio/index.html>

## Overview

Theme / Setting / Genre

Core Gameplay Mechanics Brief

Targeted platforms

Project Scope

The elevator Pitch

What sets this project apart?

Core Gameplay Mechanics (Detailed)

- (Level One) Runs
- (Level One) Grid Space Revealing
- (Level Two) Grid Space Toggling
- (Level Two) Guessing Image
- Statistics Capturing

Gameplay

Gameplay (Brief)

Gameplay (Detailed)

Assets Needed

- Scripts and Code
- Sprites
- Background

- Sound

- Objects

- Rooms

Game Screen Shots

- Main Screen
- Instructions
- More Explanation
- Credits
- Player Name Input Screen
- Level One Difficulty Choice
- Level One
- Level One End Screen
- Level Two Difficulty Choice
- Level Two
- Level Two Pause Screen
- Level Two Guess Screen
- Level Two End Screen
- Credit

## **Overview**

### **Theme / Setting / Genre**

- Hidden Image game is a supplemental educational computer science game that helps reaffirm and teach data representation in a fun way without intimidating players with hard to understand coding methods. The game is modeled after its analog counterpart. Refer to Screen Shots at end of Document

### **Core Gameplay Mechanics Brief**

- (Level One) Runs: players can utilize runs to decrease total move count
- (Level One) Grid Space Revealing: players can click on a grid space to reveal hidden data
- (Level Two) Grid Space Toggling: players can toggle between black/white tiles
- (Level Two) Guessing Image: players may guess what the hidden image is
- Statistics Capturing: after each level played, core stats are recorded and appended to a txt doc

## Targeted platforms

- This game is made to work as a standalone on Windows PC platforms

## Minimal System Requirements: Games On Windows PC

- Windows XP, Vista, 7 or 8
- 512MB RAM
- 128MB graphics
- Directx 9 or greater
- Directx 3D

## Project Scope

- <Game Time Scale - 3 months>
  - Allotted 3 month time frame to design, code, and implement game.
  - Cost: several game maker professional studio licenses. (Approx. \$99 / license).
- <Team Size - 2>
  - <Core Team>
    - Stephanie Eordanidis - Kean University Computer Science Dept. eordanis@kean.edu
      - Designed, coded, implemented large portion of the game. Also fixed bugs and performed fine tuning.
    - Maitri Chakraborty - Kean University Computer Science Dept. chakrabm@kean.edu
      - Assisted in coding portion of game.

## The elevator Pitch

Hidden Image game is a supplemental educational computer science game that helps reaffirm and teach data representation in a fun way without intimidating players with hard to understand coding methods. The game is modeled after its analog counterpart.

## What sets this project apart?

- This game encourages repetition of image representation computer science concepts in a fun and enjoyable way
- This game demonstrates computer science concepts without intimidating users/players with code.

## Core Gameplay Mechanics (Detailed)

### - (Level One) Run encoding

#### - <Details>

- Players can click on previously revealed grid spaces to check for a *Run*. Utilizing runs can help the player keep their move count low by revealing more data (grid spaces) in a shorted time frame and with less moves.

#### - <How it works>

- Players can click on previously revealed grid spaces to check for a *Run*.
  - A run would be 2 or more grid spaces total (starting to the right of a previously revealed grid space) that have the same data (color in our case) as the original grid space the player clicked.
  - If a run exists, ALL of the grid spaces with the same data (color) to the right of the one they checked will be revealed at no extra move cost to the player.

### - (Level One) Grid Space Revealing

#### - <Details>

- Players must click on the grid spaces (squares) to reveal what the data hidden beneath.

#### - <How it works>

- Players must click on the blue/gray grid spaces (squares) to reveal what the data hidden beneath. When clicked, the data underneath that space (either a black or white space) is revealed to the player. The more revealed, the cleared the image becomes.

### - (Level Two) Grid Space Toggling

#### - <Details>

- Players must click on the grid spaces (squares) to toggle between black/white grid spaces.

#### - <How it works>

- Players must click on the grid spaces (squares) to toggle between black/white grid spaces.
- When clicked, the grid space tile becomes either black or white. Player may toggle space to switch between the two colors.
- Players use the tiles to represent the decoding of the encodings provided to the left of the screen.

## - (Level Two) Guessing Image

### - <Details>

- Players may take a guess as to what they think the image is to earn bonus points.

### - <How it works>

- Players may take a guess as to what they think the image is to earn bonus points.
- The player can click the “My Guess” button to be brought to a screen with multiple images. The player then can click on the image they think matches the one they are decoding. If the image and guess match at the end of the game, the player gets an 8 point bonus.

## - Statistics Capturing

### - <Details>

- At the end of each gameplay level, Core statistics are recorded and appended to a text document. These stats can be used by educators to monitor student/player progress.

### - <How it works>

- The text document is located at :  
“C:\Users\<username>\AppData\Local\HiddenImageGameV37\HiddenImageGameStats.txt”
- The data recorded after each game play is:
  - Level One
    - Level Played
    - Date and Time
    - Player Name
    - Level Difficulty Chosen
    - Level Image Used
    - Time Remaining
    - Total Move Count
    - Black Space Count Remaining
  - Level Two
    - Level Played
    - Date and Time
    - Player Name
    - Level Difficulty Chosen
    - Level Image Used
    - Time Remaining
    - Total Points
    - Last Guess Made

# Gameplay

## Gameplay (Brief)

Two levels Total:

Level One:

Players must reveal the hidden image before the timer runs out and in as little moves as possible.

Level Two:

Players must reveal the hidden image by decoding the run encoding provided before the timer runs out and with as many points as possible.

## Gameplay (Detailed)

Level One:

Objective:

Players must reveal the hidden image before the timer runs out and in as little moves as possible.

How To Play:

- Players must click on the grid spaces (squares) to reveal what the data hidden beneath.
- Players can click on previously revealed grid spaces to check for a *Run*.
  - A run would be 2 or more grid spaces total (starting to the right of a previously revealed grid space) that have the same data (color in our case) as the original grid space the player clicked.
  - If a run exists, ALL of the grid spaces with the same data (color) to the right of the one they checked will be revealed at no extra move cost to the player.
- Players can pause/play the game by toggling the pause button or by pressing “p” on the keyboard to take a break.
- Players can mute/play the music in game by toggling the music note button or “m” on the keyboard.
- When a player reveals a black space, their next move is “free” (does not increase move count).

Level Two:

Objective:

Players must reveal the hidden image by decoding the run encoding provided before the timer runs out and with as many points as possible.

How To Play:

- Players must reveal the hidden image by decoding the run encoding provided to the left of the game screen.
- The encodings are displayed one at a time, row by row.
- Players must click (toggle) the grid spaces (squares) to switch between black and white spaces.
- Players can pause/play the game by toggling the pause button or by pressing “p” on the keyboard to take a break and to also check and see previously shown encodings.
- Players can mute/play the music in game by toggling the music note button or “m” on the keyboard.
- Players may take a guess as to what they think the image is to earn bonus points.
- Players who reveal the entire image before the timer runs out can press the “finish” button to end early.

## Assets Needed

### - Scripts and Code -

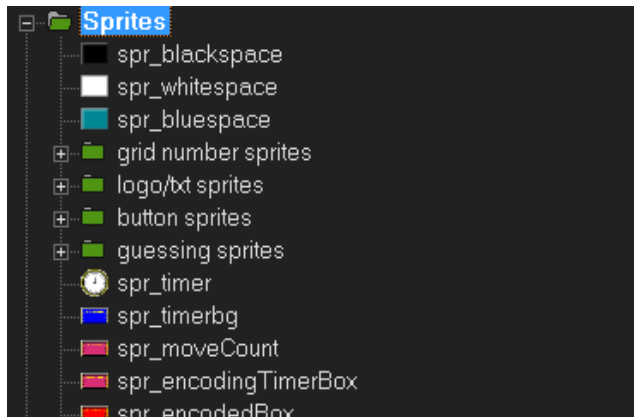
Core startup variables and functions are all located in scripts residing in the Logo object. This includes hard coded arrays, various variables important to different image arrays.

Level one hardcoded arrays and associated variables are located in obj\_logo -> Game Start -> Actions -> 1<sup>st</sup> script file

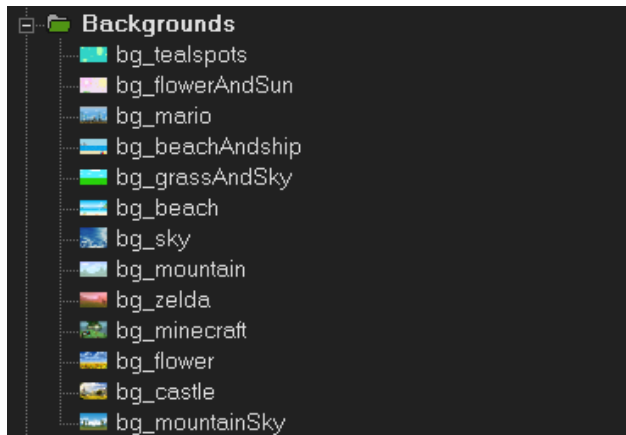
Level two hardcoded arrays and associated variables are located in obj\_logo -> Game Start -> Actions -> 2<sup>nd</sup> script file

Also important, if adding additional image arrays you must also increase the randomized range and follow format located in obj\_logo -> Game Start -> Actions -> 3<sup>rd</sup> script file in order for game to access them.

- **Sprites** - Below is an image with the list of sprites (images) use in the game. Some are organized into folders to make the file neater. The prefix for sprites are “ spr\_ “.

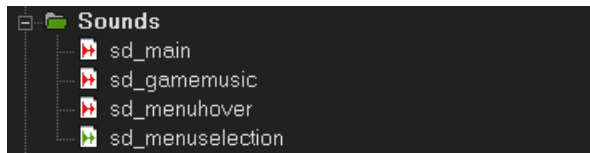


- **Backgrounds** - Below is an image with the list of backgrounds used in the game. The game randomized them, so they won't likely repeat with each gameplay. The prefix for backgrounds are “ bg\_ “.

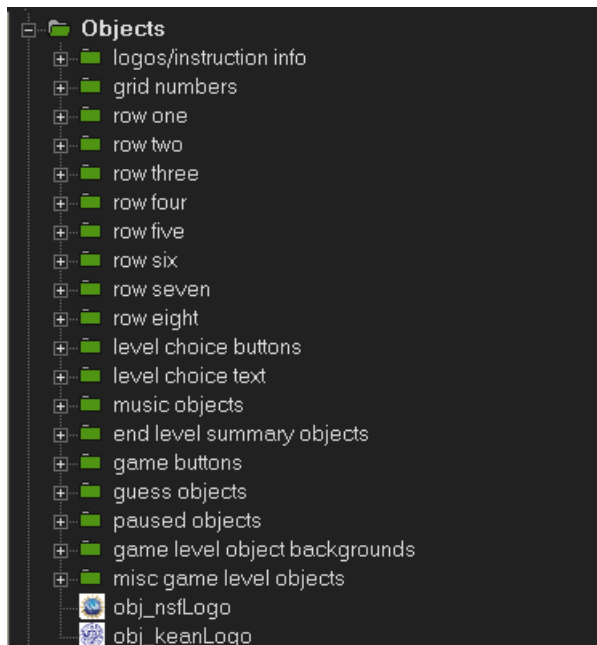




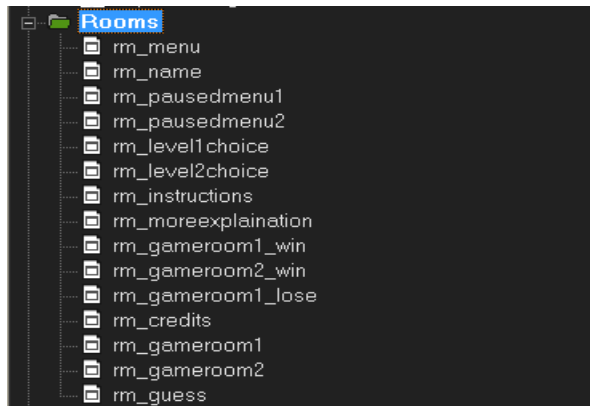
- **Sound** - Below is an image with the list of sounds used in game. They are labeled according to what they are used for. The prefix for sounds are “ sd\_ “.



- **Objects** - Below is an image with the list of objects used in game. They are labeled according to what they are used for in the game. They are sorted into folders for consistency and organization. The prefix for objects are “ obj\_ “.

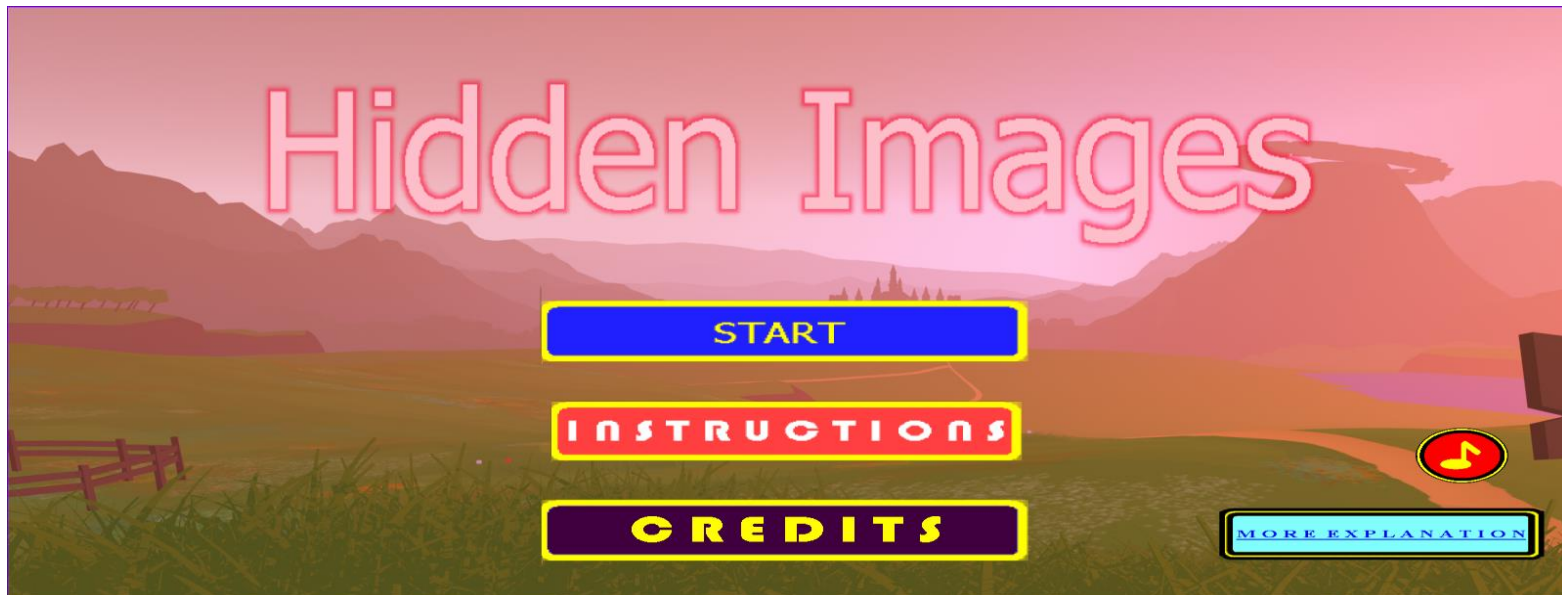


- **Rooms** - Below is an image with the list of rooms used in the game. They are labeled according to what room they are in the game. The prefix for rooms are “ rm\_ “.



## Game Screen Shots

- Main Screen



## - Instructions

### Level One

---

**Objectives:**

- Reveal the hidden image before the timer runs out and in as little moves as possible.

**How To Play:**

- Click on the grid spaces (squares) to reveal what is hidden beneath.
- Click on a grid space already revealed to check for a *Run*.
  - A Run would be 2 or more grid spaces total (starting to the right of a grid space already revealed) that have the same data (color) as the original one you clicked.
  - If a Run exists, ALL of the grid spaces with the same data (color) to the right of the one you check will be revealed at no extra move costs to the player.
- You can press the pause button (or “p” on your keyboard) to pause the game and take a break. Press “p” again or the pause icon to resume playing.
- When you reveal a black space, your next move is “free”!

---

### Level Two

---


**Objectives:**

- Reveal the hidden image by using the run encoding provided before the time runs up.

**How To Play:**

- Reveal the hidden image by using the run encoding provided to the left of the game screen.
- The encodings are displayed one at a time, row by row.
- Click (or toggle) the grid spaces (squares) to switch between black and white spaces.
- You can press the pause button (or “p” on your keyboard) to pause the game and also see all the encodings displayed so far. Press “p” again or the pause icon to resume playing.
- Once per row, you can guess what you think the image is to gain bonus points by pressing the guess icon at the top of the screen or by pressing “g” on the keyboard.
- Reveal the entire image before the timer runs out and press the “finish” button to end early to win!

---



## - More Explanation

### HIDDEN IMAGE GAME

#### DATA REPRESENTATION

Data can be represented in many different ways, and still have the same meaning. The different representations are for different purposes (faster, shorter, more visual, more hidden, and more secretive). Changing from one representation to another has to follow agreed translation rules.

#### WHAT IS RUN LENGTH ENCODING?


**Run-length encoding (RLE)** is a very simple form of data compression in which **runs** of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original **run**.

*Run-Length Encoding Example:*

1B2W3B2W = Run Length encoding  
10011100 = Binary encoding  
Whose image is :

--	--	--	--	--	--	--	--

There are many different protocols you can use when it comes to encoding data. *Protocols* are rules for encoding data which changes the look of the data but not the meaning of the data. In the Run length encoding above, the data is B for black and W for white, but could just as easily be R for red and B for blue. The Protocol also includes Runs of variable length. For example, the black has a run of length three and the white has two runs of length two.



- Credits





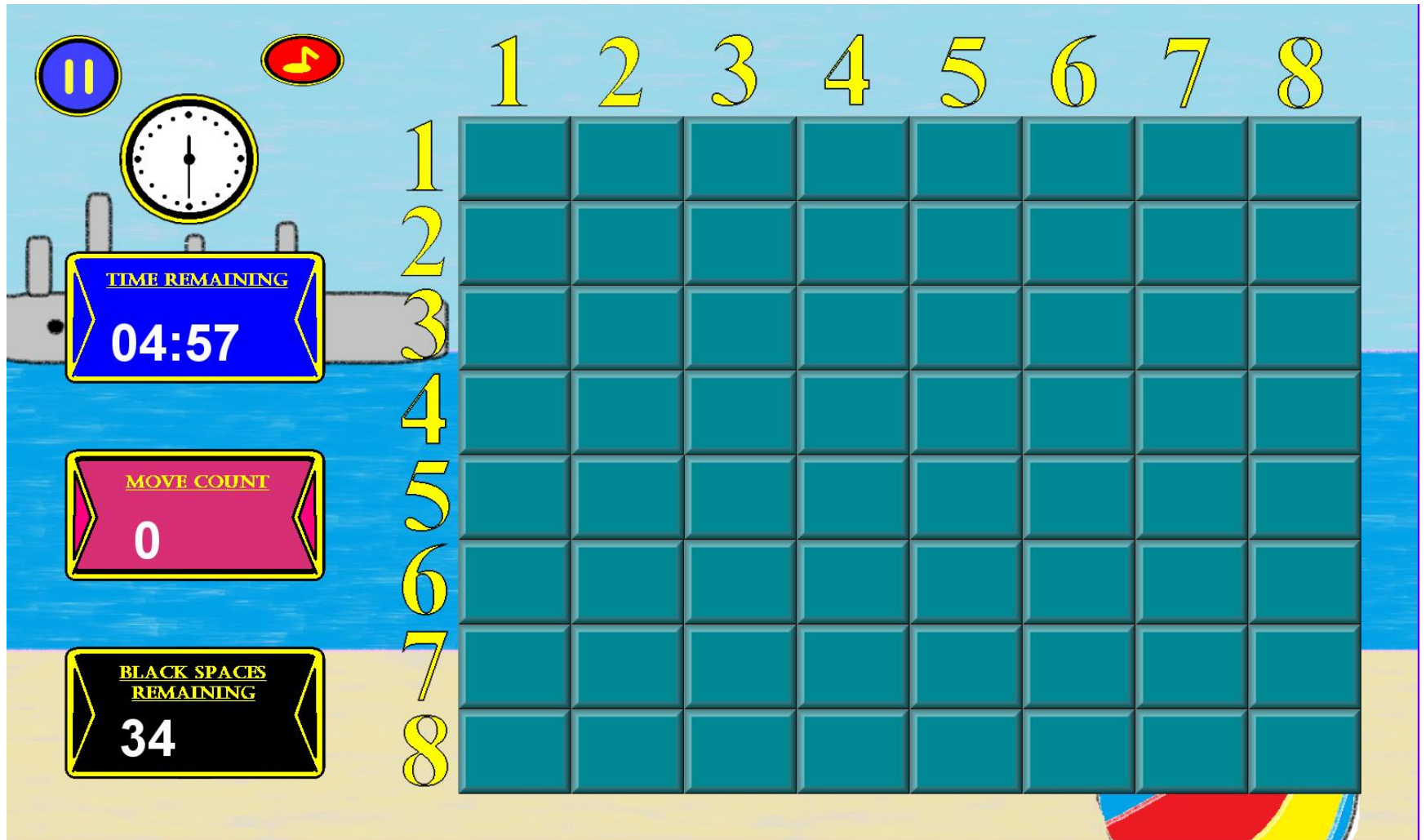
- Player Name Input Screen



- Level One Difficulty Choice



- Level One





- Level One End Screen



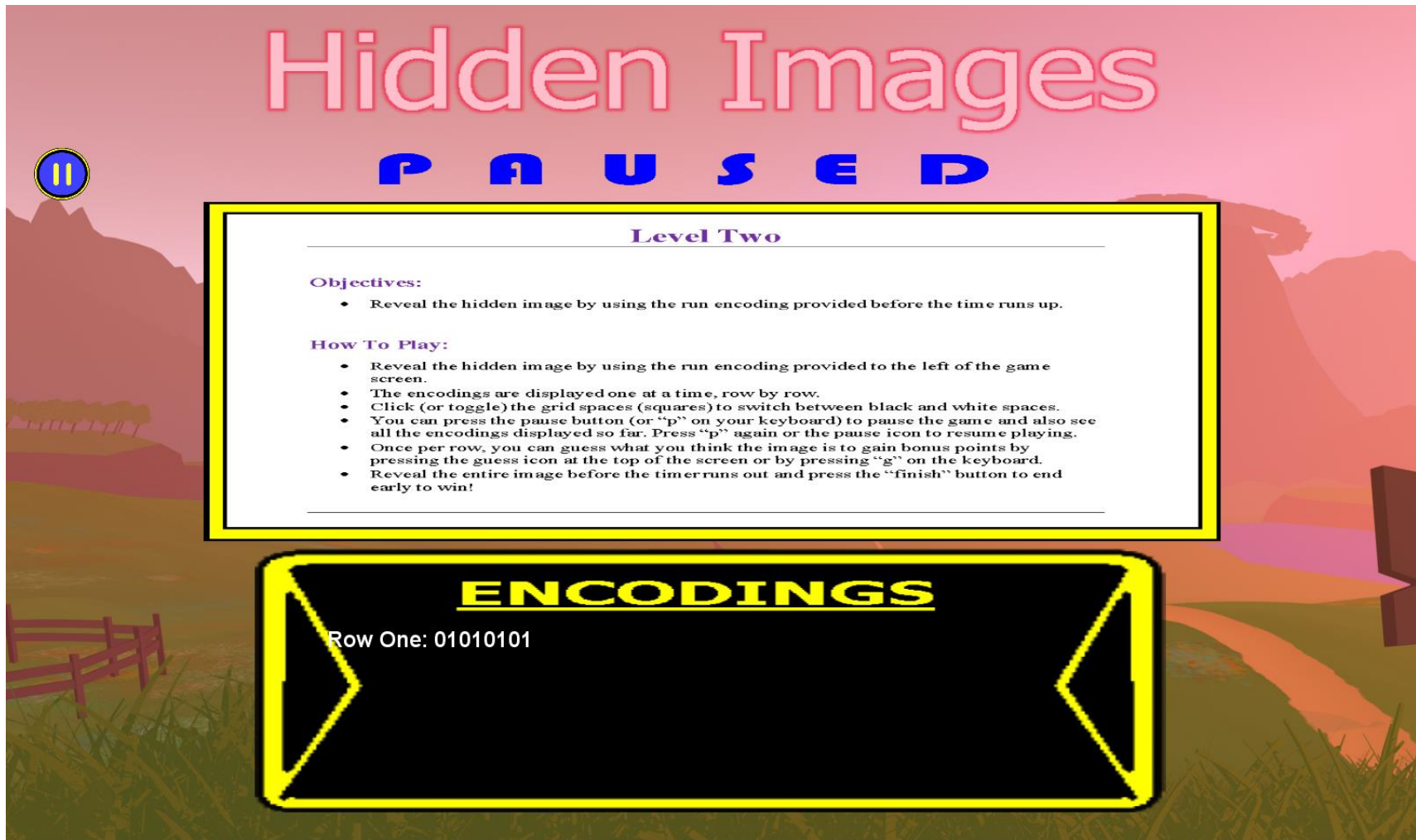
- Level Two Difficulty Choice



- Level Two



- Level Two Pause Screen





- Level Two Guess Screen



- Level Two End Screen



## **CREDITS:**

Music and Backgrounds - Credit is given below to authors or author source for Images and Sounds in this game. All Sources were located using google search.

### **Backgrounds Courtesy of:**

#### **Main Screen**

<https://images7.alphacoders.com/> - Legend of Zelda OOT

#### **Game Level**

<http://images6.fanpop.com/> - Minecraft

Unknown Source (May have been Removed) - Mario

<http://wallpaperbeta.com/> - Sunflowers and Sky

<https://www.pinterest.com/pin/536632111823593558/> - Castle

<http://content.chupamobile.com/> - Mountain and Sky

<http://opengameart.org/> - Mountain

<http://www.walldevil.com/> - Clouds

<http://wallarthd.com/> - Beach

Stephanie Eordanidis - Teal Spotted

Stephanie Eordanidis - Flower and Sun

Stephanie Eordanidis - Beach and Ship

Stephanie Eordanidis - Grass and Sky

### **Sounds Courtesy of:**

<http://www.playonloop.com/music-loops-category/videogame/>

<http://soundbible.com/>