Double Helix

Design Document

It's time to unwind.

Version 3.0

Annie Dickerson

Document Overview

Overview

This is the design document for *Double Helix*. This document is broken into major sections, each describing a particular aspect of the game. The appendices offer supporting information for the development of the game.

Revision History

This is a brief description of this document. A list of the major changes is provided following each revision number. Furthermore, a list of any outstanding topics or any topic that needs further details is provided.

Revision	Date	Change Description
1.0	Nov. 28, 2010	• First Draft Pass
2.0	Dec. 14, 2010	• Interface screens added
3.0	Dec. 15, 2010	• Peer and instructor feedback incorporated; appendices, tables, and figures updated; movement mechanic updated

Table of Contents

Table of Figures

Table of Tables

Game Concept

Overview

This section describes the main concept of *Double Helix* and explains the philosophy, purpose, and focus of the game.

High Concept

Double Helix is a run-forever rhythm platform game for iOS devices that combines the principles of real world DNA replication with the challenges of quick decision-making. In *Double Helix*, you play as Junior, a nanobot tasked with replicating sequences of DNA in an effort to preserve the human genome before a meteorite hits the Earth. Using knowledge of nucleotide pairs, you must tap nucleotide buttons in time with the beat to guide Junior along the DNA strands, fixing errors and replacing missing nucleotides.

Philosophy

Combining DNA replication with rhythm game mechanics makes an otherwise complicated biological principle easier to understand and more appealing for a broader audience.

It’s time to unwind.

Double Helix challenges you to make quick decisions and to react in a timely manner as the DNA double helix unwinds, presenting new obstacles at increasing rates.

Common Questions

Why Create this Game?

Double Helix provides a new theme and mechanic for the rhythm game genre, adding elements of decision-making and platforming to otherwise repetitive sequences and combos. The goals set forth in creating this game include, but are not limited to, the following:

- Add player choices and platforming to traditional rhythm game mechanics
- Combine real world scientific principles with casual game mechanics

What is the Target Hardware?

Initial development will be for iOS devices, namely iPhone and iPod Touch.

Where Does the Game Take Place?

Double Helix takes place within the nucleus of a single human cell as your nanobot Junior travels along sequences of an unwinding DNA double helix.

What Do I Control?

You control the movement of your nanobot Junior along the double helix by tapping the four buttons associated with the four basic nucleotides (A – adenine, T – thymine, G – guanine, and C – cytosine). Junior will move to the nucleotides you specify.

What is the Main Focus?

Double Helix challenges you to combine platforming and rhythm to make quick decisions in time with the beat.

Feature Set

This section describes the features that make up the overall gameplay experience of *Double Helix*.

Major Features

Rhythm-Based Platforming

Moving your nanobot along the double helix requires choosing the correct nucleotides at the right time. You can choose different paths for your nanobot, as long as you stay with the beat. As you progress, the tempo will increase, requiring faster decisions and reactions (see Rhythm-Based Movement in the Gameplay Details section).

Pickups

Along the way, pickups appear at certain intervals. Pickups can double or triple the points you accrue during a certain period, temporarily change the tempo, or send in a replacement nanobot to take over for Junior, thus providing you a brief break (see Rewards in the Gameplay Details section).

Nanomite Invasions

Though Junior performs the main work of repairing the DNA sequences, he is not alone in the cell. Nanomite invasions occur at set intervals, changing the nucleotides to try to cause replication errors (see Enemy Characters in the Game Characters section).

Real World Theme

Double Helix uses the principles of DNA replication, allowing you to see and participate in the process. Complications such as replication errors and missing nucleotides challenge you to use memory and knowledge of nucleotide pairs to make quick decisions.

Minor Features

A Variety of Music

With 20 different upbeat instrumental tunes, each game will provide new and different experiences (see Appendix B for more details on game audio).

Scores & Stats

A variety of statistics about your gameplay will be tracked, including scores and play times (see the Scores and Stats Screen section in Appendix C).

Game Characters

Overview

This section describes the main character and enemies found in *Double Helix*.

Playable Character

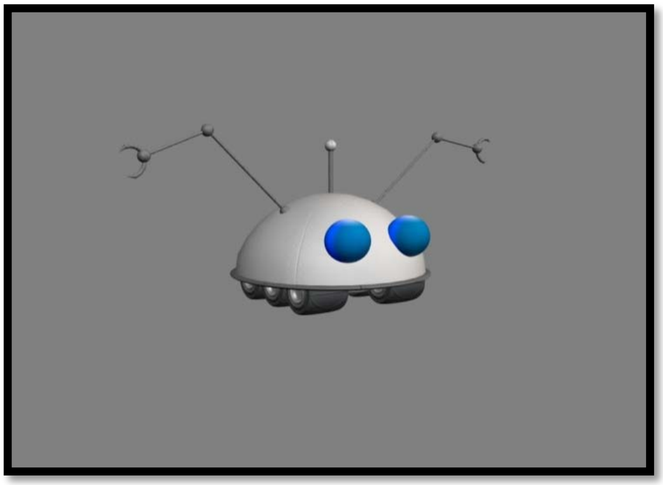
Overview

The only playable character in *Double Helix* is your nanobot Junior, who moves through the cell based on your commands.

Main Character

Measuring just one nanometer, Junior is microscopic in size. He moves along the double helix based on the tempo and can move either on the same strand or back and forth between the two DNA strands upon command (see Rhythm-Based Movement in the Gameplay Details section).

Figure 1: Main Character Concept



Enemy Characters

Overview

Nanomites invade the cell at set intervals, altering the existing sequence in an attempt to cause replication errors (see Nanomite Invasions in the Gameplay Details section).

Nanomites

Each nanomite alters one nucleotide by appearing briefly on the nucleotide, then disappearing and leaving behind a changed or missing nucleotide (see Nanomite Invasions in the Gameplay Details section).

Nanomites cannot be killed, though they will not directly harm your nanobot. However, every mutation they successfully bring about leads to one of your three allotted errors if you fail to navigate Junior to the mutations to fix them. The two types of nanomites include the following:

- **Nanomite Blaster** – destroys the nucleotide it lands on, making that nucleotide disappear (see Figure 2)
- **Nanomite Converter** – changes one nucleotide to another nucleotide (see Figure 3)

Figure 2: Nanomite Blaster Concept

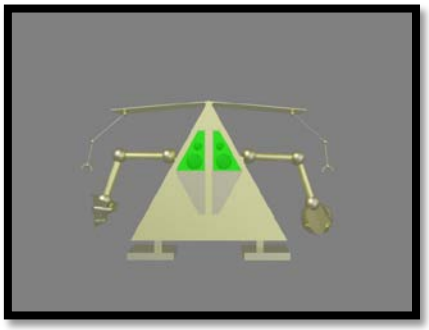
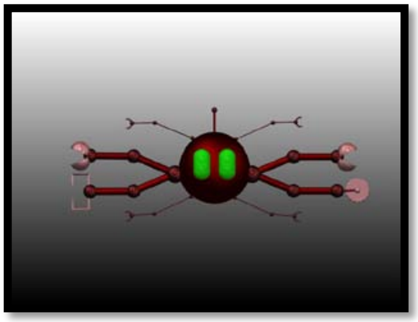


Figure 3: Nanomite Converter Concept



Camera

Overview

This section describes the camera used in *Double Helix*.

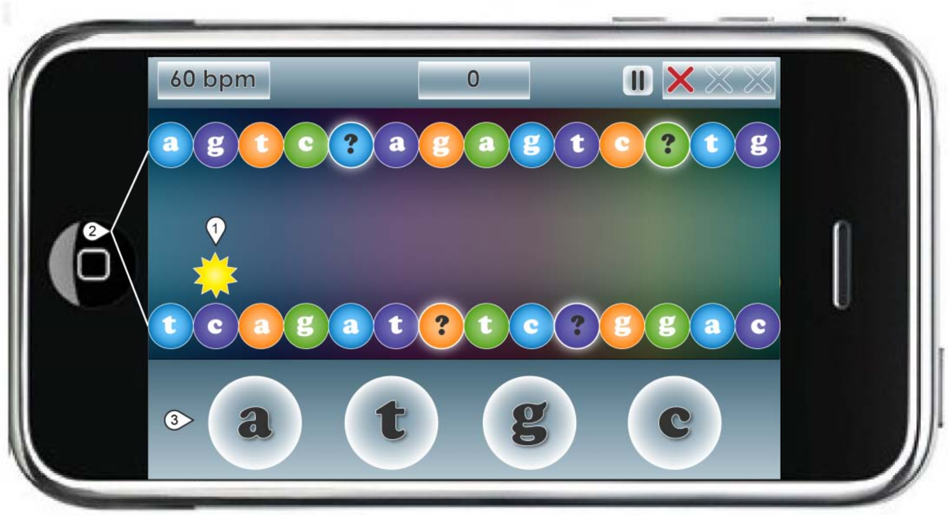
Fixed Camera

Double Helix uses one camera that remains fixed at a certain offset from your nanobot Junior to track his movement. The camera does not move; only the nucleotides scroll from right to left. Movement speed of the nucleotides in the DNA strands depends on the current tempo (see Screen Progression Rate in the Gameplay Details section).

Sample Screen

Figure 4: Sample Screen

(See the HUD section in Appendix C for more details on individual HUD components.)



1. **Junior**

   Junior measures 1 nanometer and will always remain on the second nucleotide pair from the left (see Movement and Navigation in the Gameplay Details section). The faster you input nucleotide commands, the faster the screen will scroll (see Screen Progression Rate in the Gameplay Details section).

2. Nucleotides

As upcoming nucleotides are revealed on the right side of the screen, previous nucleotides will scroll off the screen to the left. The screen will show 14 nucleotide pairs at all times (subject to adjustment). Each nucleotide represents 1 nanometer in diameter (roughly the same size as Junior), and the two strands should be at least 2 nanometers apart to make it clear which strand Junior is currently on.

NOTE:	Measurements and number of nucleotide pairs on screen are subject to adjustment.

3. Player Buttons

The four player buttons will appear in the bottom third of the screen. Buttons should measure at least ½ inch in diameter (subject to adjustment).

Controls

Overview

This section describes the main controls used to guide the movement of the main character in *Double Helix.*

Default Control Scheme

The four nucleotides you can use to guide Junior appear as circular buttons along the bottom third of the screen. Each button should be ½ inch in diameter (subject to adjustment), large enough to ensure that button size doesn't hinder ease of player input.

The default control scheme lines up the four nucleotide buttons in the following order: A, T, G, C (see Figure 5). However, you can change the order to suit your preferences (see Figure 6). See Appendix C for more details.

Figure 5: Default Control Scheme

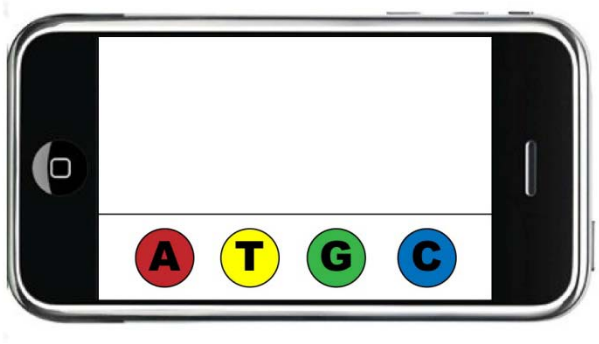
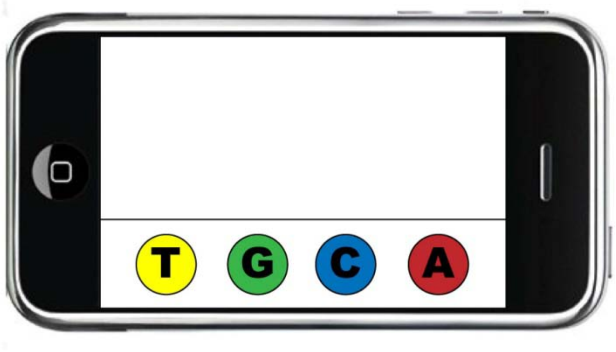


Figure 6: Player-Chosen Control Scheme Sample



Nucleotide Buttons

Button Functions

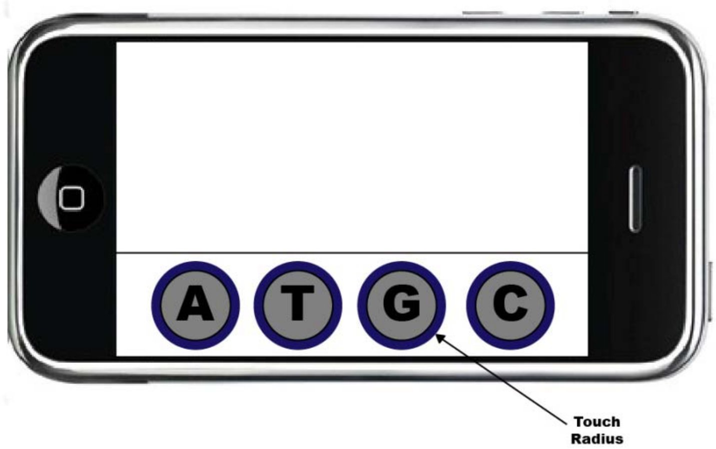
In order to move Junior to different nucleotides, you need to tap the nucleotide button corresponding to the nucleotide you want Junior to move to. Each button you tap will move Junior to the corresponding nucleotide. If you tap more than one button simultaneously, no input will be counted.

For example, in the sample screen above (see Figure 4), if Junior is to the left of the nucleotide pair A-T and you want to move him to T, you would tap the T button. If you tap G or C, you would have made one of your three allotted errors, since Junior can only move to one of the nucleotides in the pair to its immediate right (see Table 2 for movement scenarios).

Touch Radius

To move Junior, you must tap the nucleotide buttons in time with the beat. The center of your tap must fall within the touch radius of each button, which should measure 125% the size of each button graphic. For example, if each button measures 0.5 inch in diameter, the touch radius should measure 0.625 inch in diameter (see Figure 7).

Figure 7: Touch Radii for Nucleotide Buttons



Gameplay Details

Overview

This section describes the gameplay and mechanics found in *Double Helix* and contains topics describing the mechanics, how they are used, and their implementation in the game.

Rhythm-Based Movement

In order to guide Junior’s movement along the strands of the double helix, you must tap the correct nucleotide buttons in time with the beat. This section describes the player inputs, game components, and timings necessary for Junior’s rhythm-based movement.

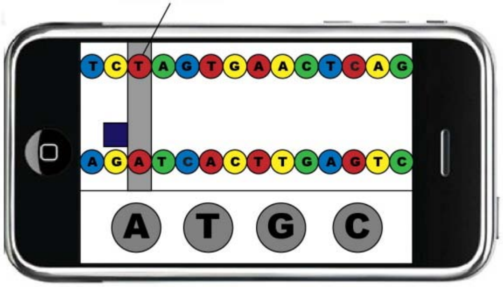
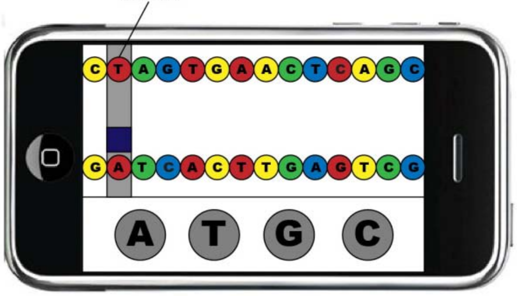
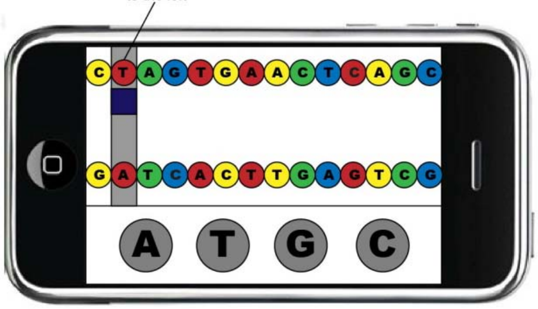
Current Tempo

The current tempo will be displayed in the top left corner of the screen (see the HUD section in Appendix C). The current tempo box will flash white on each beat to communicate the speed of the beat to the player, similar to a metronome. The current tempo is measured in beats per minute (bpm) and corresponds with the tempo of the music (i.e., as bpm increases, the music will speed up, and as bpm decreases, the music will slow down).

Nanobot Movement

Junior’s physical location on screen remains fixed throughout the game, as Junior always stays on the second nucleotide pair from the left. Junior “moves” from left to right and can only move from one nucleotide pair to the next; he cannot skip over nucleotides. The nucleotides scroll from right to left, simulating his movement (see Screen Progression Rate in the Run-Forever section below). Each movement to a new nucleotide takes 100 milliseconds. You can direct Junior to stay on the same strand or jump to the opposite strand (see Table 1).

Table 1: Movement Paths

Junior in original location	Nucleotide pair Junior will move to next  T C T A G T G A A C T C A G A G A T C A C T T G A G T C A T G C
Movement on the same strand	**Tap A to move Junior to the right** Nucleotides scroll to the left  C T A G T G A A C T C A G C G A T C A C T T G A G T C G A T G C
Movement to the opposite strand	**Tap T to move Junior to the opposite strand** Nucleotides scroll to the left  C T A G T G A A C T C A G C G A T C A C T T G A G T C G A T G C

Nucleotide Targets

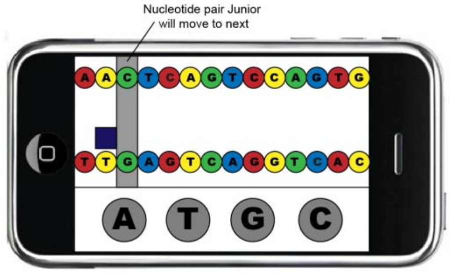
In order to direct Junior to move to specific nucleotides, you must tap the corresponding nucleotide buttons. For example, if you want Junior to move to T, the nucleotide to his immediate right, you would tap the T button (see Table 2 for examples).

Different Paths

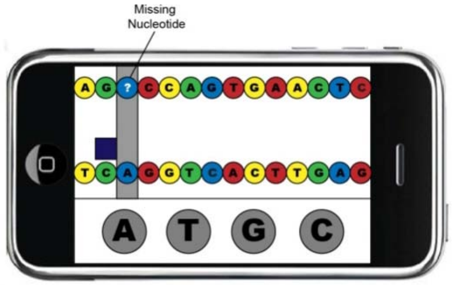
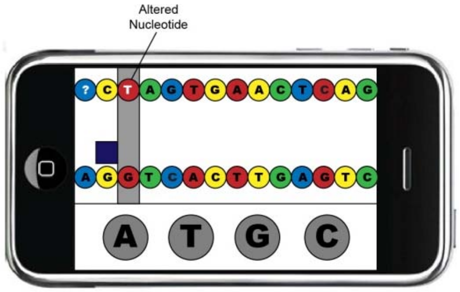
Though you must guide Junior to spots with missing and altered nucleotides, you can choose where Junior navigates the remainder of the time. Staying on the beat involves moving Junior to one nucleotide per beat. For example, by tapping one button per beat at 60bpm, you would move Junior to one nucleotide per second.

If the next nucleotide pair does not contain a missing or altered nucleotide, you can choose whether to keep Junior on the same strand or move him to the opposite strand. However, the nucleotide button you press must correspond with one of the nucleotides in the adjacent pair, since Junior cannot skip over any nucleotide pairs (see Table 2 for sample scenarios).

Table 2: Movement Scenarios

<table>
<tr><th rowspan="2">Scenarios</th><th colspan="4">Nucleotide button you choose to press</th></tr>
<tr><th>A</th><th>T</th><th>G</th><th>C</th></tr>
<tr><td>

Nucleotide pair Junior will move to next
A A C T C A G T C C A G T G
T T G A G T C A G G T C A C
A T G C

</td><td>Error</td><td>Error</td><td>Junior moves one nucleotide to the right</td><td>Junior moves to top strand</td></tr>
<tr><td> </td><td>Error</td><td>Correct!
Junior moves to the missing nucleotide spot and fills it in with a T.</td><td>Error</td><td>Error</td></tr>
<tr><td> </td><td>Error</td><td>Error</td><td>Error</td><td>Correct!
Junior moves to the altered nucleotide spot and fills it in with a C.</td></tr>
</table>

Input Timing

In order to guide Junior's movement along the double helix, you must tap the correct nucleotide buttons in time with the beat (i.e., one button per beat).

Input Accuracy

Depending on the tempo, you have a set amount of time to move Junior to the next nucleotide. You must tap the nucleotide buttons within a certain margin of the beat. A slower tempo provides a larger margin, while a faster tempo has a smaller margin (see Table 3).

Table 3: Margin of Error

Tempo Range **(bpm)**	**Margin of Error** **(__ milliseconds within the beat)**
60-79	500
80-99	300
100-120	200

Run-Forever

Double Helix consists of a single level in which you work to guide Junior as far as possible along the DNA strands. This section details the specifics of the tempo and player lives central to the run-forever mechanic.

Player Accuracy

You can make three errors before the game ends. The following error types are possible:

- Choosing a nucleotide not adjacent to Junior’s current location
- Choosing the wrong nucleotide for a missing nucleotide spot
- Guiding Junior past an error caused by a nanomite without fixing it
- Missing the beat by more than the allowed margin of error (see Table 3)

Tempo Changes

Each game starts at a tempo of 60bpm. As you progress along the double helix, the tempo will change based on your accuracy rate.

- Every five seconds you go without any errors, the tempo will increase by 4bpm.
- Each error you make will result in a tempo decrease of 8bpm.

Screen Progression Rate

The rate of nucleotide movement from right to left across the screen corresponds to the current tempo. One nucleotide pair will be revealed per beat. Thus, the higher the tempo, the faster the nucleotides in the sequence will be revealed (see Table 4). The following equation will be used to determine the screen progression, defined in terms of the number of nucleotide pairs revealed per second:

$$\boxed{\text{screen progression} = \frac{\text{current tempo}}{60}}$$

Table 4: Screen Progression Rate

Tempo (bpm)	Nucleotide Pairs Revealed Per Second
60	1.0
64	1.1
68	1.1
72	1.2
76	1.3
80	1.3
84	1.4
88	1.5
92	1.5
96	1.6
100	1.7
104	1.7
108	1.8
112	1.9
116	1.9
120	2

Missing Nucleotides

Along the way, certain nucleotides in the sequence will be missing. This section details the frequency of missing nucleotides and how to guide Junior's movement through them.

Frequency of Missing Nucleotides

The frequency of missing nucleotides depends on tempo. At faster tempos, more missing nucleotides will appear on screen (see Table 5).

Table 5: Frequency of Missing Nucleotides

Tempo Range (bpm)	Maximum Number of Missing Nucleotides on Screen
60-79	3
80-99	5
100-120	5

Appearance of Missing Nucleotides

Missing nucleotides will scroll on screen from the right at random times, as long as the number of missing nucleotides doesn't exceed the maximum (see Table 5). Missing nucleotide locations will scroll with the nucleotides they originate on. Missing nucleotides can appear within adjacent nucleotide pairs, but only one nucleotide at most within each pair can be missing. Each missing nucleotide replaced by Junior enables the appearance of an additional missing nucleotide.

Moving to Missing Nucleotides

In order to move Junior to spots with missing nucleotides, you must find the related nucleotide in the opposite strand and tap the matching nucleotide. Nucleotide pairs will be color-coded to more clearly communicate their relationship to each other (see Appendix A). For example, if the missing nucleotide is opposite A, you would tap T to move Junior to the missing nucleotide spot (see Table 6).

Table 6: Nucleotide Matches

Related Nucleotide in Opposite Strand	Nucleotide Needed
A (adenine)	T (thymine)
T (thymine)	A (adenine)
G (guanine)	C (cytosine)

Related Nucleotide in Opposite Strand	Nucleotide Needed
C (cytosine)	G (guanine)

Correct Nucleotides

Once you tap the nucleotide button to move Junior to a missing nucleotide, the correct nucleotide will be revealed. If you choose the correct nucleotide, Junior will move to the nucleotide, and the correct nucleotide will be revealed in a different text color (see Figure 8). If you choose an incorrect nucleotide, Junior will move still move to the nucleotide, but an error will be logged, and the correct nucleotide will be revealed with a red X (see Figure 9).

Figure 8: Correct Nucleotide Chosen

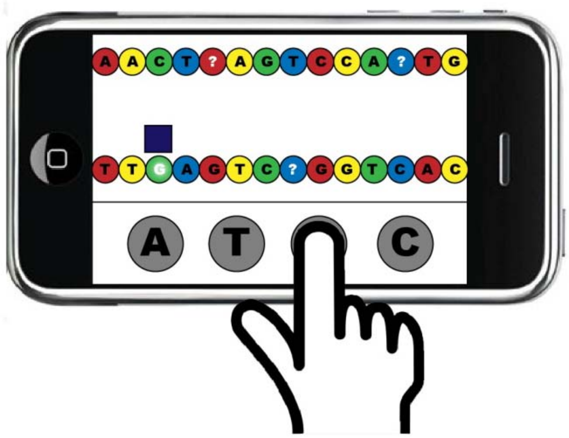
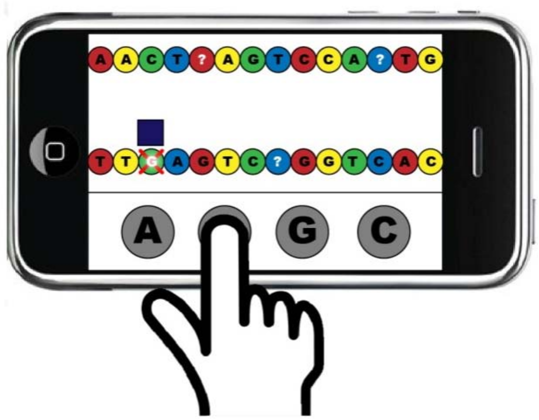


Figure 9: Incorrect Nucleotide Chosen



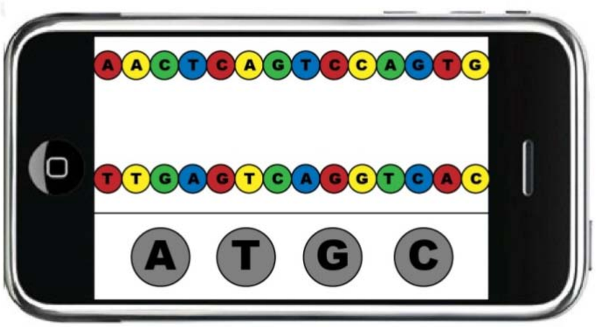
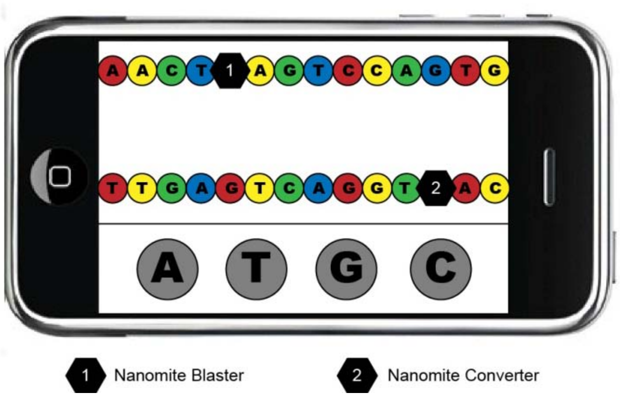
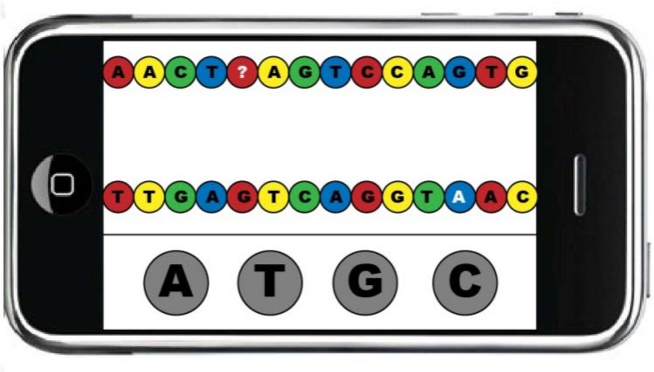
Nanomite Invasions

Along the way, nanomites will invade and change existing nucleotides. This section includes information on frequency of nanomite invasions and types of nanomites.

Stages of Nanomite Invasions

Each nanomite appears and covers a nucleotide, causes an error, then disappears after 500 milliseconds (see Table 7). The text color for any nucleotides altered by nanomites will change after the nanomites disappear.

Table 7: Stages of Nanomite Invasions

Before	 A A C T C A G T C C A G T G T T G A G T C A G G T C A C A T G C
During	 A A C T 1 A G T C C A G T G T T G A G T C A G G T 2 A C A T G C 1 Nanomite Blaster 2 Nanomite Converter
After	 A A C T ? A G T C C A G T G T T G A G T C A G G T A A C A T G C

Frequency of Nanomite Invasions

Nanomites invade at increasing rates based on the tempo (see Table 8).

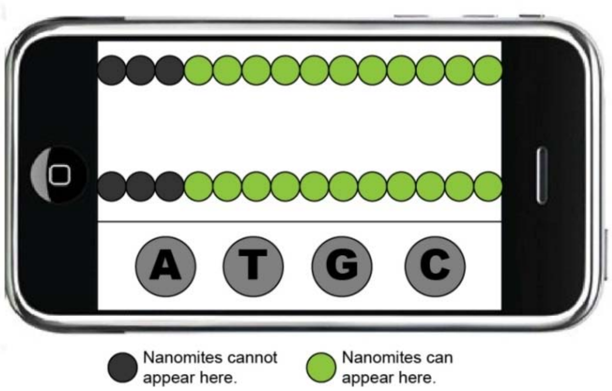
Table 8: Nanomite Invasions

Tempo Range (bpm)	Frequency of Invasions	Number of Invaders	Type(s) of Invaders
60-79	Every 5 seconds	3	Blasters
80-99	Every 3 seconds	4	Blasters and Converters
100-120	Every 3 seconds	6	Blasters and Converters

Nanomite Placement

Nanomites can appear on any nucleotides beyond and not including the third nucleotide pair from the left hand side of the screen (see Figure 10). The nucleotides where nanomites appear will be chosen at random.

Figure 10: Nanomite Placement



Nanomite Blasters

Each nanomite blaster will destroy the nucleotide it lands on, making that nucleotide disappear (turning it into a missing nucleotide). Thus, you must remember that nucleotide when your nanobot gets to it in order to replace it with the correct nucleotide. You can also check its partner on the opposite strand to figure out the correct nucleotide.

Nanomite Converters

Each nanomite converter will change one nucleotide to another nucleotide. When your nanobot gets to the altered nucleotide, it must change it back to the original. If the changed nucleotide is opposite a missing nucleotide, you must remember the original nucleotide in order to find the correct missing nucleotide.

Rewards

Points

The amount of points you accrue depends on the number of nucleotides Junior moves to and the number of correct nucleotides chosen before making an error (see Table 9).

Table 9: Point Accrual

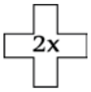
Number of Correct Nucleotides in a Row	Points per Nucleotide
1	1
5	2
10	3
20	5

NOTE: Point values are subject to adjustment.

Pickups

Pickups appear at certain intervals. Pickups can appear directly above a nucleotide in the bottom strand or below a nucleotide in the top strand. Pickups are locked to the nucleotides they appear above or below and will scroll with those nucleotides until picked up. To collect a pickup, you must guide Junior to the nucleotide directly above or below the pickup. Pickups will stay on screen until Junior moves to them or until they scroll off screen. Pickups are not required, so you can choose whether or not to guide Junior toward them (see Table 10).

Table 10: Pickups

Pickup	Pickup Concepts	What the pickup does	When the pickup appears
Score doubler/tripler		Doubles/triples the score for every nucleotide hit on beat until you make a mistake.	One score doubler/tripler appears every 15-20 seconds.
Tempo Changer		Changes the tempo to the number on the pickup for 10 seconds.	One tempo changer appears every 30 seconds.
Bonus nucleotides		If you pick up all four bonus nucleotides, a replacement nanobot will take your place in guiding Junior, repairing the DNA sequence with 100% accuracy for 5 seconds.	One bonus nucleotide (A, T, G, or C) will appear every 5-12 seconds.

Game Modes

Overview

This section includes details about the main run-forever game mode of *Double Helix*.

Single-Player Game

The single-player game is *Double Helix's* only game mode. You guide your nanobot as far as possible along the strands of DNA in order to achieve a high score. Since *Double Helix* is a run-forever, it consists of only one level whose difficulty ramps based on your inputs.

Story

You play as Junior, a nanobot responsible for traveling along the strands of DNA within a single human cell to check for replication errors in an effort to preserve the human genome before the world ends.

Story Breakdown

Along the way, you must guide Junior to correct errors caused by nanomite invaders and to replace missing nucleotides in the sequence in an effort to preserve the human genome before a meteorite hits the Earth. This simple backstory is communicated via a skippable cut-scene no longer than 10 seconds in length that plays when you hit the play button in the main menu.

Single-Player Time Projections

Each game starts at a base tempo of 60bpm. Depending on the number of errors you make, each game could last from 3 seconds (one error per second) through several minutes. See Table 11 for time projection scenarios.

Table 11: Time Projection Scenarios

Projected Player Errors	Time at which tempo increases from 60bpm to 80bpm	Time at which tempo Increases from 80bpm to 100bpm	Time at which tempo increases from 100bpm to 120 bpm	Total Game Time
0	25 seconds after game starts	50 seconds after game starts	75 seconds after game starts	Game progresses beyond 1:15
1 every 30 seconds	35 seconds after game starts	70 seconds after game starts	105 seconds after game starts	Game ends before 1:30

Projected Player Errors	Time at which tempo increases from 60bpm to 80bpm	Time at which tempo Increases from 80bpm to 100bpm	Time at which tempo increases from 100bpm to 120 bpm	Total Game Time
1 every 10 seconds	Never reached	Never reached	Never reached	Game ends within 30 seconds

Single-Player Beat Chart

The projected beat chart below shows the beats of the game in three sections based on the three tempo ranges within the game (60-80bpm, 80-100bpm, and 100-120bpm). Different beats are based on availability of pickups, nanomite invasions, and predicted player learning curve (see Figure 11).

Figure 11: Projected Gameplay Beats

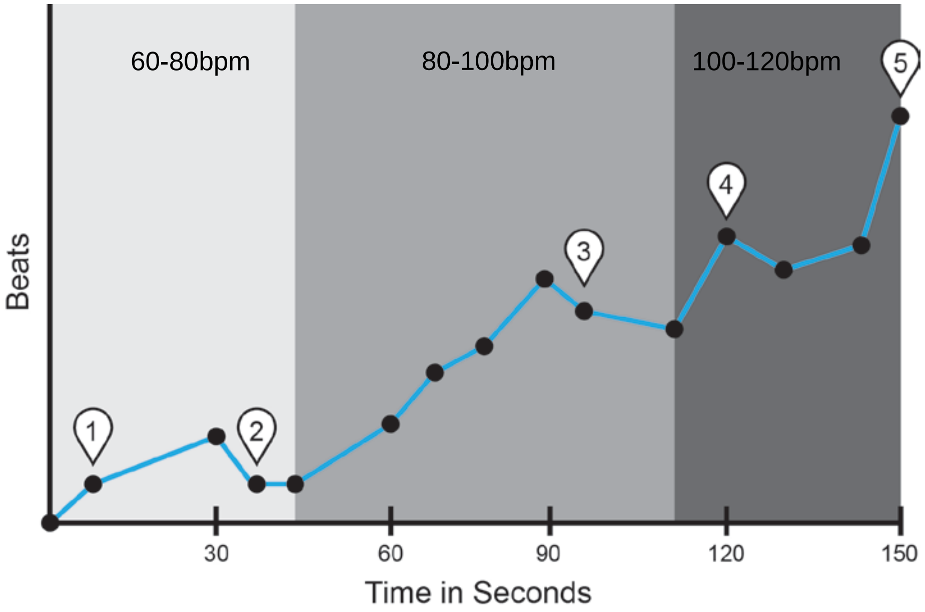


Table 12: Stages Breakdown

Beat	Gameplay Elements
1	First nanomites appear
2	Four bonus nucleotides collected; replacement nanobot takes over briefly
3	Tempo changer pickup decreases tempo briefly
4	Margin of error decreases make errors harder to avoid
5	Final error is made

Victory Condition

The goal of *Double Helix* is to attain the highest score you can by guiding Junior as far along the double helix as possible before making three errors. Thus, victory is based on surpassing your previous performances. Game score, play time, and highest bpm achieved will be displayed after each game (see Results Screen in Appendix C). High scores and game statistics are shown in the scores and stats screen (see Scores and Stats Screen in Appendix C).

Game World

Overview

This section contains a description of Micropolis, the game world within *Double Helix.*

Micropolis

Overview

This topic provides a description of the key components of the world including scale and travel.

Scale

Micropolis represents the nucleus of a single human cell. Within that nucleus, Junior (1 nanometer in length and width) moves along a double helix that measures 4 nanometers in width; in other words, the distance between the two strands is 2 nanometers.

Travel

Junior will move along the DNA double helix within Micropolis by following the nucleotide commands you provide. Since the DNA strands within a single cell can contain over 200 million nucleotide pairs, Junior will always remain within Micropolis and will not move to any other cells.

Stages

Overview

This section details the stages of the game based on increasing tempos.

Stages Breakdown

The game starts at a tempo of 60bpm. As Junior progresses, tempo increases (see Table 9).

Table 13: Stages Breakdown

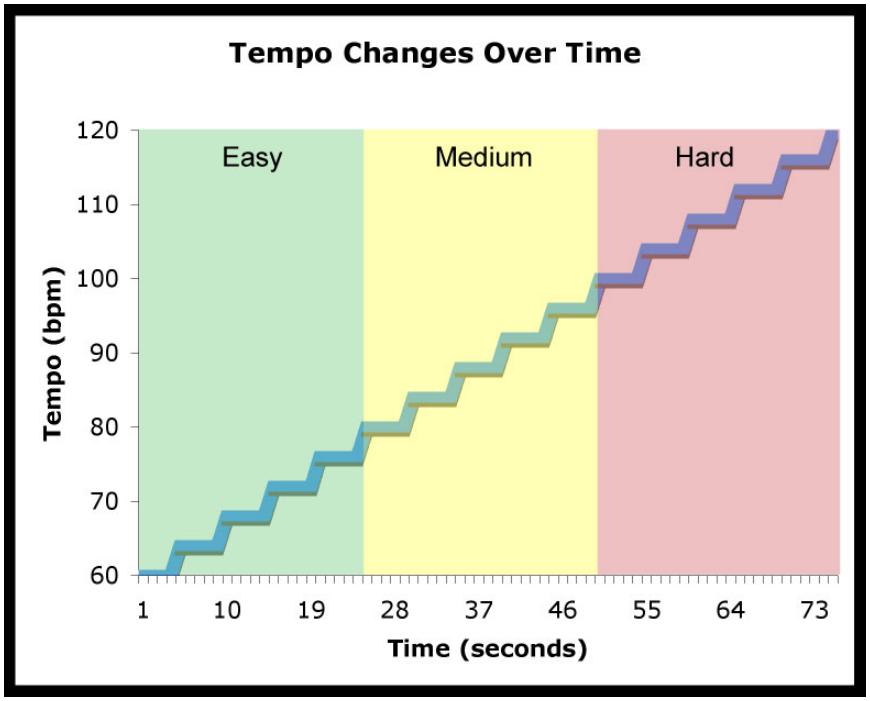
Stage	Tempo Range	Margin of Error (__ milliseconds within the beat)
Easy	60-79bpm	500
Medium	80-99bpm	300
Hard	100-120bpm	200

Stage Progression

If you make no errors, you will progress from the 60bpm starting tempo to the 120bpm maximum tempo in 75 seconds (see Figure 12). The music will speed up or slow down according to the current tempo (see Appendix B for more details). Tempo progression will occur as follows:

- Every five seconds you go without any errors, the tempo will increase by 4bpm.
- Each error you make will result in a tempo decrease of 8bpm.

Figure 12: Tempo Progression



Game Progression

Overview

This section includes details about progression through the run-forever environment of *Double Helix.*

Game Completion

You complete the game when you make your third error. Depending on your performance, each game could last from 3 seconds through several minutes.

Progression Flow

Progression through the game correlates with increasing tempo. The more accurate your performance, the faster the tempo increases, leading to increased difficulty through more missing nucleotides and nanomite invasions (see Table 14).

Table 14: Game Progression

Tempo Range	Missing Nucleotides on Screen	Nanomite Invasions
60-80bpm	4	• 3 nanomite blasters every 5 seconds
80-100bpm	7	• 4 nanomite blasters and converters every 3 seconds
100-120bpm	10	• 6 nanomite blasters and converters every 3 seconds

Appendix A – Art Guidelines

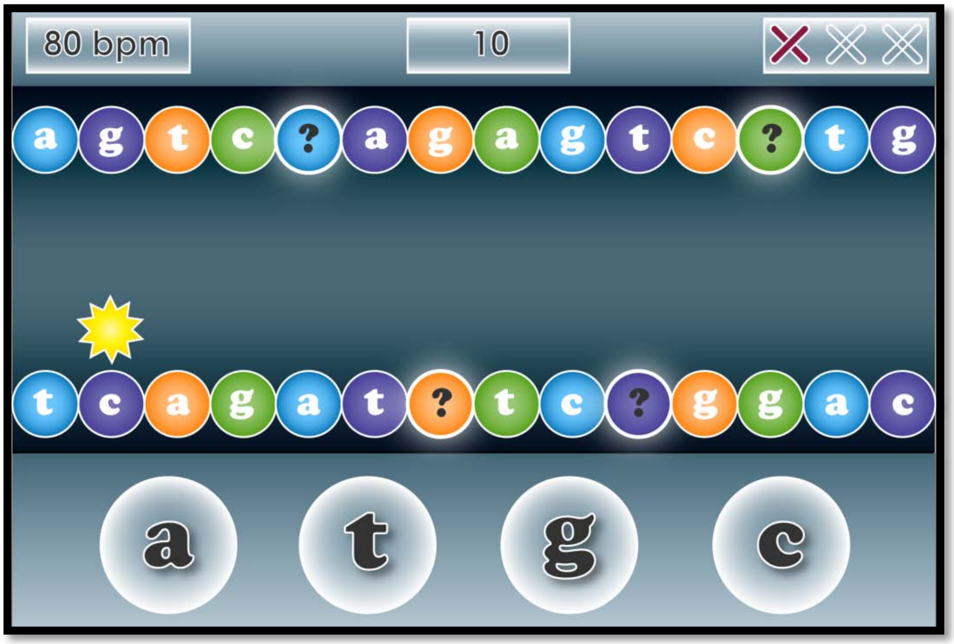
Overview

This section includes an overview of the art philosophy, as well as some sample images.

Art Design

The art for *Double Helix* should consist of relatively simple graphics with a straightforward interface that's easy to navigate. Graphics should be colorful and should consist of simple shapes. The nucleotide colors should clearly show nucleotide pairs. For example, two nucleotides that form a pair should be the same color (see Figure 13).

Figure 13: Art Design



Colorful Graphics

Game screens should utilize bright colors with simple gradients to give the game a stylized look. Nucleotide pairs (nucleotides directly opposite each other) should share the same color.

Simple Shapes

The main game screen should represent Junior and the DNA strand using simple shapes. The DNA strands in game do not need to depict DNA strands with perfect accuracy, but the shapes should provide a basic idea of the relationships between different nucleotides within a cell (i.e., that nucleotide pairs face each other and are a certain distance apart).

Intuitive Interface

The interface elements should be simple to read and easy to use. The nucleotide buttons along the bottom of the screen should be at least ½ inch in diameter (see Appendix C).

Appendix B – Audio Guidelines

Overview

This section includes the audio philosophy and guidelines for *Double Helix*.

Sound Design

The audio assets will consist largely of a range of upbeat instrumental tunes, along with cartoon sounds pertaining to player feedback.

Character Audio

Character audio consists of cartoon sound effects that provide feedback for correct and incorrect inputs. For example, tapping the correct nucleotide button for a missing nucleotide spot at the right time might trigger a “ding!” sound, while pressing an incorrect button might trigger a buzzer. No audio is needed for Junior’s movement.

Enemy Audio

Enemy audio also consists largely of cartoon sound effects. Enemies will need a warning sound to signal their arrival. The warning sound for nanomite blasters should differ from the warning sound for nanomite converters. Enemies also need a sound to signal their exit; this can be the same for both enemy types.

Pickup Audio

Additional cartoon sound effects will be needed for each type of pickup. Each pickup requires a unique alert sound to signal its appearance, as well as a sound to signify that Junior has picked it up.

Music

Music should consist of a bank of 20 looping instrumental rags whose original tempos range from 80-100 bpm so that the tunes can be scaled from 60 to 120 bpm. Music should not have lyrics, as distortion would be too noticeable when the music slows or speeds up. For each game, the tune should be chosen at random. The same tune should not be used for two games in a row.

Front End

The front end music should consist of an upbeat instrumental tune similar to in-game music. The tune should be in the 60-80bpm tempo range and should loop. The length of the loop should be short – no more than 20 seconds. The music should continue to loop through all the front end screens, regardless of the buttons you press and the front end screens to which you navigate. The buttons also need sound effects for their transition on screen (e.g., a swoosh in sound), transition off screen (e.g., a swoosh out sound), and pressed state (e.g., a click sound).

Appendix C – Interface Wireframes

Overview

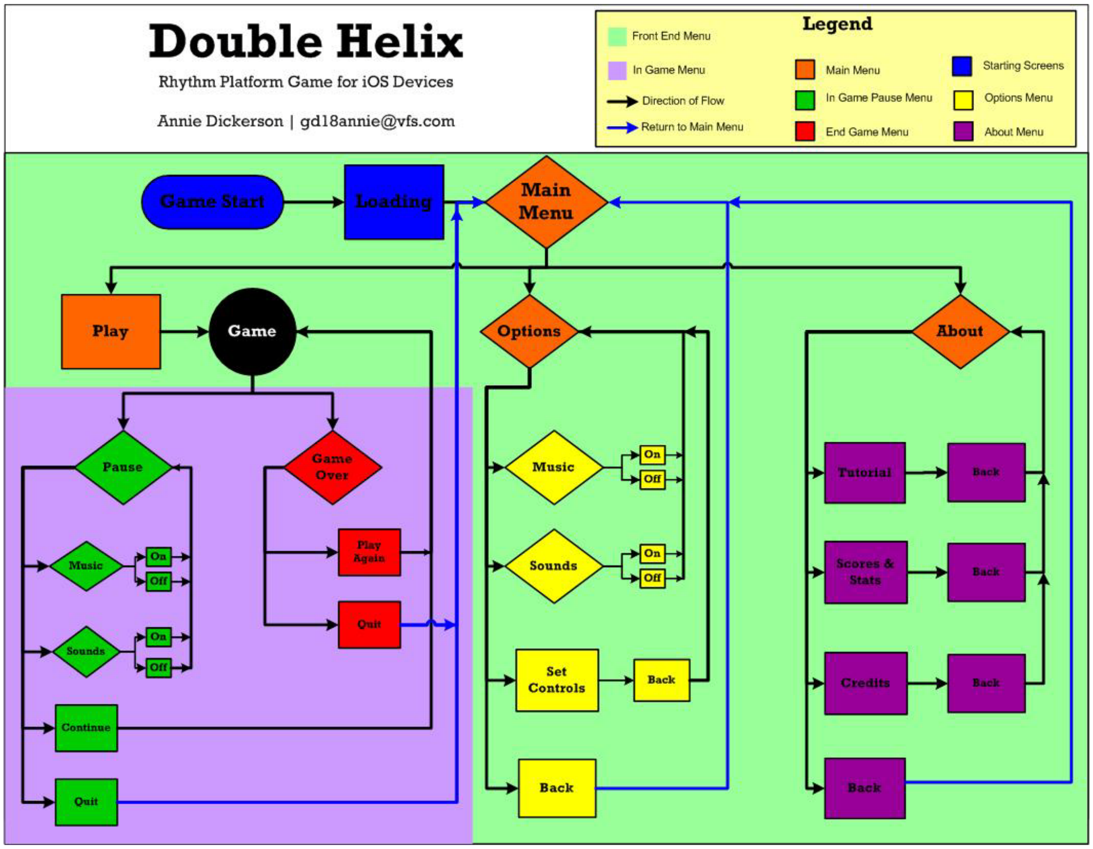
This section includes wireframes for the in-game HUD and menu screens for *Double Helix.*

NOTE:	The style, shape, and layout of the elements presented here, is merely a guide. The actual element design will be determined as they are created, and by the art style as well.

Single Player Wireframes

The single player screens begin with a loading screen, followed by a main menu. From there, you can choose to play the game, adjust settings in the options menu, or learn more about the game in the about menu (see Figure 14).

Figure 14: Screen Flow



Menu Screens

Main Menu

The main menu allows you to play the game or navigate to an options menu to adjust settings or to an about menu to learn more about the game.

Figure 15: Main Menu Screen

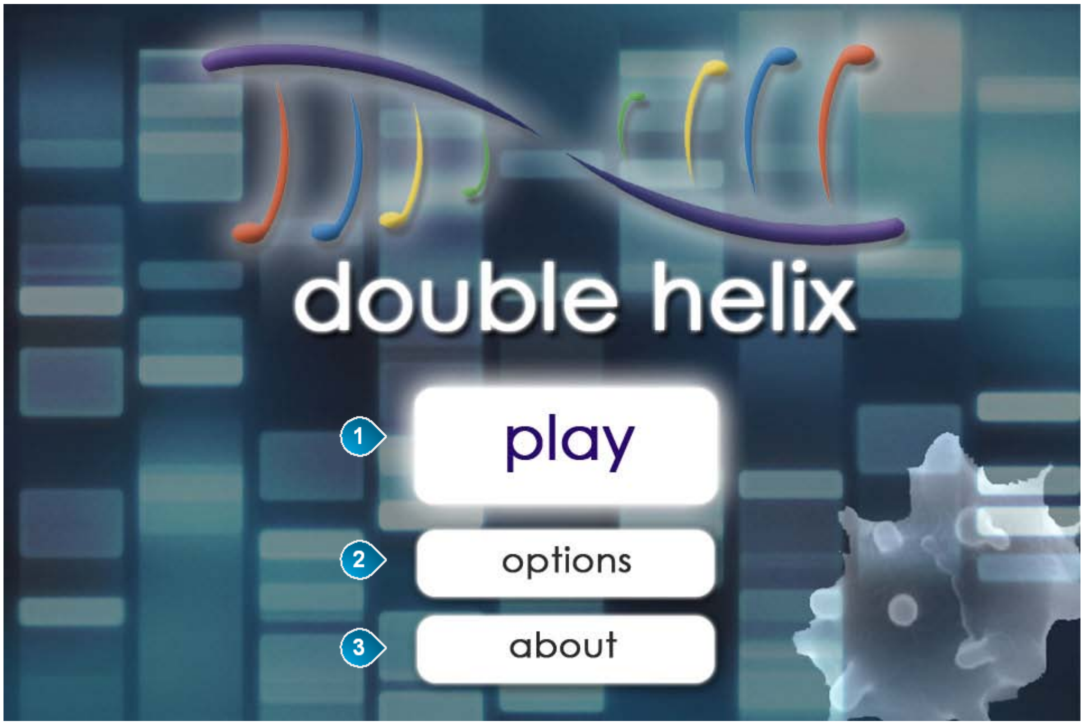


Table 15: Main Menu Screen Components

Item	Name	Details
1	Play	• Pressing the play button will begin the game. This screen shows the play button in an 'on' state.
2	Options	• The options button will lead you to a menu to adjust game settings (see Figure 15).
3	About	• The about button will lead you to a menu with options to learn more about the game (see Figure 16).

Options Menu

The options menu allows you to adjust music and sound settings, as well as arrange the nucleotide buttons in any order. The default order is A, T, G, C; the buttons can be rearranged in any order (e.g., T, C, G, A).

Figure 16: Options Menu

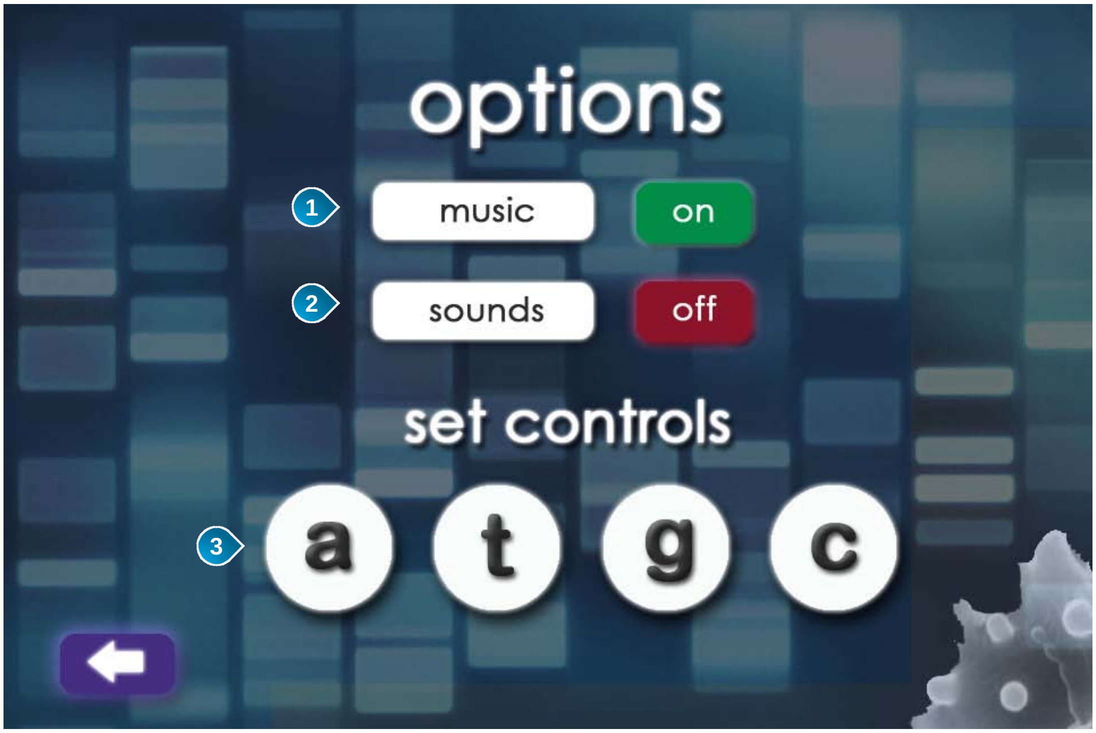


Table 16: Options Menu Components

Item	Name	Details
1	Music	• Pressing the button to the right of the music tab toggles the music on or off.
2	Sounds	• Pressing the button to the right of the sounds tab toggles the sound effects on or off.
3	Set Controls	• You can drag the four nucleotide buttons into any order you want.

About Menu

The about menu allows you to access a tutorial screen, scores and statistics, and credits.

Figure 17: About Menu

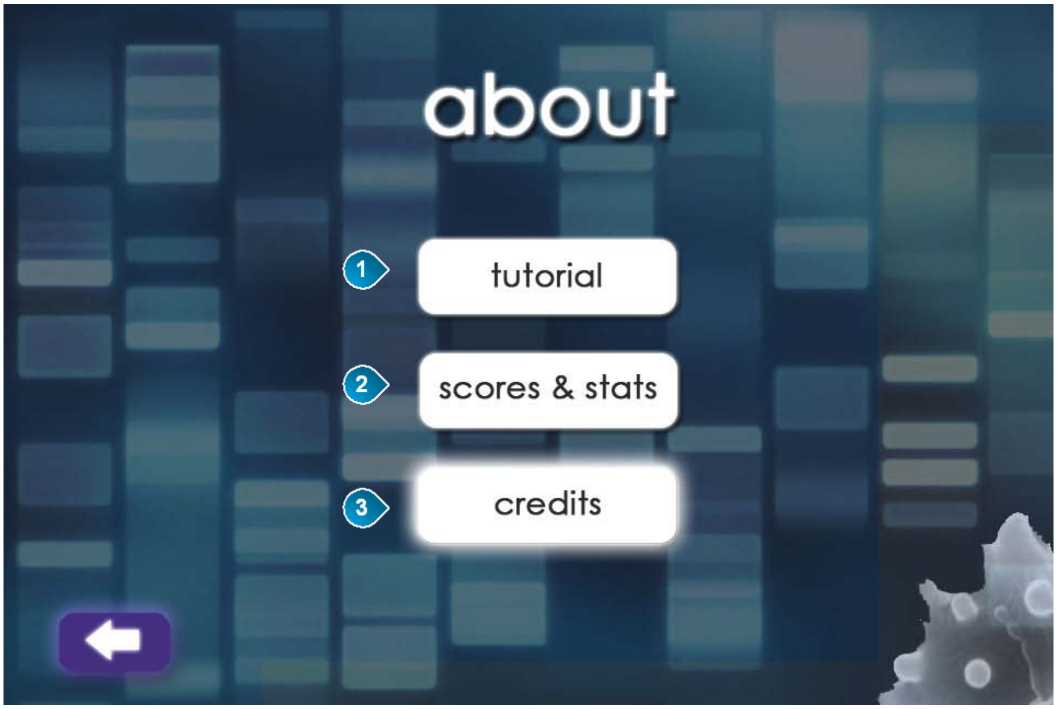


Table 17: About Menu Components

Item	Name	Details
1	Tutorial	• The tutorial button will open a simple tutorial with three animated images showing gameplay basics.
2	Scores & Stats	• Scores and stats will lead open a screen showing your high scores and gameplay statistics.
3	Credits	• The credits screen will include a list of game developers and contributors. This screen shows the credits in an 'on' state.

Tutorial Screen

The tutorial screen shows four panels with short text descriptions and basic animations demonstrating how to play the game.

Figure 18: Tutorial Screen

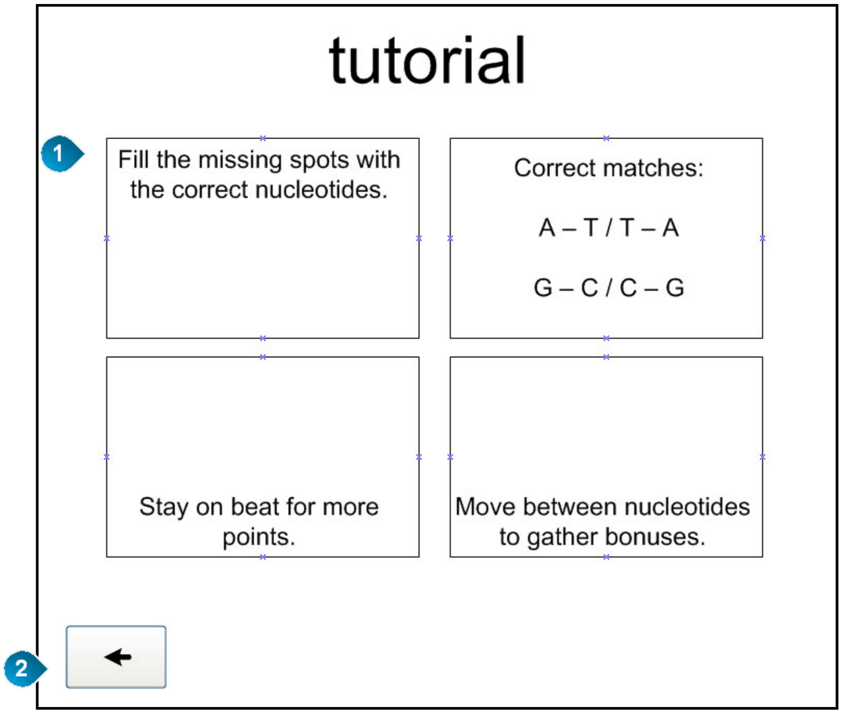

tutorial
1
Fill the missing spots with the correct nucleotides.
Correct matches:
A – T / T – A
G – C / C – G
Stay on beat for more points.
Move between nucleotides to gather bonuses.
2
←

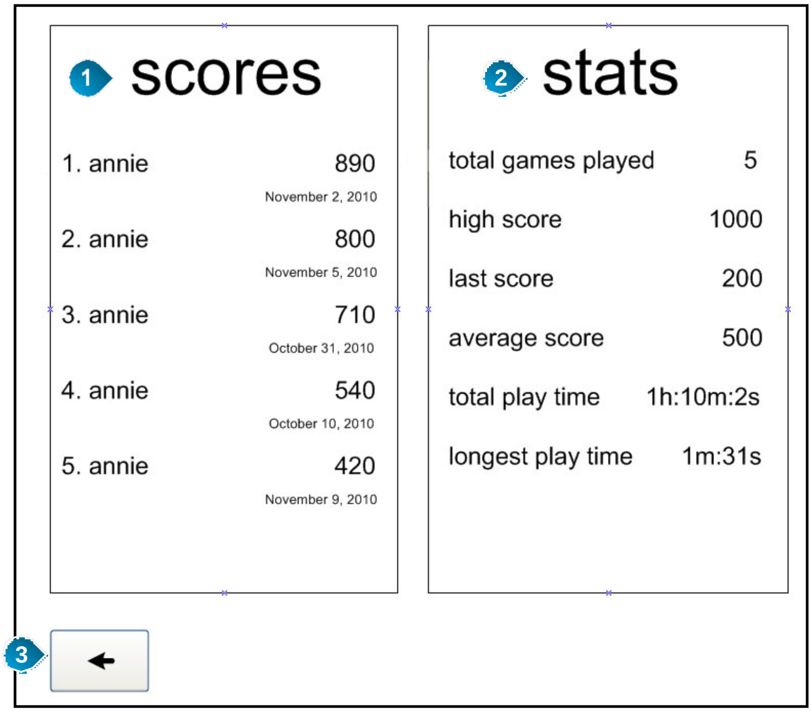
Table 18: Tutorial Screen Components

Item	Name	Details
1	Tutorial Panels	• The tutorial screen includes four panels with short text descriptions and basic animations demonstrating gameplay.
2	Back	• The back button takes you back to the about menu.

Scores & Stats Screen

The scores and stats screen includes a list of scores and statistics from previous games you've played.

Figure 19: Scores and Stats Screen



scores

1. annie	890	November 2, 2010
2. annie	800	November 5, 2010
3. annie	710	October 31, 2010
4. annie	540	October 10, 2010
5. annie	420	November 9, 2010

stats

total games played	5
high score	1000
last score	200
average score	500
total play time	1h:10m:2s
longest play time	1m:31s

Table 19: Scores & Stats Screen Components

Item	Name	Details
1	High Scores List	• The high scores list consists of the top 5 high scores, along with player name and date each game was played.
2	Stats List	• The stats list includes metadata about the games played, scores achieved, and play times.
3	Back	• The back button takes you back to the about menu.

Game Screens

HUD

The in-game HUD includes a tally of errors made, the current tempo, and points accrued.

Figure 20: HUD

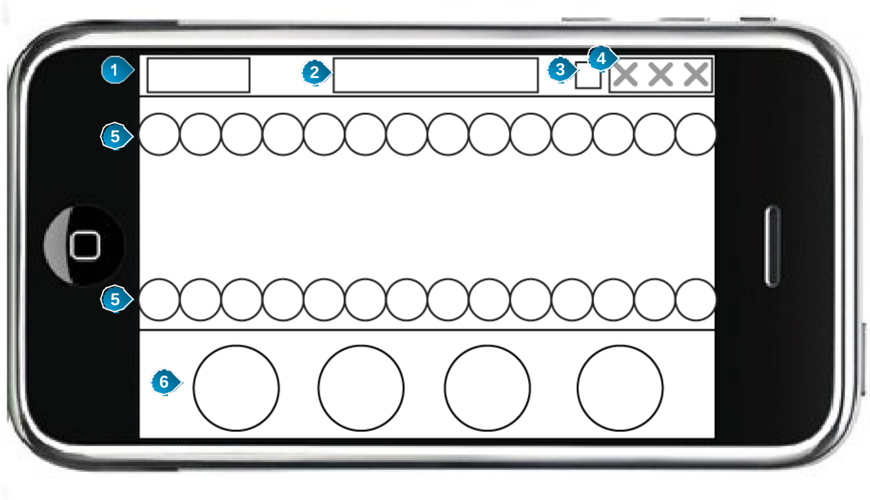


Table 20: HUD Components

Item	Name	Details
1	Current Tempo	• The current tempo ranges from 60bpm at the beginning of the game up to 120bpm, depending on your performance. The tempo box will flash white on each beat.
2	Score	• The score will increase based on the nucleotides you choose correctly.
3	Pause	• The pause button will pause the game and bring up the pause menu.
4	Errors	• The game ends when you make three errors.
5	DNA Strands	• The screen will display fourteen nucleotides in each of the two DNA strands.
6	Nucleotide Buttons	• You will tap the four nucleotide buttons to guide Junior to different nucleotides along the DNA strands. These buttons should be ~1/2 inch in diameter.

Pause Screen

The pause screen is a semi-transparent overlay that comes up when you press the pause button during the game. The pause screen allows you to toggle sound settings, restart the game, or exit to the main menu.

Figure 21: Pause Screen

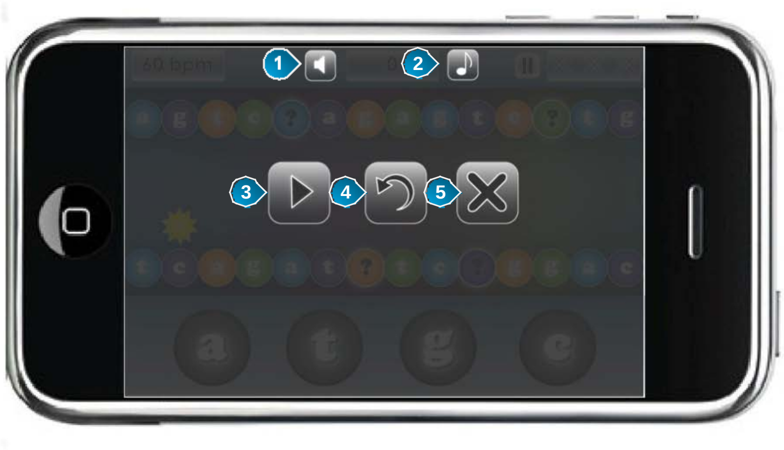


Table 21: Pause Screen Components

Item	Name	Details
1	Sounds	- You can use the sounds button to toggle sound effects on and off.
2	Music	- You can use the music button to toggle music on and off. If you toggle the music off, the current tempo in the HUD will flash with the beat.
3	Continue	- The continue button allows you to continue the game.
4	Restart	- The restart button allows you to start over with a score of 0 and a tempo of 60bpm.
5	Exit	- The exit button will take you to the main menu.

Results Screen

The game ends when you make you've made three errors, thus bringing up the results screen showing your score, gameplay time, and ending BPM.

Figure 22: Results Screen

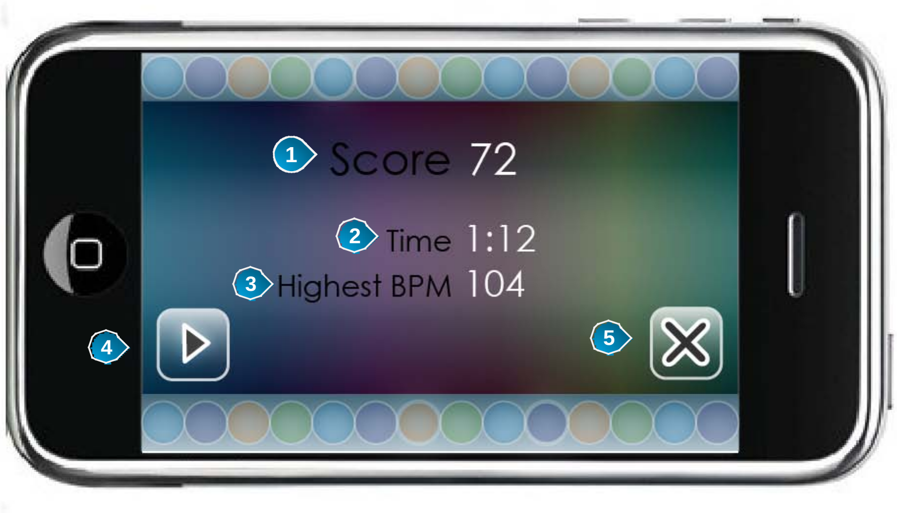


Table 22: Results Screen Components

Item	Name	Details
1	Score	• At the top of the screen is your final score.
2	Time	• The time displays the minutes and seconds that you lasted in the game.
3	Ending BPM	• The ending BPM shows the highest BPM you achieved during the game.
4	Play again	• The play again button allows you to start another game.
5	Exit	• The exit button takes you back to the main menu.

Appendix E – Competitive Analysis

Overview

This section includes a competitive analysis comparing *Double Helix* to other rhythm games currently on the market.

Double Helix Competitive Analysis

Table 23: Competitor Analysis

Competitor	Key Features	*Double Helix*'s Comparable Features	*Double Helix*'s Strengths
Bit.Trip Runner (WiiWare)	• Guide character through various obstacles by performing actions on the beat • Each dodged obstacle produces a musical note that runs in time with the background music • Collect gold bars along the way to earn more points	• Guide Junior along the DNA strands • Navigate to missing nucleotides and fix replication errors • Collect pickups along the way for bonuses	• Real world theme that ties in scientific principles • Multiple player paths allow for more choice • Multiple types of pickups and enemies force you to think and react quickly
Maestro! Jump in Music (Nintendo DS)	• Strum or pluck to guide character through various obstacles in time to the music • Enemies must be tapped in time to the music • Several correct actions in a row yields a bonus	• Guide Junior to nucleotides in time to the music • Movement must be in time with the music • Collecting pickups yields bonuses	• Multiple player paths allow for more choice • Must use nucleotide pairs to figure out missing nucleotides for correct navigation
NinJump (iOS)	• Guide character along two paths while avoiding obstacles • Can choose when to switch to the opposite side • Time jumps correctly to slice enemies	• Two paths are available for Junior to traverse • You can choose which DNA strand to traverse and when to switch • Movement must be in time with the music	• Can choose different paths for Junior • Increasing tempo challenges you to make faster decisions • You can make up to three errors before the game ends

Bit.Trip Runner

Bit.Trip Runner is the fourth installment of the Bit.Trip series, created for WiiWare. Basic gameplay involves controlling Commander Video by making him jump, slide, and kick through various obstacles in his path while collecting gold bars.

Maestro! Jump in Music

Created for Nintendo DS, *Maestro! Jump in Music* features Presto, who walks across a line of strings. You must guide Presto through various obstacles by plucking or strumming the strings in time to the music.

NinJump

NinJump is a casual run-forever game for iOS devices in which you play as a ninja trying to climb as high as possible while dodging multiple enemies along the way. Running into the enemies will cause you to fall, while timing your jumps correctly will allow you to slice them.