Project Number: MBJ GG14

VIRTUAL DR. NIM

A Major Qualifying Project Report

Submitted to the Faculty of the Worcester Polytechnic Institute

In partial fulfillment of the requirements for the Degree of Bachelor of Science

In Interactive Media and Game Development

By Ryan Melville

Advisor: Professor Brian Moriarty

*This report represents the work of WPI undergraduate students submitted to the faculty
as evidence of completion of a degree requirement.
WPI routinely publishes these reports on its website without editorial or peer review.
For more information about the projects program at WPI, please see
http://www.wpi.edu/academics/ugradstudies/project-learning.html*

ABSTRACT

The goal of this project was to develop a virtual recreation of Dr. Nim, the first consumer product marketed as a “computer game.” The application was developed using a combination of programs, including the Unity game engine, Autodesk 3DS Max, and Adobe Illustrator. Virtual Dr. Nim simulates the behavior of the device through the use of physics-based gameplay in tandem with the scripted movement of components and synchronized sound effects, recorded from an original specimen of the toy.

TABLE OF CONTENTS

INTRODUCTION

Virtual Dr. Nim is a digital recreation of the original Dr. Nim, a plastic toy marketed in the 1960s as a “computer game.” The goal of this project was to develop a faithful and easily distributable adaptation of this classic device for use as entertainment or as an educational tool.

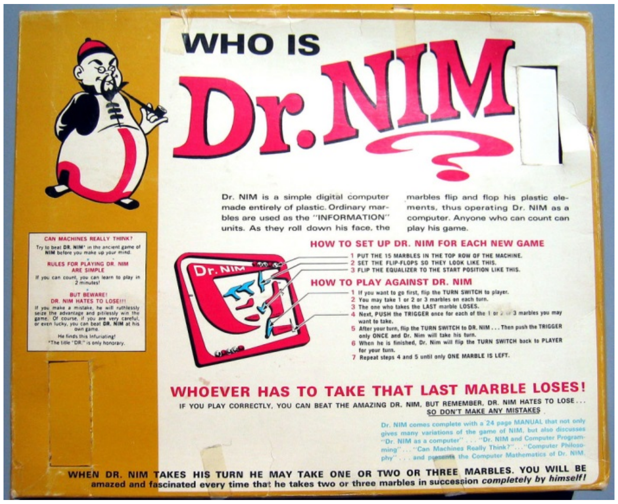
Dr. Nim challenges a player to beat the “computer” in a version of the classic game of Nim. The game’s logic is implemented via an ingenious system of mechanical flip-flops actuated by rolling marbles. The player and Dr. Nim alternate turns, releasing one, two or three marbles from an initial queue of fifteen. In the standard game, the player left with the last marble loses. Other game variants use a different number of marbles, different starting positions, and even a different win condition: be the player to take the last marble.

The player can only take two actions throughout the course of the game; they may choose to release marbles from the queue at the top of the board, or hand over the turn by manually releasing Dr. Nim’s first marble. The position of the flip-flops on the game board determines the route marbles will take once released. On Dr. Nim’s turn, if certain paths are taken, a marble may trigger the automatic release of another marble, thus representing its decision as an AI player to continue playing.

PROJECT CONCEPTION

The idea of this project came about in a meeting with Professor Brian Moriarty, while considering possible options for a Major Qualifying Project to be completed in only one term of academic study. I was introduced to Dr. Nim for the first time in his office, where he showed me his physical specimen of the toy and told of his ambition to virtually recreate some of the primitive “computer games” produced during the 1950s and ‘60s.

Dr. Nim was chosen for its modest complexity compared to other candidates. The engineering and asset creation requirements seemed to fit well within the five-week period in which it would have to be developed.



Dr. NIM is a simple digital computer made entirely of plastic. Ordinary marbles are used as the "INFORMATION" units. As they roll down his face, the marbles flip and flop his plastic elements, thus operating Dr. NIM as a computer. Anyone who can count can play his game.

CAN MACHINES REALLY THINK?

Try to beat DR. NIM* in the ancient game of NIM before you make up your mind.

RULES FOR PLAYING DR. NIM ARE SIMPLE

If you can count, you can learn to play in 2 minutes!

BUT BEWARE! DR. NIM HATES TO LOSE!!!

If you make a mistake, he will ruthlessly seize the advantage and pitilessly win the game. Of course, if you are very careful, or even lucky, you can beat DR. NIM at his own game.

He finds this infuriating!

*The title "DR." is only honorary.

HOW TO SET UP DR. NIM FOR EACH NEW GAME

1 PUT THE 15 MARBLES IN THE TOP ROW OF THE MACHINE.
2 SET THE FLIP-FLOPS SO THEY LOOK LIKE THIS.
3 FLIP THE EQUALIZER TO THE START POSITION LIKE THIS.

HOW TO PLAY AGAINST DR. NIM

1 If you want to go first, flip the TURN SWITCH to player.
2 You may take 1 or 2 or 3 marbles on each turn.
3 The one who takes the LAST marble LOSES.
4 Next, PUSH the TRIGGER once for each of the 1 or 2 or 3 marbles you may want to take.
5 After your turn, flip the TURN SWITCH to DR. NIM . . . Then push the TRIGGER only ONCE and Dr. Nim will take his turn.
6 When he is finished, Dr. Nim will flip the TURN SWITCH back to PLAYER for your turn.
7 Repeat steps 4 and 5 until only ONE MARBLE IS LEFT.

WHOEVER HAS TO TAKE THAT LAST MARBLE LOSES!

IF YOU PLAY CORRECTLY, YOU CAN BEAT THE AMAZING DR. NIM, BUT REMEMBER, DR. NIM HATES TO LOSE . . . SO DON'T MAKE ANY MISTAKES

Dr. NIM comes complete with a 24 page MANUAL that not only gives many variations of the game of NIM, but also discusses "Dr. NIM as a computer" . . . "Dr. NIM and Computer Programming" . . . "Can Machines Really Think?" . . . "Computer Philosophy" . . . and presents the Computer Mathematics of Dr. NIM.

WHEN DR. NIM TAKES HIS TURN HE MAY TAKE ONE OR TWO OR THREE MARBLES. YOU WILL BE amazed and fascinated every time that he takes two or three marbles in succession *completely by himself!*

Figure 1. Rear of packaging

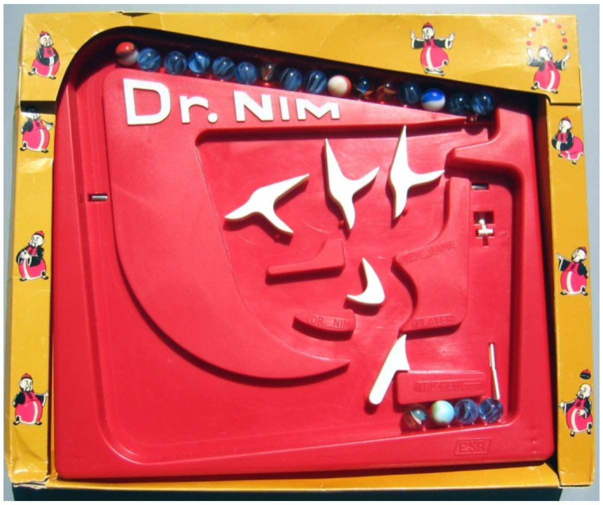


Figure 2. Inside packaging

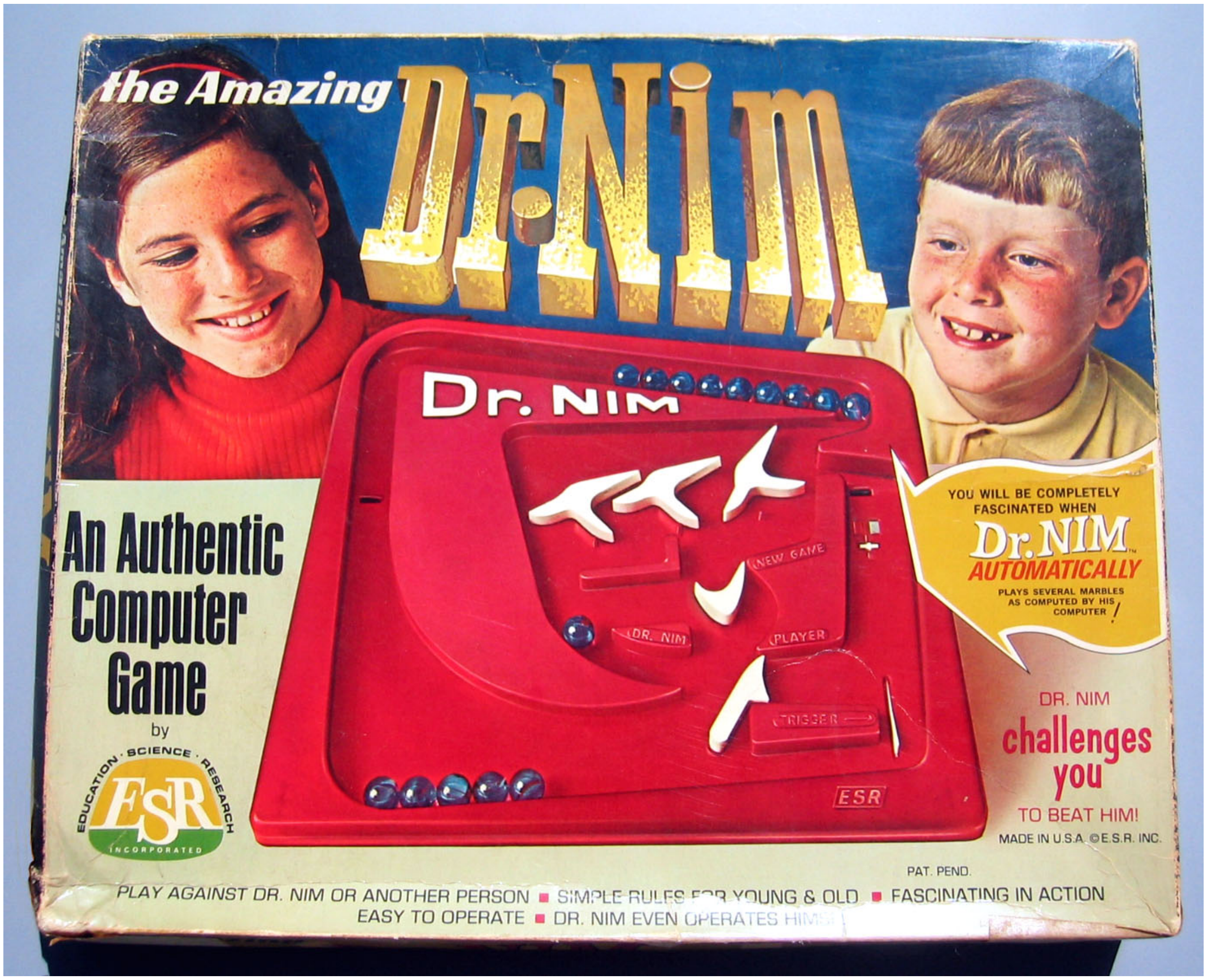

the Amazing
Dr. Nim
Dr. NIM
NEW GAME
DR. NIM
PLAYER
TRIGGER
ESR
An Authentic Computer Game
by
EDUCATION · SCIENCE · RESEARCH
ESR
INCORPORATED
YOU WILL BE COMPLETELY FASCINATED WHEN
Dr. NIM
AUTOMATICALLY
PLAYS SEVERAL MARBLES AS COMPUTED BY HIS COMPUTER!
DR. NIM
challenges you
TO BEAT HIM!
MADE IN U.S.A. ©E.S.R. INC.
PAT. PEND.
PLAY AGAINST DR. NIM OR ANOTHER PERSON ■ SIMPLE RULES FOR YOUNG & OLD ■ FASCINATING IN ACTION
EASY TO OPERATE ■ DR. NIM EVEN OPERATES HIM

Figure 3. Original retail packaging of Dr. Nim (1966).

HISTORY OF DR. NIM

Dr. Nim was a plastic toy invented by John Thomas Godfrey and published by ESR Inc. in 1966. It was the first mass-market consumer product explicitly marketed as a “computer game,” and was generally sold in toy stores, with a list price of $3.50. The box contained a plastic board with five movable flip-flops, a lever, fifteen marbles and an instruction booklet.

CHOOSING THE GAME ENGINE

As this project was meant to be a faithful recreation of the classic toy, it was decided that the completed product would be a three dimensional video game, therefore, a game engine with 3D capability was necessary. I narrowed the decision down to two options: Unreal Engine 4 or Unity 5.

There were many things to consider when making my final decision. I have had almost equal experience with both engines, having used each in previous coursework, but I had to consider the relative ease of which I could research anything I was unfamiliar with. Unreal Engine 4 was released less than a year prior to starting this project (March 2014), and many of the engine's features were significantly different from its long-standing predecessor Unreal Engine 3. Unreal 4 is also still in heavy development, with new features and updates being released on a monthly basis, so documentation is sparse and preliminary.

By contrast, Unity 5 is an extension of the existing Unity 4 engine. It is extensively documented through video tutorials and online forums. It also provides support for a wide range of target devices, including most popular consoles and mobile devices, as well as an HTML5 player which would allow Virtual Dr. Nim to be hosted on any website.

I was also more acquainted with the art pipeline for Unity projects. This was an important factor in my decision-making process due to my desire to have the ability to quickly adjust any dimension on the 3D models and re-import them into the game engine with ease. The scale and shape of the game components were likely to require many adjustments, and it seemed important to minimize complications in the art pipeline.

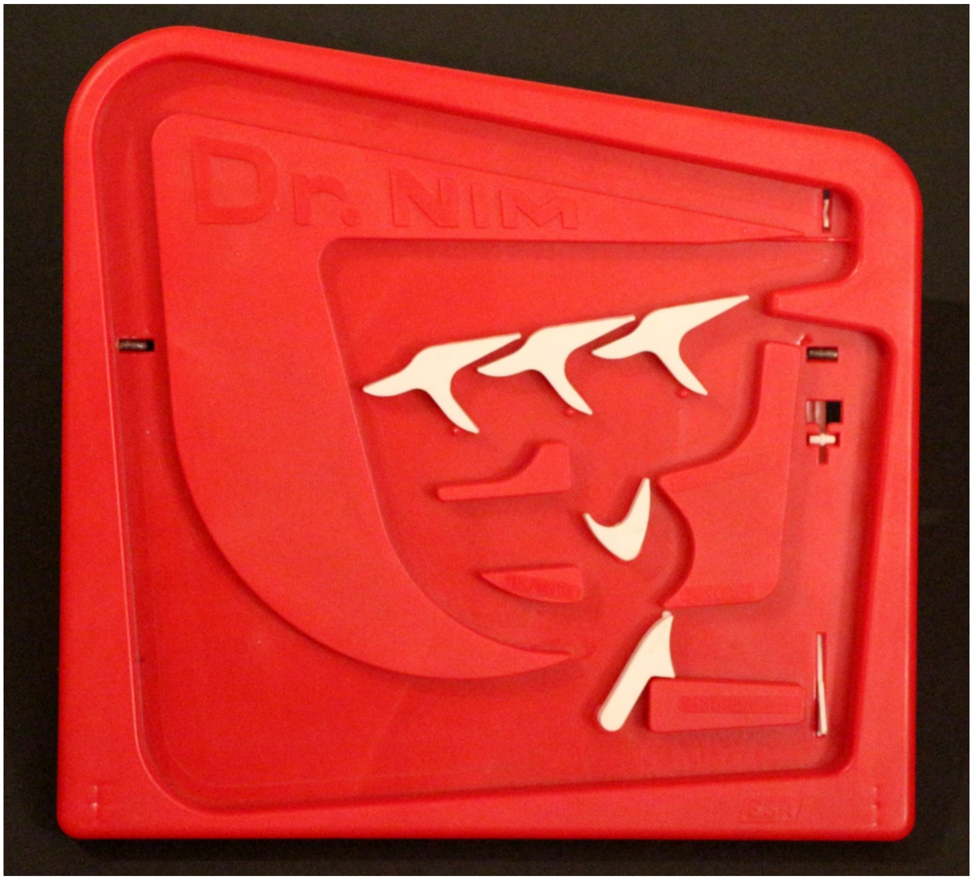
ART PIPELINE

As a technical Interactive Media & Game Development major, being responsible for the artistic side of this project was a new challenge for me. While familiar with the steps necessary for importing art into a Unity project, I have never needed to develop the 3D models myself. I developed an efficient three-step process for creating the three dimensional art assets and importing them into Unity. This process required the use of two additional programs, as well as scans of the physical version of Dr. Nim owned by Professor Moriarty.

INITIAL SCANS

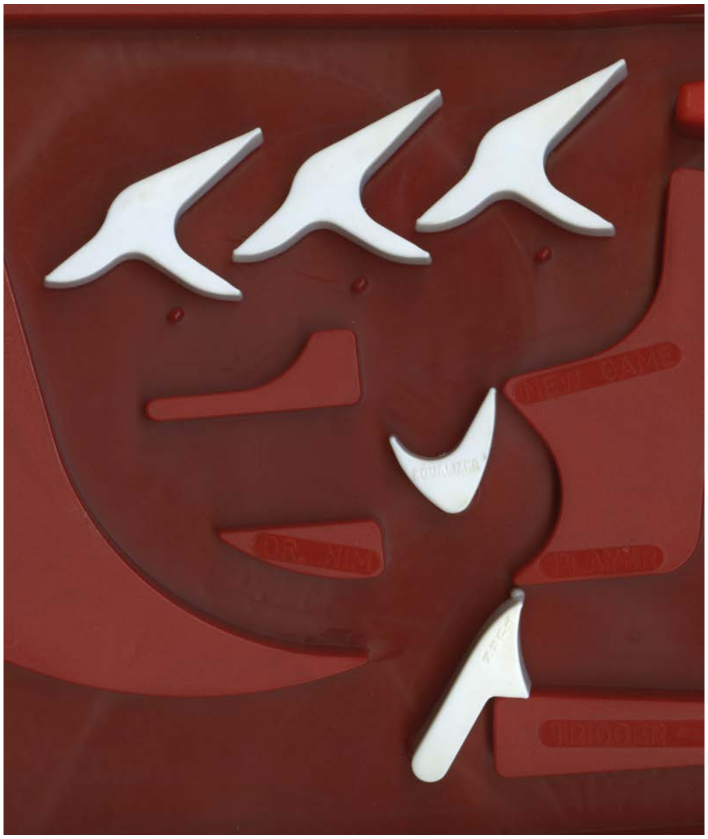
In order to make the 3D models of Dr. Nim as faithful to the original as possible, I utilized scans of the actual game board as a base layer for my modeling. Professor Moriarty provided me with high quality scans of his physical copy of the game. As well as being used to design the parts of the game board, these images were used as the reference when assembling the 3D models inside the Unity game engine.

Two high-resolution scans were taken in the first week of project development. One provided a direct, view from above, image of the Dr. Nim specimen. The other scan provided a more focused image of the switches on the game board. Both scans had their uses, and were crucial in the development of Virtual Dr. Nim, especially when the actual, physical specimen was not available for direct study.



SCAN 1

The image above is a full scan of the actual Dr. Nim specimen. This scan was used for modeling the entirety of the red-plastic game board and as a reference for the relative positions of the game’s movable pieces. It was also used as the main reference image for the extent of the project development, when the actual specimen was not at hand. This image provided the best example of a “minimum viable product” as a goal for the project development scope.

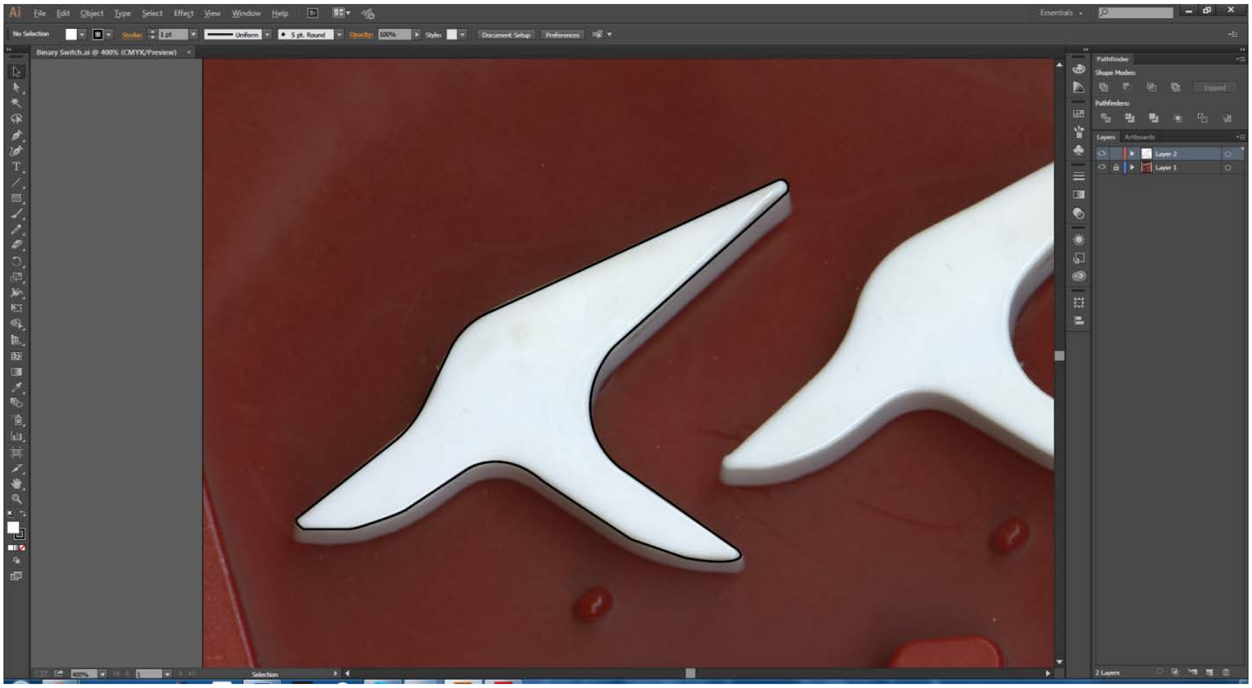


SCAN 2

The second scan was made to emphasize the moving pieces on the game board. By using this image as a base, I was then able to accurately model the contours of each of the white flip-flops. This scan also provided a visual aid when developing the User Interface text, as it depicts the text on the game board very legibly.

ADOBE ILLUSTRATOR

Once I had Professor Moriarty's scans in my possession, I imported them into Adobe Illustrator. I was then able to use Illustrator's pen tool to trace directly on top of the images, creating an accurate two dimensional profile of whichever part I required for my model, while containing it all within a single layer of the composition. Once complete, I then exported this layer as a .dwg file. This file could then be imported into Autodesk 3DS Max to continue the modelling process.

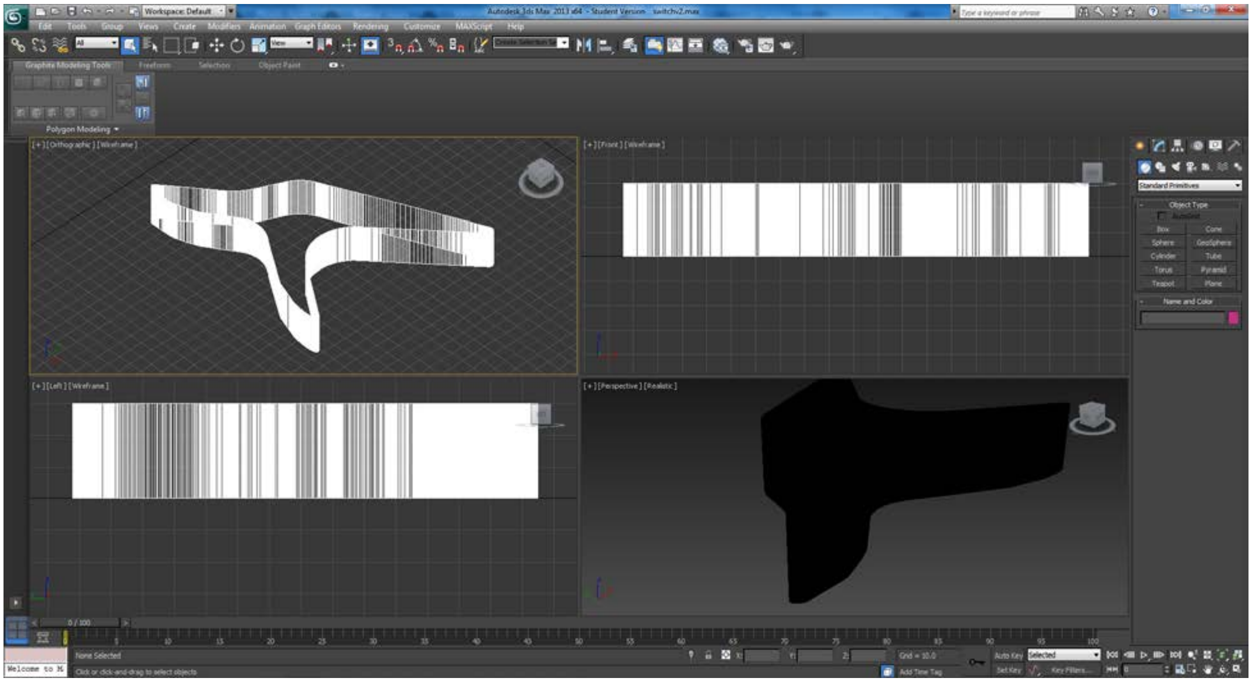


ILLUSTRATOR FIGURE

The black outline shown in the figure above was made by using Adobe Illustrator's pen tool. With it, I placed vertex points at the apex of each contour and adjusted the anchors until the curves between the vertex points matched the curves of the object underneath.

AUTODESK 3DS MAX

After importing the .dwg file into 3DS Max, the parts could be converted into 3-dimensional models. By converting the closed 2-dimensional shape into a polygonal mesh, I then extruded the face it created into a 3D object with an open bottom. After this process was completed, the vertices surrounding the open bottom of the object were able to be selected and capped off to form a closed 3-dimensional mesh that faithfully represented whichever part of the game board I had traced from the scan. I exported this final three dimensional model as a .fbx file that could be imported into the Unity game engine as a valid game object.

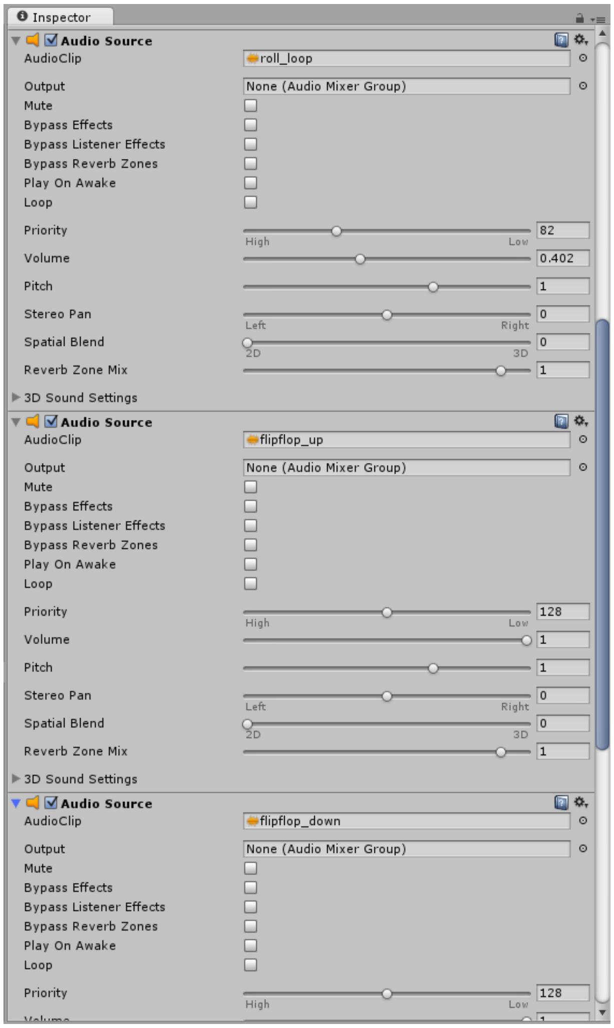


3DS MAX FIGURE

The black object in the bottom-right of the figure above represents the final object created using this modeling method. The three other viewports offer wireframe views of the same object, demonstrating the top and side views of the object. This figure shows the final 3D model of one of the flip-flops.

SOUND DESIGN

To create an accurate representation of Dr. Nim, I required convincing sound effects as well. From the start of the project, Professor Moriarty and I determined that recording the actual sounds of the physical Dr. Nim in action would provide the most authentic gameplay experience. After recording many separate sound files of the game in motion, highlighting the movement of each individual dynamic piece on the board, these files were imported into the Unity game engine as .ogg files. I could then trigger these files in Virtual Dr. Nim where necessary, recreating the sound of the original device.



Inspector

▼ ☑ Audio Source
AudioClip: roll_loop
Output: None (Audio Mixer Group)
Mute ☐
Bypass Effects ☐
Bypass Listener Effects ☐
Bypass Reverb Zones ☐
Play On Awake ☐
Loop ☐
Priority: 82
Volume: 0.402
Pitch: 1
Stereo Pan: 0
Spatial Blend: 0
Reverb Zone Mix: 1
▶ 3D Sound Settings

▼ ☑ Audio Source
AudioClip: flipflop_up
Output: None (Audio Mixer Group)
Mute ☐
Bypass Effects ☐
Bypass Listener Effects ☐
Bypass Reverb Zones ☐
Play On Awake ☐
Loop ☐
Priority: 128
Volume: 1
Pitch: 1
Stereo Pan: 0
Spatial Blend: 0
Reverb Zone Mix: 1
▶ 3D Sound Settings

▼ ☑ Audio Source
AudioClip: flipflop_down
Output: None (Audio Mixer Group)
Mute ☐
Bypass Effects ☐
Bypass Listener Effects ☐
Bypass Reverb Zones ☐
Play On Awake ☐
Loop ☐
Priority: 128
Volume: 1

SOUND DESIGN FIGURE

The image above shows the Audio Sources on the Marble object in the Unity game engine. Multiple sources can be seen in the figure; during the course of gameplay, these sounds will be triggered by their respective events. The sound “roll_loop” is the looping sound of a marble rolling down the game board. When triggered, the pitch of this particular sound is altered relative to the speed the marble is traveling at the moment it is played. The sound “flipflop_up” will trigger when the marble causes a flip-flop to rotate counter-clockwise, and “flipflop_down” plays when a flip-flop rotates clockwise.

DEVELOPMENT BY WEEK

With the entire project being developed over the course of five weeks, I decided very early on to adhere to certain project milestones. By setting requirements for my progress at each weekly iteration, I was able to focus on certain elements of the design process above others, with the 3D art as my first priority, followed by the key gameplay features, sound design, and any other additions I wanted to develop to enhance the gameplay experience goals of Virtual Dr. Nim.

PROJECT SCOPE AND SCHEDULE

Week one of the development of Virtual Dr. Nim consisted of laying the groundwork for the rest of the project by finishing all of the art assets and finalizing the schedule for each week to follow. During this week of development, I also made it my priority to create an asset list to keep track of my progress and note any assets, art or otherwise, that I would require by the end of the project.

PROJECT SETUP

I knew in week one that the weekly structure to the rest of my project work going forward had to be decided on. At this stage, I developed the aforementioned asset list and project schedule. It was at this point that I decided on the game engine I would be using to develop the game, as well.

MODELLING 3D ASSETS

The entire core of my 3D modelling took place in the first week of my project. Utilizing the three-step modelling process noted above, I modeled the red base of the Dr. Nim game board first, in order to determine the in-game scale of the rest of the models. I became aware that the

Unity game engine physics scale with object size, so I had to find the perfect scale for the game board through experimentation, by letting a marble roll from the top of the game board to the bottom, making note of the collisions between the marble and the walls of the game board to assist with determining the scale.

After the most accurate scale of the game board was found, the next piece I modeled and imported was the turn switch, which is used on the game board to signify which player must drop a marble next. This piece was a fairly non-complex shape, and was incorporated into the game engine with ease. I attached this game object to an empty, invisible object on the game board with a hinge joint, which allows the turn switch to rotate around the empty object, just as it would on a pin in the physical game board. This hinge performed extremely well after some adjustment to the strength of the join connection. This led me to decide to use hinge joints for each of the remaining dynamic pieces of the game board.

The next dynamic game board piece I modeled and imported was the equalizer. The standard position this piece starts at is the upward position, where the piece resembles the letter 'U', but once it is flipped, it can never be righted during the course of the game in progress, therefore the movement of this dynamic piece was very easily replicated through the physics engine alone. However, upon importing this asset, I encountered an unforeseen issue when generating collider meshes from my 3D models that I, as someone unfamiliar with the standard art pipeline, did not consider. Some of my 3D models were concave shapes, which the Unity game engine cannot generate colliders for, and therefore generates convex colliders instead. These convex colliders completely nullify the purpose of the switches, preventing them from allowing the marbles to roll along their length and manipulate the switches. For example, the equalizer switch, intended to behave like a physical 'U' shaped object, would instead have a collider that does not have the ‘u’-shaped indentation inside, and the collider would consider an invisible line between the prongs of the ‘U’. This issue would not be solved until the third week of the project.

FINAL ADJUSTMENTS TO 3D MODELS

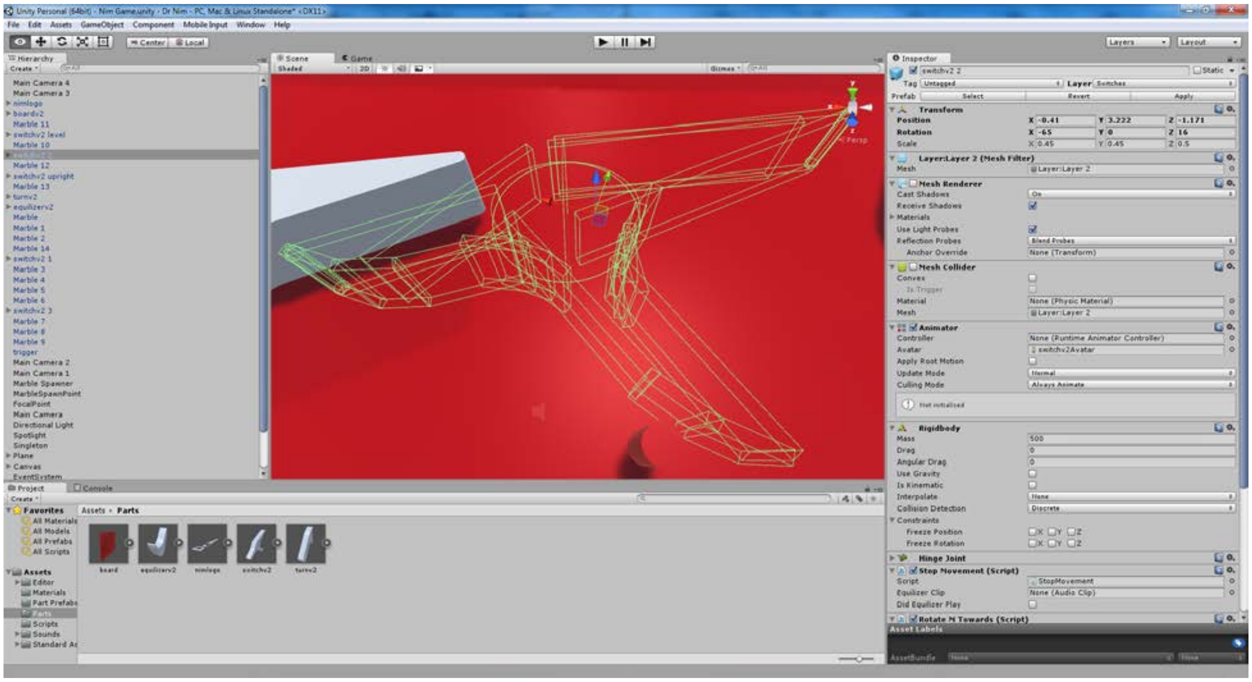
After importing some of the art assets I created, I eventually had to revisit some of the models in 3DS Max in the second and third week of development in order to fix a couple of nuances in the meshes. The main game board required the addition of three small pegs beneath the three ‘flip-flop’ binary switches that they would need to use to support their positions when completing their rotations. I also deepened the channels of the paths in the game boards to prevent marbles from falling out of the grooves. After importing the newly minted game board back into Unity, I also decided to trace and model the Dr. Nim text logo from the original game and import that into Virtual Dr. Nim.

KEY GAMEPLAY ELEMENTS

By the end of the third week of development, I was approaching the milestone I had set for incorporating the key major gameplay elements of Dr. Nim. At this point, I had imported all of the three dimensional assets into the Unity game engine and had solved nearly all minor issues I had come across, except for the outstanding concave collision issue. The gameplay features in effect at this point in development were: spawning the next marble from the queue and activating the switches correctly in the marble’s path, except for the ‘flip-flop’ binary switches, which had been the most affected by the concave collision problem. I discovered a solution in an online forum, Blender Artists Community. With a potential path to fix the problem, I knew this was an issue I needed to have solved by the end of the week.

CONCAVE COLLISION PROBLEM

With Unity still not generating an accurate collider for my concave models, I took it upon myself to simplify the mesh for the collider within the game engine on my own. By creating a multitude of primitive objects with colliders and making them children of the switch meshes I created, I was able to simulate the topography of the concave object using these smaller convex objects. After these changes, the switch-marble collisions behaved exactly as intended.



☑ switchv2 2 □ Static
Tag Untagged Layer Switches
Prefab Select Revert Apply

Transform
Position X -0.41 Y 3.222 Z -1.171
Rotation X -65 Y 0 Z 16
Scale X 0.45 Y 0.45 Z 0.5

Layer:Layer 2 (Mesh Filter)
Mesh Layer:Layer 2

□ Mesh Renderer
Cast Shadows On
Receive Shadows ☑
Materials
Use Light Probes ☑
Reflection Probes Blend Probes
Anchor Override None (Transform)

□ Mesh Collider
Convex □
Is Trigger □
Material None (Physic Material)
Mesh Layer:Layer 2

☑ Animator
Controller None (Runtime Animator Controller)
Avatar switchv2Avatar
Apply Root Motion □
Update Mode Normal
Culling Mode Always Animate
Not initialized

Rigidbody
Mass 500
Drag 0
Angular Drag 0
Use Gravity □
Is Kinematic □
Interpolate None
Collision Detection Discrete
Constraints
Freeze Position □X □Y □Z
Freeze Rotation □X □Y □Z

Hinge Joint

☑ Stop Movement (Script)
Script StopMovement
Equilizer Clip None (Audio Clip)
Did Equilizer Play □

☑ Rotate M Towards (Script)

CONCAVE COLLISION FIGURE

This figure above shows a multitude of primitive shapes within the form of one of the flip-flop switches. Each of the green wireframes represents a primitive shape with its ‘Mesh Renderer’ disabled, so it remains invisible during gameplay, while still providing its collision mesh to the parent object.

UNIQUE SWITCH BEHAVIOR

Even after fixing the collider physics of the binary switches, there were still issues with the switches not turning completely to the angle they needed to go in order to let the marbles drop through. After experimenting with the scale of the objects and the gravity metric of the game scene, I determined that the best course of action was to hard-code the switches. When colliding with a marble, the switches will force themselves to turn completely around to the angle required, moving the marble in the process. This hard coded, collision triggered method served to better represent the actual Dr. Nim gameplay than just simple physics.

FEATURE COMPLETE GAMEPLAY

By the end of the fourth week, all of the intended gameplay features were implemented. A full game of Dr. Nim could be played to completion, with every switch behaving exactly as it should. It was at this stage that I also decided to increase the game speed to four times the normal speed, to better provide an authentic gameplay experience.

ADDING SOUND EFFECTS

With the gameplay features complete, the next step was to expand upon the user gameplay experience. This step included the addition of sound effects. Professor Moriarty's recordings of the physical Dr. Nim in action were imported into Unity. To trigger these effects precisely, I did so by placing each effect on a trigger, with the exception of the rolling marble sound. When a switch was flipped, the corresponding sound effect was played. I also dotted the game board with invisible trigger zones that would set off a marble hit sound effect when collided with, giving the marble an authentic sense of weight, when colliding with certain walls and producing the correct sound effects.

GAME PROGRESSION

The progression of any game of Dr. Nim relies on the ability for Dr. Nim to take its turn effectively. To do so, I had to implement a trigger that would release another marble when the current marble takes a certain path that would set off the trigger. This is the only way the Dr. Nim "AI" could call for additional marbles on its turn. Once implemented, a game against Dr. Nim could be played to completion. At this step in the development process, I also added a few additional features that allow the user to interact with the game board to setup the switches and active the marble release trigger with a mouse click, furthering the user experience goals I had set out to accomplish.

USER EXPERIENCE GOALS

The key pillar of my development of Virtual Dr. Nim was keeping the recreation as faithful as possible to the actual Dr. Nim product that was sold in 1966. The final week of project development was directed toward fulfilling that requirement by making numerous, slight gameplay adjustments and user experience modifications.

I knew from the start of the project that allowing the user to interact with the game via the computer mouse was to be a direct substitute for manipulating the actual Dr. Nim toy by hand. Every switch and toggle that could be pressed or flipped in the actual specimen would have to be able to be interacted with in a faithfully similar way in my virtual recreation. Therefore, towards the end of the development cycle, I began coding every switch and toggle in just this way. Every white game board piece that affords interactivity is able to be manipulated. The user can click to flip each game piece with a click of the mouse, within the confines of the rules of the game, just as they would use their finger on the Dr. Nim toy. The game also restricts the manipulation of pieces that would cause the gameplay to be interrupted or allow the player to cheat.

GRAPHICAL USER INTERFACE

Another user experience goal of the project was to make the user interface informative, but as minimal as possible. Virtual Dr. Nim is meant to mimic the gameplay of the toy, and as such, I saw fit to make any text or gameplay tooltips helpful, while not obstructing gameplay or the faithful representation of the toy. Any pop-up text that references gameplay rules will disappear as soon as the conditions it is calling for is satisfied. For example, when a player tries to release more than three marbles on their turn, a tooltip with explain that three marbles have already been taken and the turn must be handed over to Dr. Nim. Once the player toggles the turn switch to give Dr. Nim its turn, the tooltip will disappear. Any text labels on the board are in the exact same locations as any text that resides on the toy Dr. Nim board, including labels for: the current player turn, the position of the equalizer toggle, and the location of the trigger that releases one marble at a time.

GAMEPLAY POLISH

Towards the end of the project development, I made an effort to work out any graphical, audio or gameplay glitches that had occurred during the course of development and personal testing. For example, one audio glitch I encountered caused the triggering of the same sound from both objects in a single collision, causing two instances of a sound to play almost in unison; this glitch was nearly eradicated in the final build of the game. Also, while polishing the gameplay, I added features for multiple camera angles and the ability to rotate the main camera around the game board slightly, with assistance from a tutorial on the Unity3DStudent website.

These alternate camera features provide a more tangible representation of the 3D game board, and allow the player to explore how faithful this virtual recreation is to the original.

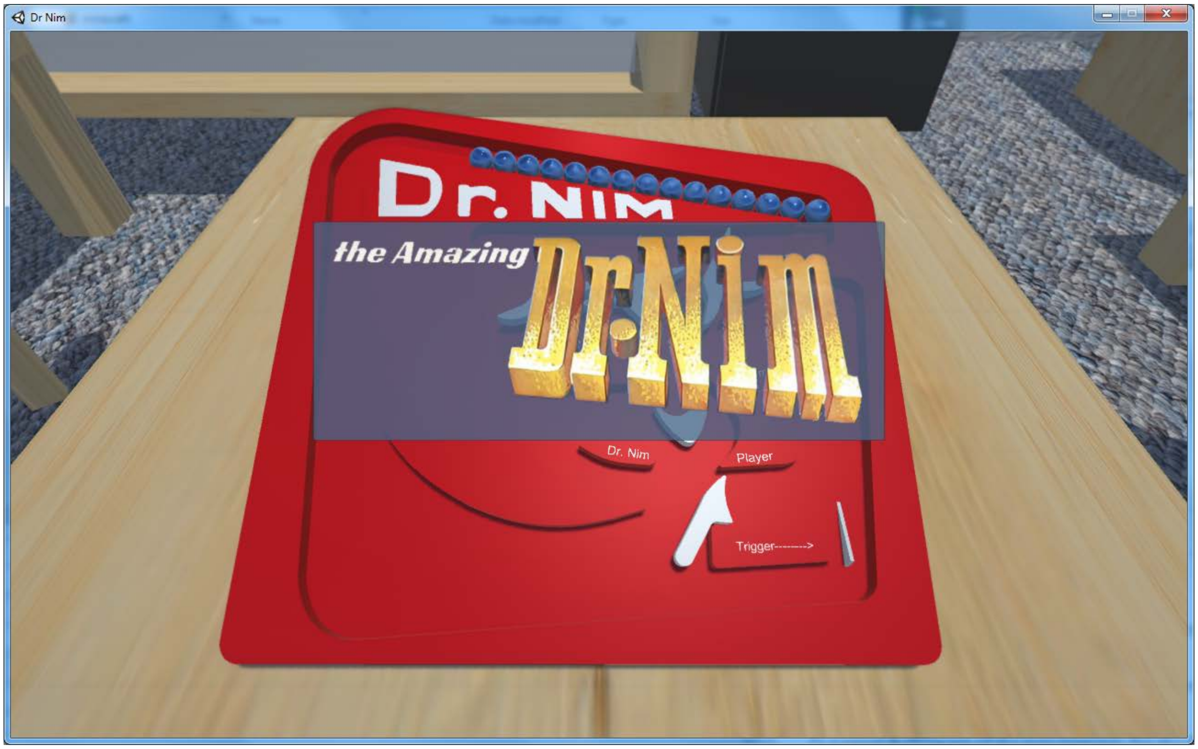
POST MORTEM

Working on this project alone gave me a sense of pride upon its completion, having had to accomplish each of the various aspects of the development process as an individual. The scope of this project turned out to be the perfect size for an MQP of just one term, with just one person developing it. The final product of my work this term has hit many of the marks that I was hoping it would. And while there are a few areas of the development I would have done differently, I believe that overall, this project was very successful and I achieved what I had set out to achieve.

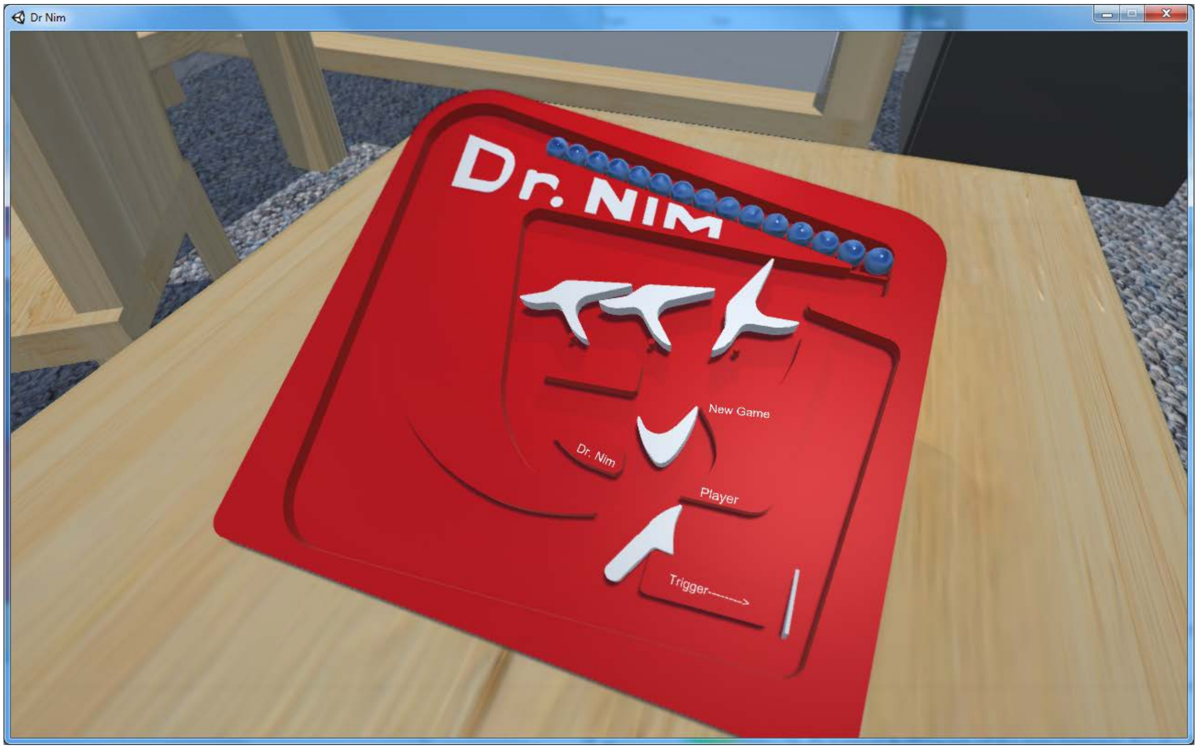
PROJECT SUCCESSES

One aspect I believe I really succeeded in is the gameplay mechanics. Virtual Dr. Nim plays a full game of Dr. Nim against a human player to completion, in 100 percent the same way that the toy Dr. Nim game would. Every switch and moving piece on the board behaves just as its real-world counterpart does. I have also taken steps to prevent some ways of cheating in Virtual Dr. Nim, without changing the spirit of the game as it was intended.

I also accomplished the sound design and user experience goals I set out to achieve. While the game behaves as the toy counterpart would, it also sounds and looks like the toy. By capturing the sound of an actual Dr. Nim specimen in motion, Prof. Moriarty and I ensured that the audio in Virtual Dr. Nim would be 100 percent faithful to the toy product. And thanks to the high-resolution scans we were able to make at the project’s conception, every shape and contour on the Virtual Dr. Nim game board and pieces is accurate as well.



The figure above shows the start-up screen of the final version of Virtual Dr. Nim, while the figure below shows the game environment during play.



CHANGES I WOULD MAKE

If I could go back and develop certain aspects more finely, or further develop this project, there are certainly some things I would like to address. For one, the gameplay speed of Virtual Dr. Nim is a bit slow compared to its real-world counterpart. After reviewing the Unity Engine physics options, and viewing video tutorials by developers like ETeeskiTutorials, it seems likely that the slower speed could be due to the scale of the game board and other game objects, as Unity's physics scale with object mass and volume. On a smaller scale, the physics might perform more faithfully to the Dr. Nim toy.

With more development time, I would also make it a goal to focus on the environment of Virtual Dr. Nim. I envisioned beyond my initial "minimum viable product," placing the virtual board on a table in a living room setting, with the camera focus trained on the game board, while the environment in the background remained out of focus. I figured additions like this would give the recreation a more tangible setting, and therefore, more faithful appearance to the original. I managed to develop a rough setting for the final product, but I would have liked to have spent more time making the background immersive.

FUTURE WORK

This project affords a few different options for further work. In the original Dr. Nim instruction manual, which is available online for reference, there are diagrams of many additional starting gameplay scenarios. These scenarios include: varying numbers of starting marbles, differing positions of the switches based on the number of marbles in play, and a variation on the gameplay goal by asserting that the player to take the last marble wins rather than loses.

I would also insist on taking this opportunity to develop a mobile version of Virtual Dr. Nim. Not only as a logical extension of the work to reach a larger amount of people, but as a personal means to get my start in mobile application development. With the Unity game engine as a very flexible tool for publishing games to different platforms, making the jump to mobile devices could serve as a simple and logical next step.

WORKS CITED

"Beginner B19 – Following with LookAt()." Unity3DStudent. Will Goldstone, n.d. Web. 17 Apr. 2015. <http://www.unity3dstudent.com/2010/08/b19-following-with-lookat/>.

ETeeskiTutorials. "Unity 3D - Physics, Gravity, Mass, and More [UnityQuickTips]." YouTube. YouTube, 7 June 2013. Web. 26 Mar. 2015. <https://www.youtube.com/watch?v=wKgKSg0Nr6A>.

"Physics." Unity3D. Unity Technologies, n.d. Web. 4 Apr. 2015. <https://unity3d.com/learn/tutorials/modules/beginner/physics>.

"Thread: Collision Bounds for Concave Objects?" Blender Artists Community RSS. CG Cookie, Inc., n.d. Web. 6 Apr. 2015. <http://blenderartists.org/forum/showthread.php?215114-Collision-bounds-for-concave-objects>.

Unity Documentation. Unity Technologies, 2015. Web. 20 Mar. 2015. <http://docs.unity3d.com/Manual/index.html>.

IMAGE URLS

1. Figure 1: https://cf.geekdo-images.com/images/pic126421_lg.jpg
2. Figure 2: https://cf.geekdo-images.com/images/pic126424_lg.jpg
3. Figure 3: https://cf.geekdo-images.com/images/pic126419.jpg

APPENDIX A: ASSET LIST

Asset Name	Filename / Components	Status	Placeholder
3D Models			
Binary Switch x3	switch.fbx	Complete	
Equalizer	equalizer.fbx	Complete	
Turn	turn.fbx	Complete	
Gameboard	gameboard.fbx	Complete	
Marble	marble.fbx	Complete	
Sound Files			
Bounce 1	bounce1.ogg	Complete	
Bounce 2	bounce2.ogg	Complete	
Equalizer Toggle	equalizer_toggle.ogg	Complete	
Binary Switch CW	flipflop_down.ogg	Complete	
Binary Switch CCW	flipflop_up.ogg	Complete	
Marble Bounce	marble_bounce.ogg	Complete	
Marble Hit	marble_hit.ogg	Complete	
Marble Stop	marble_stop.ogg	Complete	
Marble Stop (multiple)	marble_stop_multi.ogg	Complete	
Trigger Release	release.ogg	Complete	
Marble Roll (loop)	roll_loop.ogg	Complete	
Turn Toggle	select_toggle.ogg	Complete	

APPENDIX B: PROJECT 5-WEEK SCHEDULE

Project Schedule by Meeting:
March 23:

- Asset list and schedule complete
- Work done on 3D models
  - If progress is not great, get help
- Game engine decided on, github repository created

March 30:

- All models in-engine
- Gameplay and physics working
- Key press to spawn one marble at a time
- Default white background

April 6:

- Gameplay complete
- Switch physics final adjustments
- Dr. Nim turns complete

April 13:

- All audio complete and implemented
- User Experience and menus functionally completed
- Background environment updated

April 20 (Patriots Day)(No classes):

- Complete User Experience goals
- Any final polish to gameplay