Rocket Builder: Supplementary Learning for Forces & Motion Curricula

by

Connor Adam Shipway

A final project submitted

to the Graduate Faculty of

North Carolina State University

in partial fulfillment of the requirements for the

Master of Art + Design

Interactive & Experimental Media Concentration

Raleigh, North Carolina

2019

APPROVED BY

Todd Berreth, M. Arch.

Committee Chair

Emil Polyak, M.A.

Co-Director of Graduate Programs

Eric Wiebe, Ph. D.

Acknowledgements

A huge thank-you to Todd Berreth for his boundless patience, to Emil Polyak for his constant helpful advice, and to Eric Wiebe for his kind compliance in assisting a student he barely knew. Thanks as well to my fellow graduate students who encouraged me, especially my co-worker Stephanie Huang, whose company I have come to greatly appreciate. Lastly, thanks to my family, whose support and belief in my work has never wavered.

ABSTRACT

Teaching physics in middle school classrooms can be a difficult task to negotiate due to the abstract nature of concepts such as Forces and Motion. Traditional classroom props are, somewhat counterintuitively, limited in conveying these ideas in that their adherence to all the laws of physics is unceasing. It can be difficult to demonstrate, for example, inertia (that property of motion which describes the tendency of motion to preserve itself) when friction tends to slow objects to a halt here on the surface of the earth. *Rocket Builder* is a game-based interactive digital experience intended to act as a supplementary learning aid in the teaching of these abstract concepts. The concept for this project was initially conceived as an interactive museum exhibit, but as the idea developed, a greater emphasis was placed on a fine degree of player input which suggested more traditional computing setups as a suitable platform.

The decision to utilize video games is not a decision made merely out of acknowledgement for the potential the medium has to engage students, but also for the format's inherent experimental nature. A game's systems determine such things as the manner in which game objects interact with one another, the goals the player has, and the obstacles which might prevent the player from obtaining that goal. In the case of this project, and those of its many of its influences (such as the sprawling *Kerbal Space Program*), these systems are analogous to the principles which govern real-world physics. This unique vocabulary is the foundation for an expressive mode of communication which enables learning by allowing the player to engage in an experiential dialogue with the game's systems.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

INTRODUCTION TO THE WORK

The basic concept for Rocket Builder was originally conceived as a solution to a project assigned in Professor Todd Berreth's course, ADN 561 -- Narrative and Artmaking Through Digital Interfaces. The goal of the project was to design and propose a museum exhibit which would occupy one of a few predetermined spaces in the Durham Museum of Life and Science. Eventually, each proposal would actually be reviewed by employees of the museum to be considered as a real, potential installation. As a means of preparation for this project, a class field trip was conducted to familiarize the students with the potential installation sites, as well as to give the students an idea of what sort of exhibits occupied the museum and, on that basis, which sort of proposals might fit in well with the existing array of content.

During the visit, I noticed a great deal of content which dealt with space travel and aeronautics in general. One example of such content was a station at which users could construct a paper airplane from an ample supply of provided paper, then deploy the paper airplane by means of a "launcher" contraption. I was intrigued by the fun factor and hands-on nature of the installation, but was dismayed to see that the launcher was not in good working condition -- a gradual development which had rendered the station more or less unusable. It occurred to me that an installation which enabled users to construct and deploy an aerospace artifact e.g. a paper airplane, but forgoed an error-prone mechanical solution in favor of some sort of digital game, might accrue some degree of success.

This was a happy coincidence, as the subject of some of my technical studies in the Unity 3D game engine dealt largely with the creation, movement, and manipulation of game objects which behaved realistically according to the laws of physics. I enthusiastically chose this concept

as the basis for my project proposal. In a confidence-inspiring turn of events, my peer and coworker Stephanie Huang coincidentally chose a nearly identical concept, and we received permission to combine our efforts and produce a doubly refined proposal. Over the course of the duration of the project, we developed a concept for an installation which we would eventually come to entitle *Rocket Builder*. In this imaginary installation, users would gather around a tabletop covered with a variety modular construction-paper rocket components. These components could be assembled per the preference of the users, scanned using a camera-enabled kiosk, and deployed into a digital physics environment which would be projected onto the floor of an adjacent room. Scrabbling about the floor and clutching custom-built game controllers, users would explore an imaginary solar system using their very own rocket. In addition to being a socially engaging experience where users express their creativity and have fun, this physics sandbox would even feature physics analogous to those of real space travel, for example, taking gravity and orbital trajectories into account.

The proposal was met with polite interest by the museum and, though it never progressed beyond consideration in the affair of becoming a real exhibit, it served myself and my partner well as far as the class project was concerned. The proposal and its accompanying concept art and poster settled into the annals of my portfolio and the back of my mind.

As it turned out, the concept's retirement was short-lived. The following semester arrived and with it, the need to explore research topics and potential projects which might have the depth to sustain my larger graduate studies. I was still fascinated with the possibilities posed by the idea Stephanie and I had designed, and interested in trying to develop it into a real product. Furthermore, the concept was no longer limited by the constraints which are inherent to the

medium of a museum installation, so I was free to imagine other avenues for distribution. It was the result of this combination of events that Rocket Builder became one of my first avenues for serious experimentation that semester, and consequently the topic of my graduate project.

Statement of Work

Rocket Builder is a game-based learning experience intended to act as a supplementary teaching aid for the Forces and Motion section in middle school science curricula. Using a drag and drop grid interface, players can construct a rocket out of simple components, and then deploy their built craft into a dynamic physics environment which will put their designs to the test. Rocket Builder seeks to engage and entertain students, while illustrating motion concepts which might otherwise be frustratingly abstract.

Summary of Papers

The design of Rocket Builder was influenced by ideas posited by various prevalent researchers. These include not only educational theories which suggest models and techniques for effective learning that Rocket Builder seeks to harness effectively, but also ideas which address directly the medium of games and its unique ability to convey concepts expressively, such as in a learning context. Rocket Builder is influenced by many historic and contemporary influences, ranging from the arcade classic *Asteroids* to the relatively modern physics simulation tool *Algodoo*. Though initially conceived as a museum installation, Rocket Builder developed to its current state as a more traditional computer application early in its design process in response to better-clarified learning objectives and an interest in maintaining accessibility to a broad audience. Over the course of this larger process, various design and development decisions were

made in an effort to realize Rocket Builder as a learning tool which would find an effective place in a traditional teaching setting such as a classroom.

THEORETICAL ASSUMPTIONS

One of the primary objectives of Rocket Builder is to explore the viability of video games as a suitable medium for expressing sophisticated learning concepts. Beyond my own anecdotal experience, I would argue that theories posited by various prominent researchers lend credence to the notion of video games' effectiveness as a teaching tool.

Bogost's Procedural Rhetoric

The first idea that I'd like to discuss is one that I think is especially pertinent to the topic of games as educational tools, and one that I personally subscribe to pretty heavily. It's a theory introduced by Ian Bogost, a professor of Interactive Computing at the Georgia Institute of Technology, which he has coined as "procedural rhetoric". Bogost's definition of rhetoric in the context of this idea is "the notion of elegance, clarity, and creativity in communication" (124), a description which might exceed the scope of what might traditionally be considered rhetoric -- perhaps, something to the effect of "argumentative or persuasive speechcraft". This description is certainly not incompatible with Bogost's definition, it would just be classified more specifically as being oral rhetoric -- expressive or effective communication which utilizes the spoken or written word. To develop this idea, I imagine that many members of the College of Design should be familiar with visual rhetoric, which would be expressive or effective communication which utilizes photographic, cinematographic, or illustrative techniques. Bogost posits a third form of rhetoric which he calls procedural rhetoric, which would be expressive or effective communication which utilizes, as its components, those systems of rules and mechanics which, by definition, constitute a game (125).

On a very surface level, a game might appear to be characterized primarily by its aesthetic presentation. But strictly speaking, a game is more accurately assessed as being defined by the rules which govern its gameplay -- the goals of the player, the obstacles which prevent the obtainment of that goal, and the systems through which the goal is obtained. This rule-based paradigm is more easily represented by the elegantly abstracted presentation of say, a chessboard and its pieces. The goal of a chess set is not to actually visually depict a medieval battle, but simply to assist in the representation of a set of rules. In the case of chess, these rules would include such aspects as the manner in which a given type of piece is permitted to move, as well as what a player has to do in order to defeat their opponent.

A game designer knows that the in the creation and manipulation of these sort of rule sets, there exists an inherent potential for expression and communication. Bogost suggests:

> Game developers can learn to create games that make deliberate expressions about the world. Players can learn to read and critique these models, deliberating the implications of such claims. [...] When games are used in this fashion, they can become part of a whole range of subjects. (120)

The expressions to which Bogost refers do not necessarily have to be lofty statements about the human condition, or arguments toward a political stance. They can be about something as simple as how objects move about in space.

Kolb's Experiential Learning

This theory is one that is less related to games, and more so broadly applicable to effective education in general. David A. Kolb is a well-known educational theorist best known for his research concerning what he calls "experiential learning". Kolb argues in his 1984 book

Experiential Learning that "learning is the process whereby knowledge is created through the transformation of experience" (38). He develops this argument, exploring models of learning proposed by educational theorists of the past, most notably a model conceived by psychologist Kurt Lewin which illustrates a 4-stage, cyclical process (fig. 1). The argument, which Kolb incorporates liberally into his own, is that this cycle depicts the essentially experimentative act of learning:

> Immediate concrete experience is the basis for observation and reflection. These observations are assimilated into a "theory" from which new implications for action can be deduced. These implications or hypotheses then serve as guides in acting to create new experiences. (21)

This model of learning is arguably very compatible with the "experimentation"-focused approach which Rocket Builder attempts to utilize. A player, in order, conceptualizes a design for a rocket, conducts an experiment so as to determine whether the rocket behaves in an intended manner. They judge the performance of their built craft, considering whether it aligns with their original hypothesis, and hopefully reflects upon the experience, deliberating what could be changed and how, before repeating the process again.

Fig. 1. Kolb's cycle of experiential learning. Image credit: Simply Psychology.

Mayer's Multimedia Learning

This last idea is one attributed to prominent educational psychologist Richard Mayer, who explored the theory in various essays before its culmination in his 2001 book *Multimedia Learning*. From the first line of the introduction, Mayer suggests, "People learn better from words and pictures than from words alone" (1). This is the gist of multimedia learning, which is specified to be defined by its usage of both verbal and pictorial components (5). Naturally, this is attainable through a variety of forms of delivery -- multimedia learning can be as sophisticated as a verbal presentation given to hundreds of audience members alongside a lavishly animated slideshow, or it can be as simple as written prose alongside illustrative diagrams, as in a basic textbook.

The supposed effectiveness of multimedia learning rests in the idea that the human ability to understand a body of information is limited or "bottlenecked" by each sense which is instrumental in perceiving that information -- this is to say, one can only see so much at any given time, or hear so much at any given time. The idea is that, by increasing the number of senses which are used to interface with a body of information, one may increase the quality and quantity of information which may be meaningfully understood by a user.

Rocket Builder's intended presentation as being a demonstrative aid, presented alongside more traditional verbal instruction, falls squarely into that category which Mayer would classify as multimedia learning. Teachers might discuss the concept of inertia while simultaneously pointing out a ship's persistent state of motion in a Rocket Builder game session, or verbally differentiate balanced and unbalanced forces while demonstrating components which would, in light of their positioning and placement, create such forces.

HISTORIC AND CONTEMPORARY INFLUENCES

Rocket Builder can almost certainly be considered a physics game. The primary mechanic of Rocket Builder is the navigation of an imaginary rocket craft, an end which is accomplished by the manipulation of physics forces upon the craft. As such, it should not be surprising that the majority of the precedent which constitutes Rocket Builder's influences are also physics games.

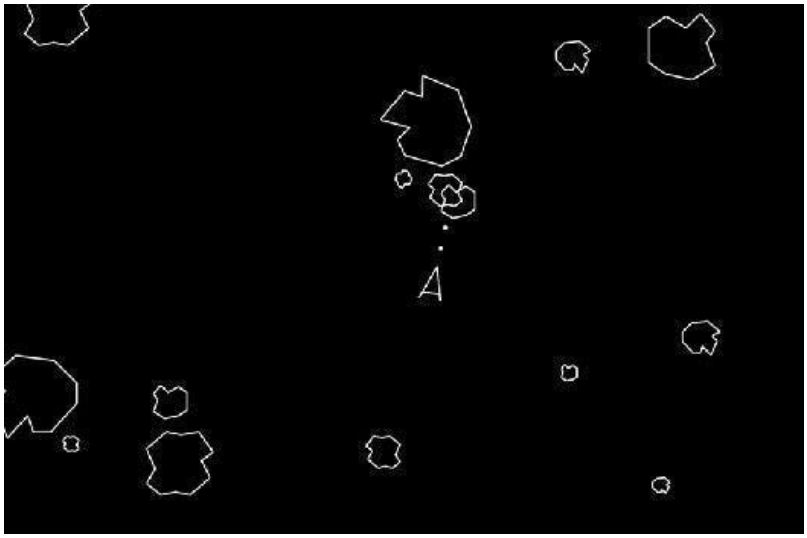

Fig. 2. Screenshot of the video game *Asteroids.* Image credit: ShortList.

A physics game might be defined as any video game which implements systems that simulate and analogize some aspect of physics -- e.g. forces and motion, fluid dynamics, gravity -- so as to enable some primary mechanic of the game. It must be conceded, however, that this definition is really more of an attempt to assess a genre than a hard classification -- to some extent or another, even the earliest video games have implemented systems which simulate physics, be it the loose Newtonian metaphor which informs the puck in the Atari classic Pong, the inertia-governed movement of Asteroids (fig. 2), or the pull of gravity which keeps the eponymous plumber protagonist grounded in the quintessential Nintendo platformer title, Super

Mario Bros. However, I would argue that any games analyst with modern sensibilities would hesitate to classify any of these titles as being a proper physics game -- one expected requisite for classification as a physics game is certain level of complexity or robustness which these earlier titles simply lack.

To this day, one of the most notoriously well-regarded physics games is Valve's 2004 first-person shooter *Half Life 2*. Half Life 2, like most other first-person shooters, prominently features combat-centric gameplay, arming the player character with an arsenal of weapons with which to dispatch an onslaught of hostile alien denizens. However, much of the game's innovation is derived from its expressive utilization of the Source physics engine. Half Life 2's sequences of action-focused combat stand alongside sequences of thoughtful, deliberate puzzle solving which use physics principles as the foundation. The player character wields a fantastical contraption called the Gravity Gun to target, grab, and reposition almost any object in the game world, from plastic bottles to wooden crates. The sheer versatility afforded by this system results in fantastically open-ended challenges that reward creative and thoughtful problem solving. Half Life 2 released to massive critical acclaim, and this success likely paved the way for other physics-based games to be developed in the future.

One such game, and one which influenced my own sensibilities in regards to game design and development, is Media Molecule's 2008 title *LittleBigPlanet*. In LittleBigPlanet, the goal is to reach the end of each level by navigating your character through an obstacle course of platforming challenges, physics contraptions, and environmental hazards. One of the game's most remarkable characteristics is its heavy implementation of physics systems into gameplay -- LittleBigPlanet prominently features a dizzying array of materials, objects, and gadgets --

components which each interact modularly and dynamically with the others. Each material has density -- wooden objects float in water, but metal sinks -- and each object has volume and weight. Pistons, hinges, motors, and levers work in tandem to constitute complex machines with dozens of moving parts. By means of a sophisticated level editor, LittleBigPlanet presents this sprawling hodgepodge of physical components to the player, as a versatile toolset with which to create their own obstacle courses and mechanical creations.

My enjoyment with LittleBigPlanet at the time of its popularity cannot be overstated -- I would spend hours in the level editor, fiddling about and experimenting with the limitations of the game's systems. One building component which always struck me as being particularly amusing was the thruster, a contraption which produces a massive pushing force upon whichever object it is anchored to, emitting plumes of black cartoon smoke in the process. The intuitive use of the thruster was to create all fashion of rocket-propelled devices which would careen into the air, containing in their interior the player-controlled character, frantically clinging on for dear life. The sheer amount of time I spent in the pursuit of this single activity alone is a testament to the potential for player engagement which rockets afford.

However, at a fundamental level, LittleBigPlanet is a platforming game -- not a game about the fine control and manipulation of mechanisms. Designing a way to maneuver these rockets beyond a simple on-off switch is largely beyond the scope of the game's input tools (little more than a binary-state lever and a big red button), and there is no way to disable the gravity present in the game world. Consequently the enjoyable situations enabled by the game's take on rockets inevitably end in some sort of comical disaster -- smacking into the ceiling at the top of the universe, spinning out of control and careening into the ground, or exploding into a

cloud of wayward components. The feeling of satisfaction which comes from the successful design of a contraption which operates consistently and successfully is regrettably absent in LittleBigPlanet -- at least, in the implementation of its wild and unpredictable thrusters. Nonetheless, the game left a lasting impression on me, and my time with LittleBigPlanet was probably my first prolonged and serious exposure to the world of physics games. At the very least, LittleBigPlanet constitutes a respectable fraction of my inspiration for the concept of Rocket Builder, if not an outright precedent.

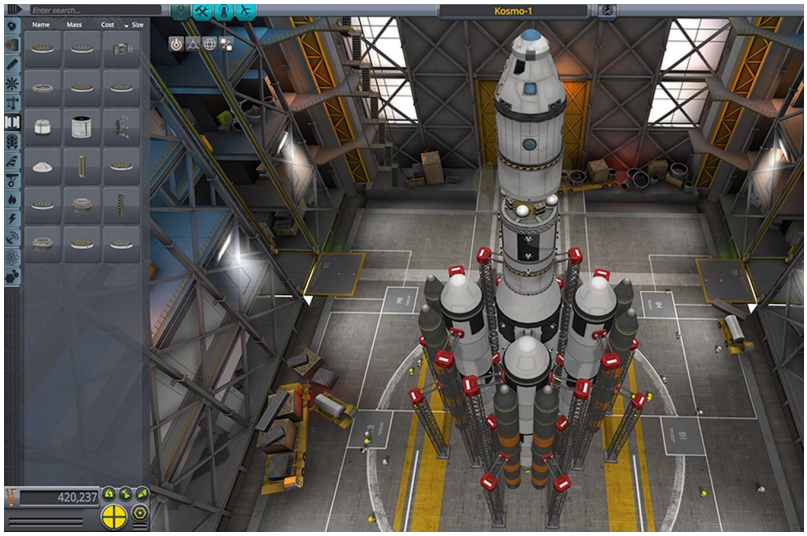

Fig. 3. Screenshot of the video game *Kerbal Space Program*.
Image credit: Kerbal Space Program official website.

Kerbal Space Program

An example of a physics game which certainly constitutes an outright precedent -- dealing primarily with themes such as rocket construction and space travel -- is Squad's 2015 title *Kerbal Space Program*. In Kerbal Space Program, you act as the overseer for an imaginary NASA-like entity, with the goal of manufacturing and controlling sophisticated spacecraft (fig. 3). Space travel is a complex and multifaceted affair, and the game's countless systems reflect this. Players must plan and organize the launch, ascent, orbit, extraplanetary landing, and reentry

of their built craft, all in three dimensional space, while simultaneously managing limited resources such as fuel, in order to successfully accomplish their goals. The result is a vast barrier to entry which may frighten away players who do not fall within the game's narrow target demographic -- essentially the appeal of the game is tied to the robustness (if not necessarily perfect accuracy) of its simulation, and players who enjoy Kerbal Space Program must be willing to commit the dozens of hours which its mastery necessitates.

Though Kerbal Space Program is a highly interesting case study in its own right, its relevance to Rocket Builder is largely as a point of reference with which to contrast. Kerbal Space Program is defined by its complexity, barrier to entry, and enthusiasm with minutia. In the context of a three-dimensional, atmospheric, gravity-influenced space, any given event is the result of countless interwoven systems -- a mechanical component misplaced by a fraction of a unit might result in an inaccuracy of a degree in the launch sequence, which might result in a spectacular explosion. In such a paradigm, a black box separates the player's decisions from the effects of those decisions, and unraveling the intricacies of that black box is an activity which requires a great deal of time, patience, and enthusiasm -- qualities which are not known to be present in abundance in the dispositions of modern middle-schoolers. Conversely, Rocket Builder is defined by its simplicity, accessibility, and emphasis on the "broad strokes" of physics learning. In the context of a two-dimensional, frictionless vacuum, any given event is the direct result of the player's decisions interfacing simply and meaningfully with one or two physics systems. Essentially the pace of experimentation is more rapid, and staggering depth is eschewed for accessibility and breadth of the target demographic.

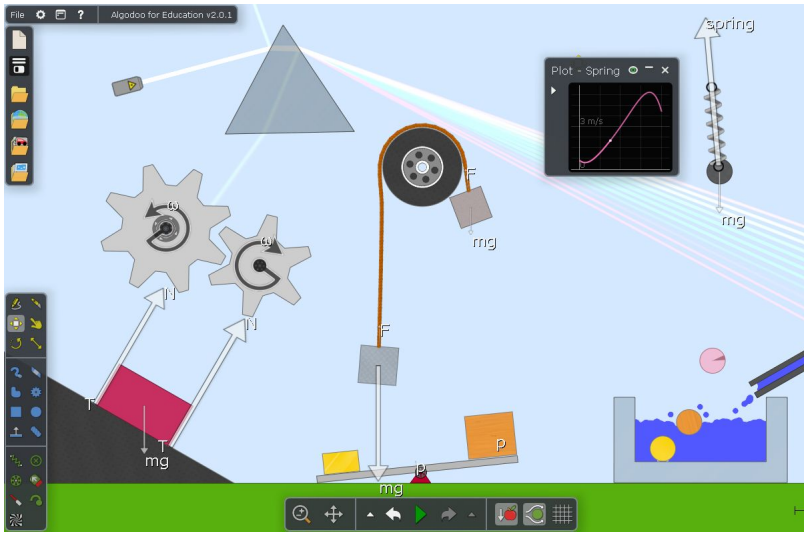

Fig. 4. Screenshot of *Algodoo.* Image credit: Algodoo official website.

Algodoo

It should also be noted that all of the examples above are, first and foremost, commercially-released video games which possess a primary goal of consumer entertainment. Though one could certainly attempt to imagine the potential application of *LittleBigPlanet* in a classroom setting, the true intention of the game is and has always been, first and foremost, to entertain. The overabundance of gamification elements such as of obstacles, points, and various major and minor victory conditions might potentially distract, obfuscating the potential for thoughtful physics learning. Therefore, in the development of what is hoped to serve as a classroom tool, it is useful to consider precedent which is targeted more specifically to an educational application. One such example is that of *Algodoo*.

Algodoo is a free 2D physics-based sandbox application. Conspicuously absent from Algodoo is the presence of any victory condition, performance tracking, and video game tropes and conventions -- the objective, if Algodoo can be said to have an objective at all, is entirely physics experimentation and exploration. Users can create objects with manipulable physics

properties, associate them using a variety of tools, and ultimately create interesting simulations and creations. Numerous studies have been conducted to analyze the effectiveness of Algodoo in a classroom setting at teaching such physics concepts as Archimedes' principle, and rolling motion, with encouraging results (Çelik 40; Nakamura).

Algodoo's success in the educational sphere bodes well for the future of other physics-based teaching tools, but nonetheless it differs from Rocket Builder in several key aspects. Algodoo's scope is far wider than that of Rocket Builder's, encapsulating such concepts as friction, fluid dynamics, and motors (fig. 4) -- Rocket Builder focuses exclusively on forces and motion. Further, while Algodoo's open-ended sandbox approach is effective to its own ends, Rocket Builder seeks to reclaim some of the targeted problem-solving inherent to puzzle games as means of maintaining player engagement.

Another interesting aspect of Algodoo is its focus on enabling the sharing of content. A scene or contraption devised in Algodoo can be easily exported into a lightweight file which can be exchanged between users -- for example, on the Algodoo forums, which to this date has accumulated hundreds of user-generated threads of discussion (Algodoo Forum). Throughout the process of developing Rocket Builder, it was inspiring to imagine the game as something that could potentially gain a community of enthusiasts. I was encouraged to consider how I might design the game as to enable that sort of possibility.

One interesting aspect of this sort of shared content is the potential to set in place the building blocks for the simulation or exhibition of a certain physics phenomena -- say, for example, a row of dominoes falling in sequence. The user who designs this scene must possess a rudimentary understanding of how to use the Algodoo interface -- how to place objects, and the

like. However, the user who loads this scene can run the preconfigured simulation and glean its learning objectives (the transfer of momentum along the sequence of dominoes) with barely any proficiency with the program's design tools. In this manner Algodoo becomes more than just a design tool for physics simulation, but more broadly, an open-ended platform for more general physics teaching and learning, utilizing a sort of dynamic relationship between instructors and students.

Rocket Builder's influences range from commercially-released consumer entertainment to educationally-focused physics experimentation, but each piece of precedent shares one quality in common -- the implementation of physics simulation as an engaging premise which influences the core mechanics of a game. In designing Rocket Builder, picking and choosing useful ideas from each of these influences has been a useful and necessary process so as to more precisely identify the objective of the software, as well as the means by which it might be accomplished.

PROCESS

Rocket Builder, in its earliest form, was designed as a means to fulfill a project for Todd Berreth's ADN 561 course, Narrative and Art-making Through Digital Interface. The project, in which I worked alongside my peer and co-worker Stephanie Huang, entailed the design of an imaginary museum exhibit which would occupy one of several real spaces in the Durham Museum of Life and Science. At this stage in development, the effective utilization of the museum space, as well as the special considerations inherent to the museum environment, were of primary importance as far as the matter of design decisions were considered.

The rocket-centric theme which would go on to become essential to the project was chosen largely due to its compatibility with many of the other exhibits which are featured at the Museum of Life and Science. Considerable space in the museum is dedicated to the demonstration of spacefaring vessels and equipment, as well as the exploration of concepts related to space travel. A concept which would echo these themes seemed likely to garner some success. Additionally, even at this early stage in the process, I had an idea of what the development of a physics-driven game environment (such as the one that would be necessary for Rocket Builder) would entail. The prospect of tackling such development head-on was enticing to me, especially since fostering my familiarity with the Unity game engine was of particular importance to me at the time.

Given that any design would likely have to facilitate the sharing of a single museum space amongst multiple users, an exhibit which framed itself as a socially-engaging space was chosen as a sensible idea. Thus at this stage a "multiplayer" environment was a given, and indeed

framed as being one of the draws of the exhibit -- the ability to explore an environment alongside other eager players was emphasized (fig. 5).

Even at this point, this aforementioned exploration was qualified by the promise that a player's craft would be one of their own creation -- indeed, from the beginning, the project's title was Rocket Builder. To some extent, the users would customize their ship by selecting between and assembling together a pre-set variety of modular rocket components (fig. 6). The idea was to enable in the construction process a physical aspect, in which the user could actually move about construction materials -- perhaps cheap wooden blocks, or cut-out pieces of construction paper -- on a work surface. Once the user was happy with their creation, they would employ some sort of scanning kiosk to deploy a digital version of their physical ship into the virtual physics space, located conveniently in the adjacent room.

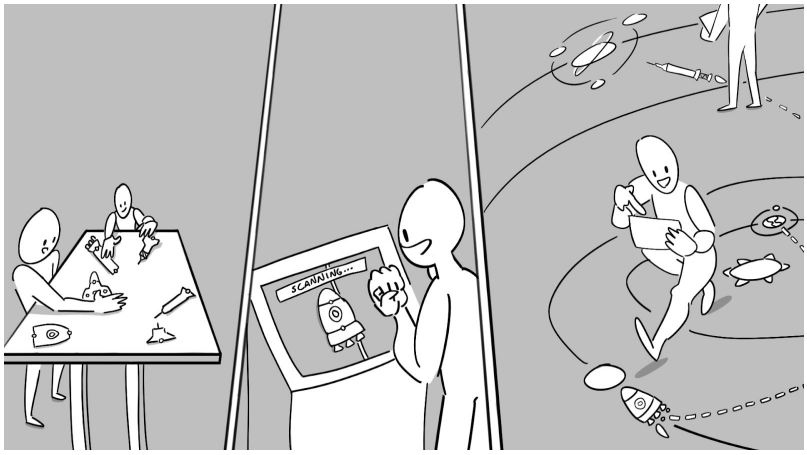

Fig. 5. Early concept work for *Rocket Builder*. Image credit: Shipway, Connor. 2017.

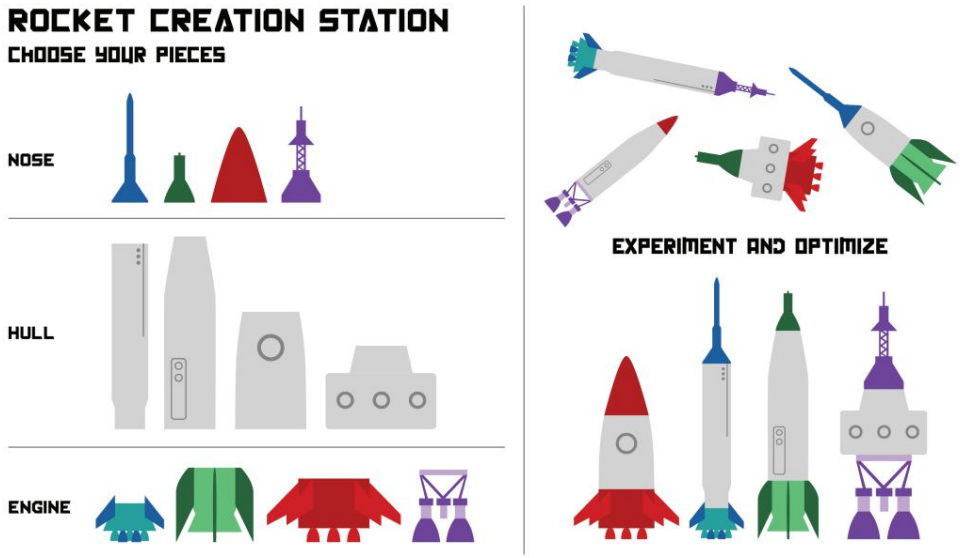

Fig. 6. Early concept work for modular building components. Image credit: Huang, Stephanie. 2017.

The novelty of this mode of interaction was thought of as being one of the main sources of appeal for the exhibit. However, the utilization of this idea imposed several design constraints upon our process. For example, given the constraint of physical fabrication, the extent of user customization would have to be decidedly limited -- construction paper stuck together with velcro bits would only remain structurally sound provided the built rockets were fairly simple, perhaps two or three components in all. This informed the "modular" design we embraced at the time. To what extent, if at all, the user customization would affect the ship's properties of movement was a question which we left largely unexplored -- the aesthetic customization seemed sufficient to engage users, allowing them to explore socially in a craft which they found visually appealing.

Despite the differences between Rocket Builder at this stage and what the game would eventually become, the overall framework of game's core interactions was essentially defined at this point, and would remain similar for the entire process. This is to say, the general premise of

building a ship and deploying that ship into a physics environment was laid out very early in Rocket Builder's conceptual design -- that is, before development proper started at all.

Refining the Concept

As I moved into the next semester and consequently into Marc Russo's ADN 560 Animation Studio, the Rocket Builder concept was at the forefront of my mind as I started to ponder what sort of topics might be appropriate to explore for my graduate studies. The design constraints which had been inherent to the stipulation of the museum environment were no longer pressing, and I was left free to consider which aspects of the concept I actually found the most interesting -- both personally, and insomuch as creating an effective learning experience was concerned.

One of the first features which I felt would be wise to reconsider was the physically-presented building process. While the concept was admittedly novel and would certainly serve to spark engagement in a museum setting, I imagined that this initial interest would eventually give way, as users quickly exhausted the limited opportunities for customization which these techniques would necessitate.

What I was especially interested in was the possibilities of a tool which would give the user a great deal of control over the particulars of the construction of their ship -- the shape, the mass, the ways the ship might move, and so on (fig. 7). In this way the user might find some outlet for creativity and expression beyond mere aesthetic customization -- some users might build a small, snappy ship while others might explore the properties of a heavy, slow-turning craft. In order to accomplish this, I knew that a physical mode of building was probably not feasible. To this end, I started to imagine a more digitally-presented Rocket Editor. A drag and

drop interface, similar to the kinds which have been implemented in both mouse-and-keyboard and touch interfaces with ample precedent, struck me as being a suitable solution.

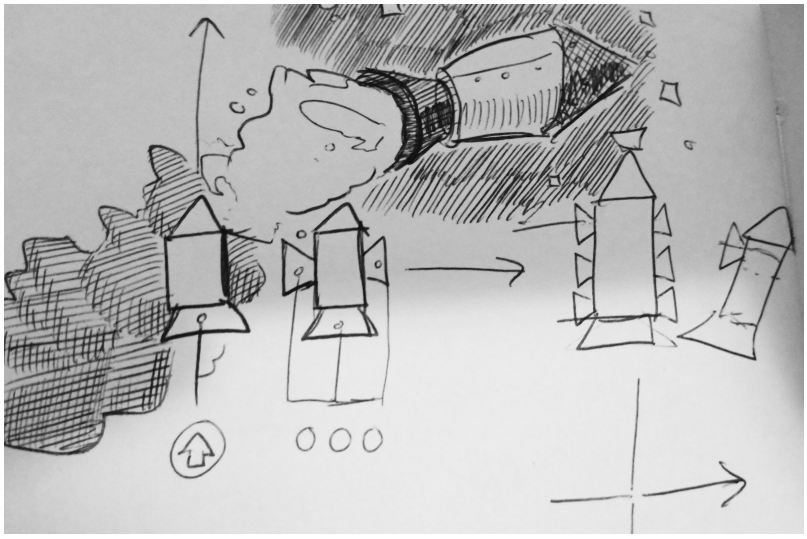

Fig. 7. Early design sketch. Image credit: Shipway, Connor. 2017.

On this basis I started to imagine platforms of deployment which would well accommodate such a solution. A simple desktop application, I decided, would provide a sensible balance between a platform which would provide a natural environment through which to manipulate a drag and drop interface, a platform that would be accessible to a wide variety of users, and a platform that would act as a foundation upon which other versions could be devised. Further, as a frequent user of desktop workstations myself, I fostered a personal preference.

With this decision in tow, I began the development process, starting with the selection of my preferred development tools.

Development Tools

The Unity 3D game development engine is the primary tool which was used in the process, and the environment in which the vast majority of development time was spent.

Secondarily, various programs included in the Adobe Creative Suite have been used at various points, most notably Adobe Illustrator in the production of vector art assets.

The choice of the Unity 3D game engine was one made partially out of pragmatism -- first, and foremost, it was the tool with which I was already familiar. However, I did not choose a tool ill-suited for the job merely because of my proficiency with it -- on the contrary, Unity is a game development environment used by hobbyists and professionals alike, and well known for its built-in physics system which would simplify the development process. This choice enabled me to spend less time realizing the game technically, allowing more time to focus on high-level design decisions.

Furthermore, a feature which has helped Unity become a widely-used tool is its ability to quickly export a game to various different platforms -- for example, PC, Mac, mobile devices, tablets, and so on. This was an appealing notion in the course of the selection process, as it posed long term possibilities for Rocket Builder as an experience which could be accessed on a wide variety of devices, therefore increasing the overall potential user base. This consideration was especially important in the context of a classroom setting, where many devices of a single given type may not necessarily be available. In this case, flexibility was paramount.

Technically speaking, the techniques used in Rocket Builder's implementation are fairly traditional. The utilization of Unity's engine and the C# programming language in Rocket Builder is quite standard, and not necessarily the objective of the research -- rather, the objective is the configuration of these systems to create a thoughtfully designed experience which could be effective for learning.

Aesthetic Style

Rocket Builder's simplistic graphical style is the result of several key considerations -- firstly, the decision to utilize such a style is one born of pragmatism. A simple graphical style can be implemented quickly, which frees up time for design and development. Since these are the qualities which were considered to be more essential as opposed to an aesthetically sophisticated art style, the choice seemed appropriate.

Secondly, the graphical style is not so flashy as to distract from the core gameplay and, correspondingly, the learning principles on display. This is to say, beyond being a helpful decision for myself in the development process, the choice of a simpler aesthetic is arguably for an appropriate for the end user, even in the absence of any sort of development time constraints.

Finally, the game's simple graphical style was primitive enough to be developed mostly within the confines of the Unity game development engine, rendering the need for external software to be quite minimal, with some exceptions.

The Adobe Creative Suite was used to create any art assets which were unable to be constructed within the confines of Unity's included primitive assets. In practice, the program used the most was Adobe Illustrator to create vector art assets, such as various UI elements such as the icons which line the top of the Rocket Editor.

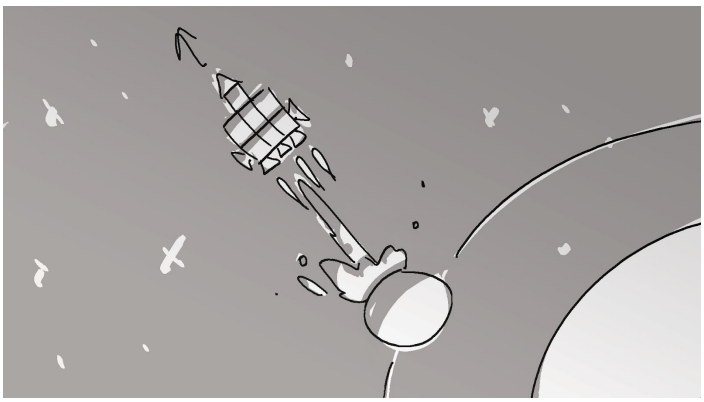

Fig. 8. Design sketch of physics environment. Image credit: Shipway, Connor. 2018.

Early Development

The first stage of development proper was devoted largely to the design of a suitable physics environment (fig. 8) in which the game session would occur, as well as a generic ship "template" which a user could control so as to navigate said environment. The utilization of the Unity "Rigidbody" physics engine made the development a rather straightforward affair. The association of keystrokes with the generation of forces which would act upon the ship, the defining of the system whereby ships would "land" on a planetary body upon collision, and the abstraction of the solar system into a model of rotating game objects -- the development of each of these features occurred in this early stage of development.

One objective which was of considerable importance at the time, was the implementation of a gravity system wherein each celestial body would emit a gravitational force proportional to its mass (fig. 9). The player, therefore, would have to compensate for the gravitational attraction of nearby bodies in the navigation and manipulation of their ship (fig. ?). This system, it was hoped, would encourage the thoughtful planning-out of routes between planets -- clever players would wait to seize the launch window or "moment of opportunity" wherein travel between two planets is especially tenable.

Fig. 9. Design sketch of rocket motion, affected by planet gravity.
Image credit: Shipway, Connor. 2018.

The promotion of thinking in this sort of way was, to some extent, a vestigial idea which originated during project's initial status as an exhibit in a museum which valued the teaching of such concepts as the launch window (the museum already featured at least one exhibit to such an end). However, in addition, the implementation of such a gravity mechanic would add nuance and sophistication to the act of controlling the ship -- making fine adjustments as the ship slung around celestial bodies would be, as it turned out, good fun.

The implementation of the gravity system was accomplished with no notable trouble, and with it, the establishment of a suitable physics foundation. With this foundation in tow, it was time to start thinking more seriously about how modular ships would be constructed and customized, so as to be eventually be deployed into the space I had developed. To this end I began development on what I called the Rocket Editor.

The Rocket Editor

Though my goal with the Rocket Editor was to devise a paradigm through which highly-customizable and expressive ships could be designed, I held simultaneously the intention to make such a process as intuitive and effortless as possible. I knew that in order to accomplish this, some clever design constraints would have to be implemented that would prevent users from getting lost in minutia, while also preserving a sufficiently broad possibility space in which thoughtful decision making could occur.

The most notable of such constraints was the implementation of the grid system, wherein each building component would be a simple square shape that would fit into a grid neatly alongside other, adjacent square shapes. While this limits the sort of precise shapes which can be created in the editor -- it's impossible to create a perfect circle, for example -- the advantage of

such a system is an effortlessly simple interface (fig. 10). There need be no provision in the editor, for example, for the precise positioning of a component, down the decimal point. Rather, the horizontal and vertical position of each block can be expressed neatly as an integer. This would encourage the broad thinking which I hoped to emphasize -- a user need not worry if their ship isn't behaving as intended due to a infinitesimal mathematical oversight.

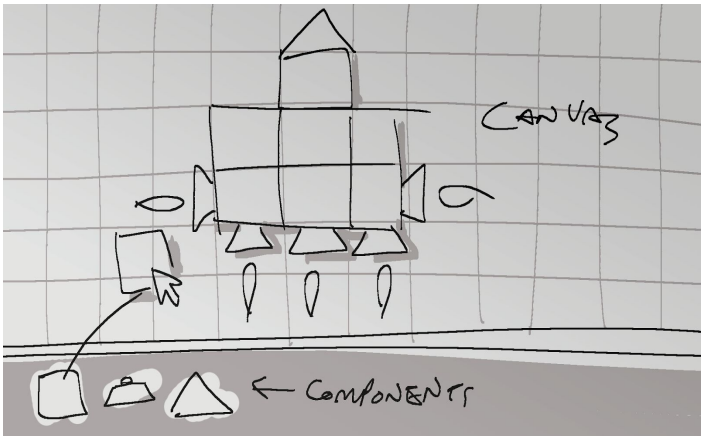

Fig. 10. Design sketch of the Rocket Editor.
Image credit: Shipway, Connor. 2018.

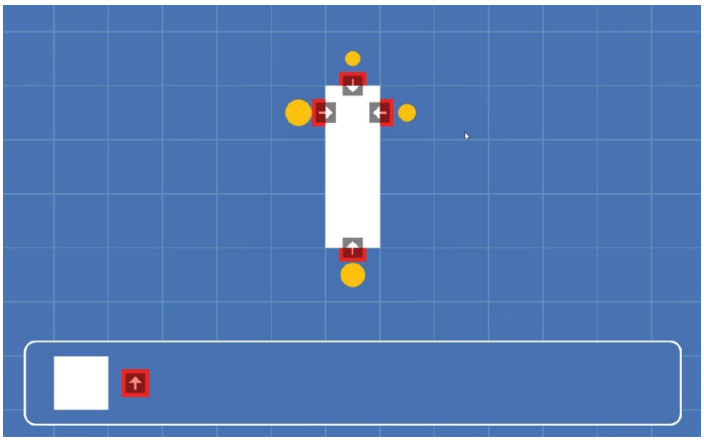

Fig. 11. Original iteration of the Rocket Editor ingame.
Image credit: Shipway, Connor. 2018.

This grid system at once began to inform what would become the "blueprint" motif which defines the Rocket Editor's visual presentation. The gridlines of the blueprint naturally

aligned with the actual grid which would constrain each square-shaped building component, and the blue color of happened to contrast nicely with the red (and white) of the components.

Development went smoothly and before long, the Rocket Editor (fig. 11) was working perfectly in tandem with the physics space, as intended. Ships of various shapes, sizes, and configurations could be designed and deployed, and their physics and mode of movement would reflect their construction and vice versa -- for example, a larger ship would take more force so as to alter its motion. Essentially a playable foundation was in place which demonstrated, in practice, the core ideas of Rocket Builder.

Equipped with this demo, I elected to spend some time sharing the concept with science educators. The work I had done so far, I hoped, would quickly convey the intent of the experienced I hoped to create, and enable experts to give me specific advice so as to accomplish my larger goals of physics education.

Honing the Learning Objectives

To this end I inquired my professors and colleagues as to candidates which might represent such a suitable expert. Happily, my studio professor Marc Russo was able to connect me with a contact of his, Professor Eric Wiebe of the College of Education. Professor Wiebe, I was told, researched the utilization of instructional technologies (including games) to the end of STEM learning. It was in my cursory familiarization with Professor Wiebe's research that I even became aware of the term "game-based learning environment", a description which I quickly assigned to Rocket Builder.

Professor Wiebe invited me to demonstrate what I had yet devised to a research group of his colleagues and students, and this meeting would go on to become one of the most valuable

resources which I utilized in the course of the Rocket Builder development. In addition to a great deal of general feedback and criticism, Professor Wiebe's research group directed me to the North Carolina Essential Standards, the document which would become the guiding foundation for all the design decisions I would make thereafter.

The Essential Standards contains the required learning goals for each section of the North Carolina schooling, and the exploration of these resources revealed the precise classification of science learning which Rocket Builder aims to impart -- namely, the Forces and Motion section of the 7th Grade Essential Science Standards (fig. 12). Up until this point I had only been able to vaguely categorize the material as being broadly "physics-related", with no precise understanding of what the exact learning goals would be or to which age group they would typically be taught.

Forces and Motion

Essential Standard and Clarifying Objectives

7.P.1 Understand motion, the effects of forces on motion and the graphical representations of motion.

7.P.1.1 Explain how the motion of an object can be described by its position, direction of motion, and speed with respect to some other object.
7.P.1.2 Explain the effects of balanced and unbalanced forces acting on an object (including friction, gravity and magnets).
7.P.1.3 Illustrate the motion of an object using a graph to show a change in position over a period of time.
7.P.1.4 Interpret distance versus time graphs for constant speed and variable motion.

Unpacking
What does this standard mean a child will know, understand and be able to do?

7.P.1.1
The motion of an object is always judged with respect to some other object or point. When an object changes position over time relative to a reference point, the object is in motion. Motion can be described with a reference direction such as North, South, East, West, up or down. The speed of an object is a measure of how quickly the object gets from one place to another.

7.P.1.2
An unbalanced force acting on an object changes its speed or direction of motion, or both. The change in motion (direction or speed) of an object is proportional to the applied force and inversely proportional to the mass. All motion is relative to whatever frame of reference is chosen, for there is no motionless frame from which to judge all motion. Friction is a force that opposes motion between two surfaces that are in contact. The amount of friction depends on factors such as the roughness of the surfaces and the force pushing the surfaces together. Newton's law describes the relationship between gravitational force, mass, and distance. An object will not start moving until a force acts upon it. An object will stay in motion forever unless an unbalanced force acts upon it. Inertia is the tendency of objects to resist any change in motion. Likewise, inertia is the reason a moving object stays in motion with the same velocity unless a force changes its speed or direction or both. *Note: Newton's Laws should not be memorized at this age. Rather, the principles which underpin the Laws ought to be well*

Fig. 12. Forces and Motion section of the Essential Standards.
Image credit: NC Department of Public Instruction. 2011.

The Forces and Motion learning goals laid out in the Essential Standards can be summarized as follows: to understand how the motion of an object can be described by its position and velocity relative to some reference point, and to understand how balanced and unbalanced forces affect (or, in the case of balanced forces, do not affect) this state of motion.

In an encouraging turn, Professor Wiebe's research group assured me that the development of a tool which would assist in the teaching of Forces and Motion was a worthwhile endeavor, going so far as to cite specific instances where school teachers had reached out to the College of Education, in search of exactly such a learning aid.

Emboldened with this knowledge and armed with the Essential Standards, I was able to return to development with a honed understanding of precisely what Rocket Builder's learning goals would be. For example, in order to more succinctly convey the idea of what the Standards refer to as reference point (North Carolina Department of Public Instruction 2), I developed a heads-up display interface which would display precise quantitative information about a rocket's position, speed, and rotation (fig. 13). The quantities displayed would be calculated relative to a reference point which is set manually by the user by clicking a particular celestial body, such as a star or planet.

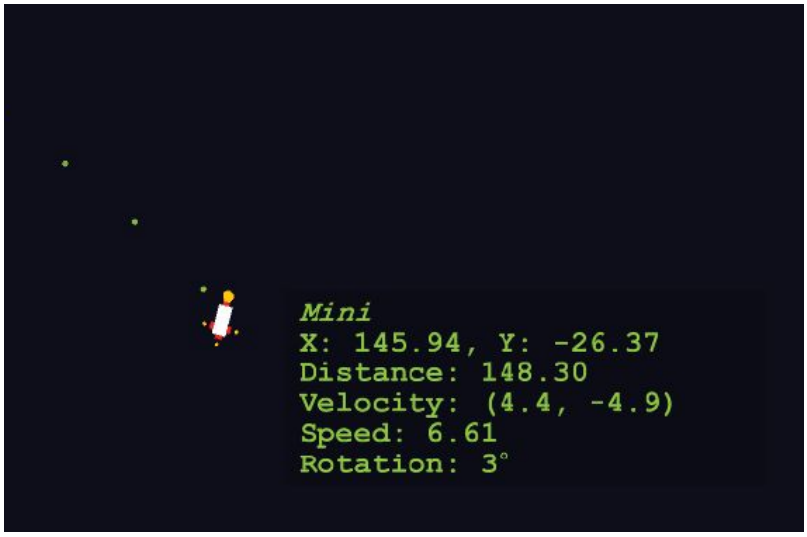

Fig. 13. Heads-up display with navigational information.
Image credit: Shipway, Connor. 2019.

Not every design decision made to this end was additive, however. The realization of a precise set of learning goals began to raise questions as to the validity of certain already-implemented features which were not particularly useful in the accomplishment of those goals. On this basis, components like the gravity system which had been implemented early in development were called into question. Though the system added depth and nuance to the control of a rocket, gravity represented an extraneous force which would constantly alter the motion of a ship regardless of a player's active manipulation. If a primary goal of Rocket Builder is to demonstrate how player-invoked forces change the motion of the ship, then the gravity system obscured this otherwise transparent relationship. For these reasons, the disabling of the gravity system was deemed prudent.

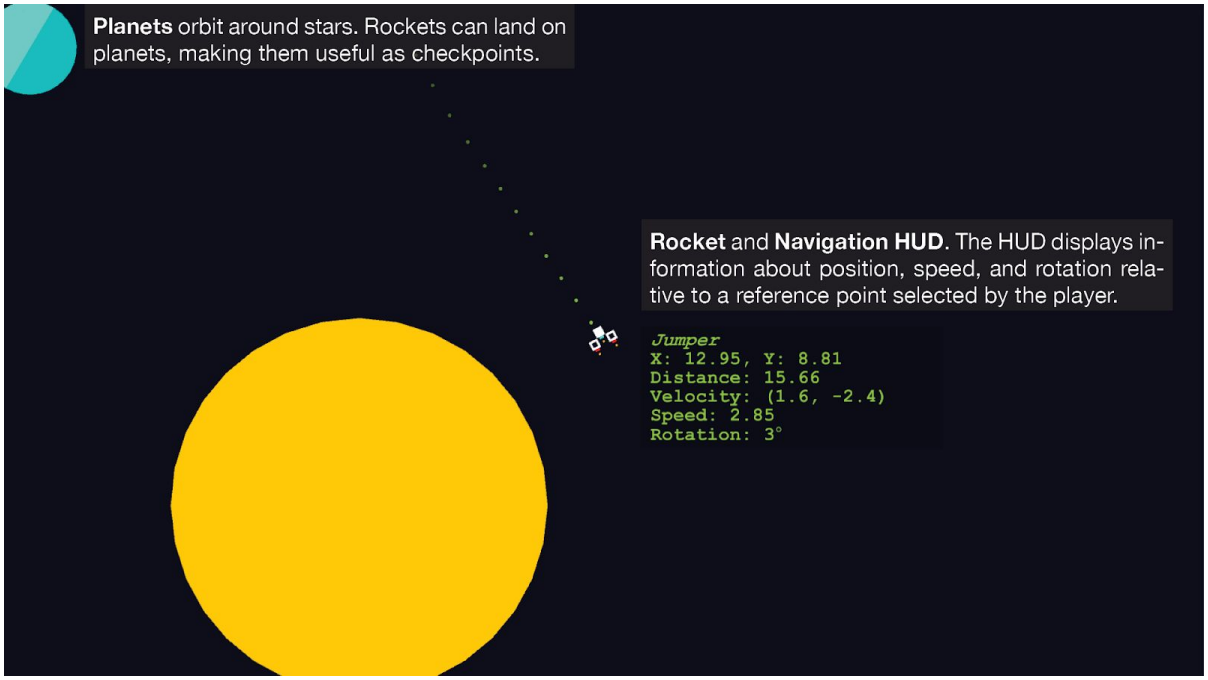

Fig. 14. Annotated screenshot of the Rocket Builder physics environment.
Image credit: Shipway, Connor. 2019.

Exploring a Tutorial

Toward later stages in development, experimentation was conducted as to the prospect of a narrative tutorial sequence, which would, through a story-driven experience, verbally explain specific learning concepts to the player. This seemed especially reasonable in the context of Mayer's research into Multimedia Learning, a topic discussed earlier in this paper. If the teacher represents the verbal component of Rocket Builder's multimedia framework, and the game represents the pictorial component, then the addition of a dialogue-based tutorial would effectively fulfill a similar role to the one the teacher would normally fulfill. In this way, Rocket Builder could perhaps minimize the need for an instructor to be present, gaining potential to behave even as an entirely self-contained learning tool.

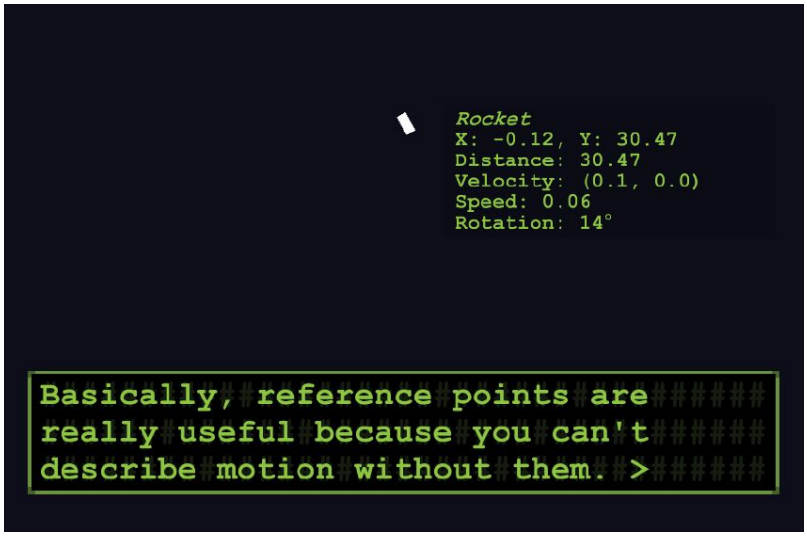

Fig. 15. Tutorial dialogue system. Image credit: Shipway, Connor. 2019.

An interface was developed which would progressively display snippets of dialogue (fig. 15), sometimes only advancing when certain conditions or criteria had been met -- for example, if the player had successfully set the reference point to a certain star, as they had been instructed to do. It was only after some time spent developing these ideas that I decided the tutorial experience felt undeniably dreary to progress through. I tried to imagine myself as a student, and my impression was that lines of prose would ineffectively command attention when the promise of building a rocket was looming on the horizon. However, more importantly, I felt that this sort of explicit tutorial was, perhaps, at odds with the fundamental ideas that I wanted to explore through Rocket Builder -- the utilization of Bogost's procedural rhetoric to convey sophisticated ideas, and the potential of video games as a teaching tool.

Supposing a tutorial like the one described above was implemented, in the event of the game's successfully conveying the learning concepts, it would be difficult to gauge which aspects of the game were responsible for the success -- the mechanics-driven procedural rhetoric, or the more traditional multimedia aspects? Ultimately I elected to forego the tutorial concept

and emphasize gameplay as the primary vehicle for learning, conceding that new players might have to spend a bit of extra time making sense of the game world.

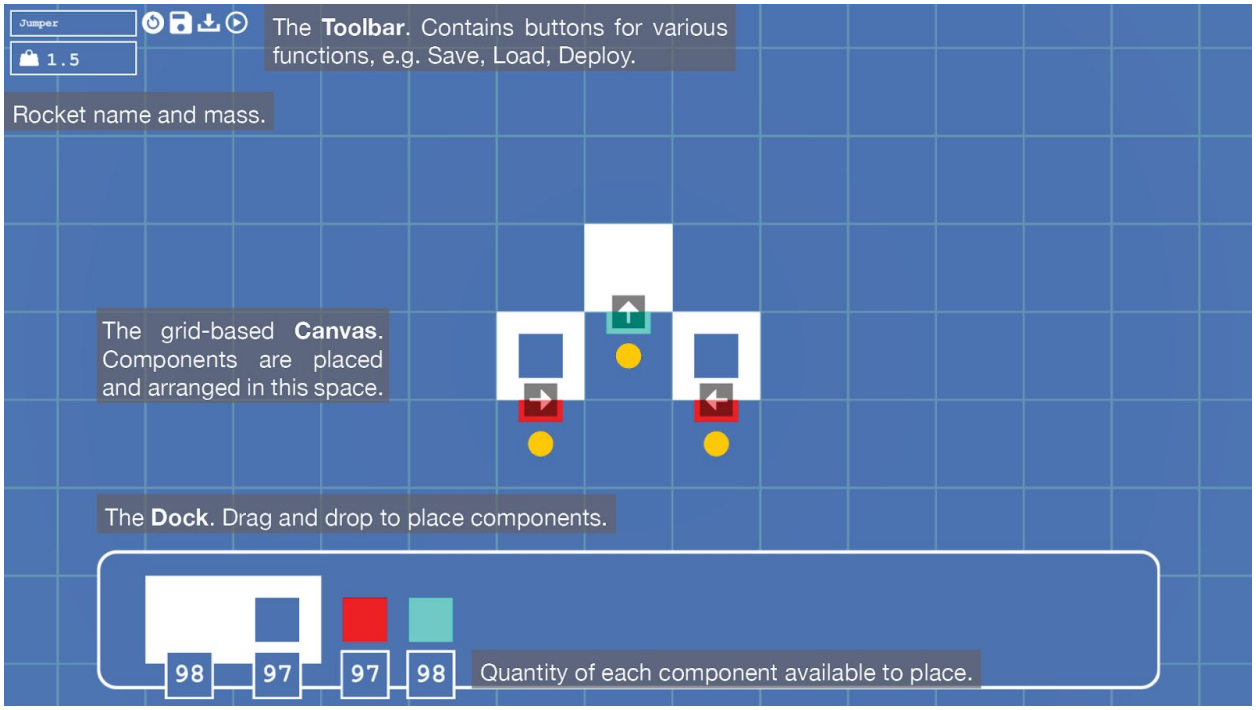

Fig. 16. Annotated screenshot of the Rocket Editor.
Image credit: Shipway, Connor. 2019.

Polish and Minor Improvements

A final stage of development was spent in the pursuit of minor quality-of-life improvements which would reduce the occurrence of elements which might frustrate the user, as well as helpful features which might expand usability. This stage included the addition of various interface improvements, such as a toolbar in the Rocket Editor, as well as the ability to save and load ships locally in Rocket Builder's game files, enabling the persistence of ship designs between sessions. Keeping in mind the socially-inspired functionality of one of Rocket Builder's influences, Algodoo, I designed saved ships to be expressed as simple and easily accessible text files.

Informal User Testing

Towards the end of the development process, I conducted some informal tests with close family members in an attempt to understand whether my design decisions were facilitating an effective learning experience, and to inform what sort of additions or changes might comprise future efforts at development. The informal nature of these tests should not be understated -- the goal of these sessions was by no means to collect rigorous data-based results, but rather to gather some loose anecdotal impression of how students might interface with the game. This, I hoped, would give me some glimpse as to the game's effectiveness. The results were often encouraging, and always informative.

Both users found some degree of engagement with the game's basic concept of constructing and piloting a rocket. A common aspect of this engagement was a penchant for mischief and destruction, a favorite feature being the fiery explosion that occurs whenever a rocket collides with the sun. User 1 spent upwards of a minute in an attempt to get her craft moving as quickly as possible, keeping a watchful eye on the heads-up display and its depiction of units travelled per second. In addition to harboring a desire to simply play, both users eventually became interested in taking on some larger goal to accomplish. User 1 verbally asked what her goal should be for the game session, whereas User 2 invented a goal of his own: to name on each planet in the solar system before finally returning to his starting point.

A particularly encouraging moment entailed the discovery by User 2 of the direct relationship between the mass of a rocket and the amount of force required to alter its motion. A bit unsatisfied with the sluggish responsiveness of his rocket, he returned to the Rocket Editor and designed a ship that was lighter while retaining a similar shape and configuration of

movement. Upon redeploying the craft, he verbally expressed his satisfaction with the lighter rocket, confirming that it responded more snappily, as he had intended.

A matter of interest which surfaced in each session was the general disregard for the arbitrary ways I might "intended" for rockets to be designed -- if the interface constraints of the Rocket Editor permitted a certain configuration or technique, that technique would be utilized. Sometimes this demonstrated an outright mistake in my programming, which I noted I would have to fix later at a later point. Another time, it created unexpected avenues for creative expression. For example, User 2 stacked multiple thrusters one on top of the other, functionally creating one especially powerful thruster which would impart a great deal of force almost instantly. I was impressed by such ingenuity; it occurred to me User 2 was demonstrating his mastery of the game's systems in a way I had not thought to do myself. I might argue that this sort of play actually reveals a successfully internalized understanding of those learning concepts which Rocket Builder seeks to impart.

Moving Forward

Today, Rocket Builder exists as a self-contained application which demonstrates a capacity as a useful pedagogic tool for Forces & Motion learning. After thoughtful deliberation as well as discussions with peers, I would suggest that Rocket Builder possesses even more potential when considered as a platform, on which to be further built and developed.

The opportunities through which Rocket Builder could be developed are numerous. Though the Rocket Builder engine currently only supports a single player per game session, the socially engaging, synchronous multiplayer experience described by Rocket's Builders initial realization as a museum exhibit remains a viable possibility. Beyond the synchronization of a

single classroom of players over a local network, the possibility of connecting disparate users from all over the world warrants a discussion in and of itself.

A feature which has emerged as being especially pertinent, in light of discussions with peers as well as observations gleaned during informal user testing, is the implementation of some goal-focused system for quantifiable ingame progression. The decision to present Rocket Builder as a more open-ended or toylike experience was one made early in the design process, in an attempt to avoid some of the more constraining aspects of gamification which might distract from the learning goals. However, as development progressed, the potential benefits of a more focused approach became apparent -- while a linear game design is not strictly necessary, the presence of some sort of goals to accomplish, or specific victory conditions to achieve, is a feature which some users have come even to explicitly expect when they are presented with an experience they understand as being a game. Happily, this is not in the least at odds with the impartation of Rocket Builder's learning goals -- on the contrary, I would posit that the most meaningful learning can take place under the context of specific constraints which demand a mastery of the game's systems in order to fulfill.

To this end, a system which enables the voluntary imposition of such constraints in exchange for rewards (if little else than bragging rights) has been discussed. This would likely take the presentation of a "mission" system through which players could tackle certain challenges -- for example, navigate to a certain planet while exhausting only a certain quantity of fuel (an activity which would require a clever utilization of the property of inertia), or, match the velocity of a particular planet for five seconds (an activity which would require an understanding of reference points). Customized missions could even be user-defined, providing opportunities

for educators to gauge students' understanding of certain learning concepts or for game enthusiasts to challenge one another with difficult scenarios, and could be shared in a similar manner as saved ship designs. This sort of system could lend itself to an ecosystem of shared user content which fosters a rich variety of teaching techniques, much like those showcased by the communities which have embraced *Algodoo* as a platform for physics simulation.

To elaborate on this mission system, let us consider the example above -- an instructor wants to gauge his classroom's understanding of the concept of inertia. To this end he decides to design a mission which will test students' ability to apply inertia as a solution to a particular puzzle. He opens the Mission Editor, where he is able to select from a variety of modifiers, limitations, and victory conditions, so as to design a mission which will be effective to his ends (fig. 17).

Fig. 17. Concept design for the Mission Editor.
Image credit: Shipway, Connor. 2019.

Our imaginary instructor designates the mission objective as Traversal -- the goal is simply to navigate to one or more particular planets. He selects Mars as the destination -- when a player lands on the planet Mars, the mission will be considered successful (perhaps a congratulatory message will be displayed). Noting the various modifiers available for his manipulation, he ponders “Fuel Limit” as being useful. Setting a limit on the amount of fuel at a player’s disposal would, effectively, necessitate efficient manipulation of forces upon the craft. A player could not “brute force” their way to a given destination by continuously making adjustments to the rocket’s motion. Rather, they would need to use their limited fuel to merely set their craft upon a course, utilizing the property of inertia to maintain that course. On this basis, our instructor sets a Fuel Limit of 200 liters. He quickly plays the mission, testing whether this quantity seems appropriate -- it seems a bit generous, as a “brute force” solution is still fairly tenable. Returning to the editor with a keystroke, he adjusts the quantity to 100 liters.

With his mission criteria set, the instructor names his mission “Inertia Quiz” and saves the mission. Now he can distribute the mission freely to his students, and each student can proceed to load and play the mission in each of their respective game clients.

The options by which Rocket Builder could be built upon are numerous -- a more-developed open world, additional building components, and deployment to a wider range of platforms (such as mobile devices and tablets) all spring to mind as obvious candidates, and the sorts of learning which each of these features could accommodate are various in their scope and nature.

SIGNIFICANCE OF WORK

The engaging nature of video games, and the affinity which modern students have developed for such games, is undeniable. To this end, many applications have been developed which frame learning material in the context of a video game, but arguably the utilization of the medium has, in many cases, been only as just that -- a frame, and little else.

The utilization of games as teaching tools should not a concession begrudgingly made, a decision made in an attempt to trick students into stumbling into the “real” material -- material wedged in between uninspired sections of gameplay, meant to serve as little more than bait for unwitting youths. The learning material can, in many cases and for many subjects, be expressed both intuitively and entertainingly through the game itself. The Forces & Motion concepts illustrated in Rocket Builder is just one simply-realized example of this, but ambitious designers and educators could use the same frameworks and techniques to illustrate evermore sophisticated and increasingly abstract concepts. Games, thoughtfully designed in this way, have a potential to be truly effective teaching tools in their own right.

REFLECTIONS

Rocket Builder found its beginnings as little more than an interesting concept by which I hoped to satisfy the requirements for a course project, but even in those early stages I think the application's potential to be something greater captivated me. Perhaps it was that regard for the sheer versatility of the idea that inspired me to pursue Rocket Builder further in the form of my graduate research. Though the platform of delivery, techniques of development, and even the learning goals of the project have changed over the course of its development, the essence of the project and its foundational idea have always remained the same -- to explore the potential of video games as a learning tool. For one such as myself who has enjoyed such games since an early age, it would be a waste, I think, to limit the broad potential of such an abundantly expressive medium to entertainment alone. Only time will tell what the future holds for games, but it is my hope that Rocket Builder may play some small part in a bright future for the craft.

GLOSSARY

Balanced force: In Forces and Motion, as opposed to an unbalanced force, a force which has no equal and opposite counterpart and will therefore alter the motion of the object to which it has been applied in some way.

First-person shooter: A genre of video game which presents as its primary mechanic the manipulation of a projectile-flinging device so as to overcome obstacles and traverse a three-dimensional space. Often designed as to facilitate combat, as in *Doom*, but occasionally nonviolent, as in the puzzle game *Portal*.

Inertia: In Forces and Motion, the tendency of an object to preserve its own state of motion, in the absence of forces which would alter that motion.

Launch window: In space flight, the "moment of opportunity" wherein travel to another given celestial body is possible.

Mechanic: In game design, a rules-based element of gameplay which acts as a constituent part of what can actually be considered the essence of a game. A mechanic could be, for example, the ability for a player to capture his opponent's pieces in chess.

Multiplayer: As opposed to single player, a type of game session in which multiple players interact as active agents.

Physics game: A video game which implements systems that simulate and analogize some aspect of physics so as to enable some primary mechanic of the game.

Platformer: A genre of video game which presents as its primary mechanic the navigation of the player character through an obstacle course -- often, eponymous platforms precariously poised over a bottomless pit.

Reference point: In Forces and Motion, an object or point by which the motion of some other object is judged

Rhetoric: As considered by Ian Bogost in the context of his theory of procedural rhetoric, the notion of elegance, clarity, and creativity in communication.

Single player: As opposed to multiplayer, a type of game session in which only a single player acts as an agent.

Unbalanced force: In Forces and Motion, as opposed to an balanced force, a force which has an equal and opposite counterpart and will therefore not alter the motion of the object to which it has been applied.

WORKS CITED

Algodoo Forum, *Algoryx*, www.algodoo.com/forum.

A web forum devoted to the discussion of the Algodoo simulation tool and sharing of scenes and contraptions created with the tool.

Bogost, Ian. "The Rhetoric of Video Games." *The Ecology of Games: Connecting Youth, Games, and Learning*, edited by Katie Salen, The MIT Press, 2008, pp. 117-140.

A discussion of what Bogost refers to as procedural rhetoric -- essentially, rhetoric which expressively communicates ideas through processes and systems. Bogost argues that video games are capable of making powerful arguments through the utilization of procedural rhetoric.

Çelik, Harun, et al. "Evaluating and Developing Physics Teaching Material with Algodoo in Virtual Environment: Archimedes' Principle." *International Journal of Innovation in Science and Mathematics Education*, 2015, pp. 40-50.

An analysis of the effectiveness of Algodoo as an aid in the teaching of Archimedes' Principle. The results of the study suggest that the use of Algodoo can have a positive impact on students' learning.

Kolb, David A. *Experiential Learning: Experience as the Source of Learning and Development*. Prentice Hall, 1984.

A discussion of Kolb's theory of experiential learning -- essentially, the idea that learning is the result of the transformation of experience. Kolb goes on to describe a four-stage process which explores his idea of the process in greater detail.

Mayer, Richard E. *Multimedia Learning*. 2nd ed., Cambridge University Press, 2009.

A book on Mayer's research into multimedia learning and his "principles" of multimedia design -- for example, the Multimedia Principle, which states that words and pictures together more effectively convey a concept than words alone.

Nakamura, Yasuyuki, et al. "Concurrent Use of Demonstrations and Simulations for Teaching of Basic Physics." *Proceedings of the 12th Asia Pacific Physics Conference*, 2014.

A study of the usage of Algodoo as a simulation tool so as to supplement standard physical demonstrations. The motion of a ball rolling down various inclines was chosen as the topic to be learned by various students. The results suggest that the simulations aided in students' understanding of the concepts.

North Carolina Department of Public Instruction. *Essential Standards: Grade 7 Science*. 2011.

The essential points of knowledge and understanding which is to be considered to standard for science education in North Carolina 7th grade classrooms.