BlocksVR

by

Collen Byer and Deyshawn Bond

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

Insert Your Name Here	4/17/17 Date
Insert Your Name Here	4/17/17 Date
Insert Faculty Advisor's Name Here, Faculty Advisor	4/17/17 Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services
November 2016

TABLE OF CONTENTS

FIGURES

TABLES

Abstract

BlocksVR mixes the old with the new, bringing the classic game of Tetris to an alternate environment where the player physically stacks "raining" blocks. Our game capitalizes on the intuitive design and feel of the HTC Vive to give a true alternate reality experience. Creating a BlocksVR game brings back memories of competitiveness and the nostalgia of completing and destroying rows to attempt to beat their high score. Players familiar with Tetris are able to dive directly into gameplay with only a few simple instructions. Users will be grab and place blocks intuitively with only a slight learning curve thanks to the full body tracking system and ergonomic hardware the Vive offers. Enjoy a retro game with modern technology to experience familiar, yet new, addictive gameplay. BlocksVR brings fresh 3D assets to a classic game to help it look new and feel amazing.

Problem Statement

Introduction 1.1
Virtual reality is a new technology with endless potential application, yet still an infant in the world of technology. Our product brings new breath to an old game with the goal of capturing the attention of existing HTC Vive owners and to gain the curiosity of potential owners. By renovating a game that many people are familiar with, we show people the capabilities of virtual reality to gain their interest and broaden the market. Ultimately, this gives other developers a reason to develop new software products for this technology.

Project Description 1.2
Play Tetris in a virtual world. Using the HTC Vive's room-scale feature, our game allows users to move and interact with blocks that fall from the "sky". Grab the blocks, orient them to fit, and stack them on the table in your play area. This is a modern rendition of the classic game. Our game has all core features of Tetris along with additional features to offer a fresh take.

Problem 1.3
A current problem with virtual reality, in general, is the lack of software. The absence of software is due largely to being new to the market and not yet being adopted by the masses, including developers. Because the market is not yet large, developers target areas where they see the most potential instead of innovating with a new technology.

Solution 1.4
Expanding on what was stated previously, BlocksVR breathes fresh life into an old title. By targeting the nostalgia factor, we capture the interest of a new demographic of players. This brings in new players and, hopefully, consumers of virtual reality.

User Profile 1.5
The users of this game are players, all with the same role, who are capable of using a VR space with the HTC Vive. The users must move throughout a play area with the HTC Vive controllers to interact with the game and its components. Movement is a requirement for both in game play and interfacing with menus. Movement required by the user consists of walking, rotating arms, and wrist.

Figure 1: User Profile

Potential Users 1.4.1 • The Player
Software and Interface Experience 1.4.2 • The player will have had at least some experience with virtual reality, preferably the HTC Vive and its controllers.

• Having prior experience playing traditional Tetris will be beneficial, but is not required.
Experience with Similar Applications 1.4.3 • Players will likely have played classic Tetris • Players have potentially used other VR applications
Task Experience 1.4.4 • Players must be capable of moving and being active in an open space
Frequency of Use 1.4.5 • This is subject to the player, but could potentially be used multiple times in a single day.
Key Interface Design that the Profile Suggests 1.4.6 • A movable experience to interact • Encompassed in an alternate environment

Project Management

Budget 2.1

Figure 2, below, is an accurate representation of the cost of our project. Since we are treating this as our product our only real investment is time. Our initial cost was high because we did not own these items prior to starting this project.

Table 1: Budget

Item	Quantity	Price / item	Total
Custom built PC	2	$1250	$2500
HTC Vive	2	$800	$1600
*Unity Assets *ones not created by us*	1	$5	$5
TOTAL			
$4105			

Objectives/Deliverables 2.2

The following chart represents the project sprint deadlines.

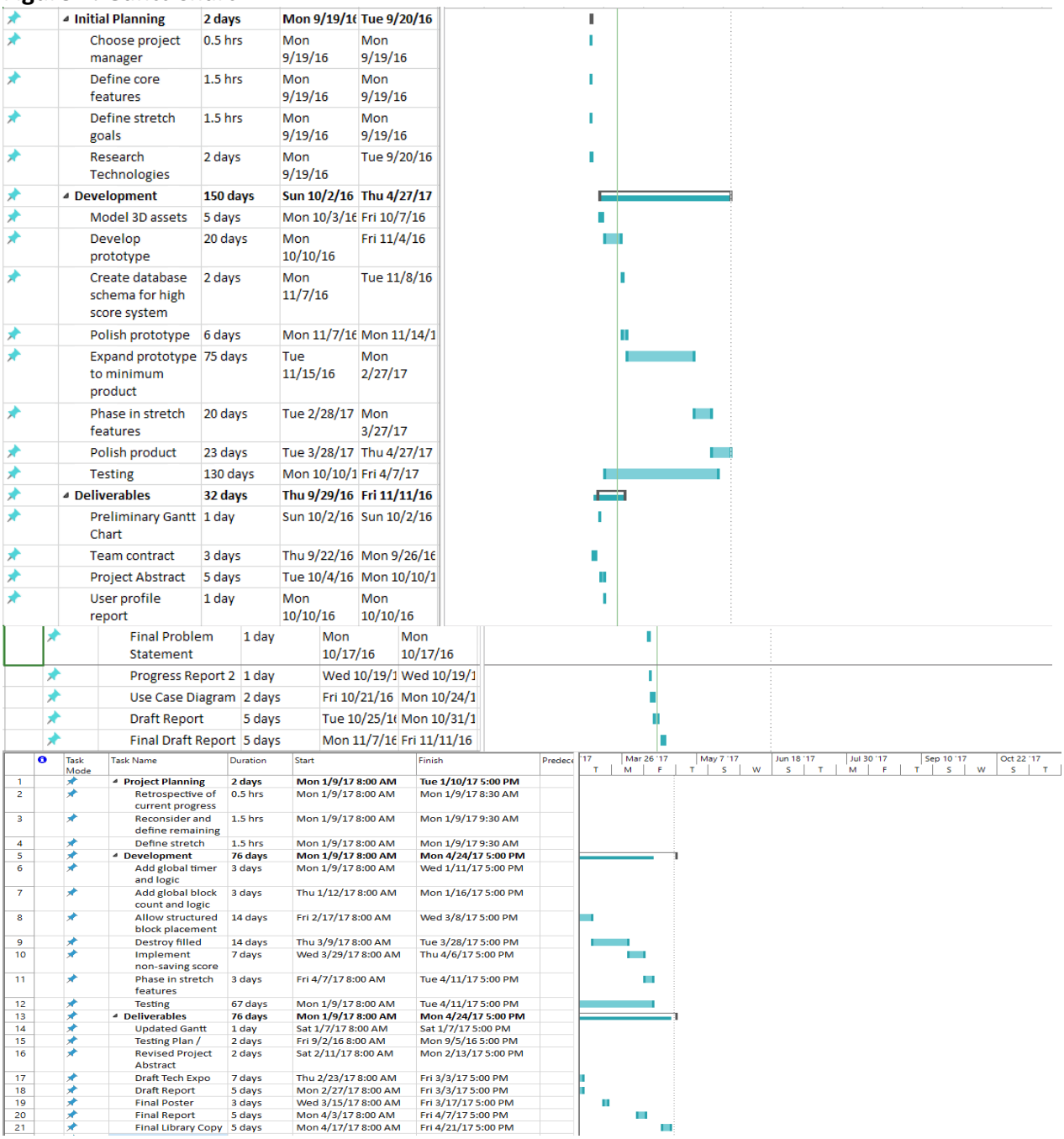
Table 2: Project Sprint Task Due Dates

BlocksVR Project Sprint Deadlines				
Research and Planning Goal	9/28/16		Coding and Presentation Prep Goal	11/8/16
Begin Project Setup Goal	10/8/16		Finishing Touches to Presentation	11/18/16
Get Assets and Begin Prototype Goal	10/19/16		Complete Foundation of Game Goal	11/30/16
Implement core game logic (will continue until testing phase)	12/01/16		Prepare for Tech Expo	04/01/17
Testing / finishing touches	03/01/17			

Project Schedule 2.3

Figure 4 is a Gantt chart that provides a timeline of our initial planning, development process, and physical deliverables from September 2016 through April 2017.

Figure 2: Gantt Chart



	Task Name	Duration	Start	Finish
1	**Initial Planning**	**2 days**	**Mon 9/19/16**	**Tue 9/20/16**
	Choose project manager	0.5 hrs	Mon 9/19/16	Mon 9/19/16
	Define core features	1.5 hrs	Mon 9/19/16	Mon 9/19/16
	Define stretch goals	1.5 hrs	Mon 9/19/16	Mon 9/19/16
	Research Technologies	2 days	Mon 9/19/16	Tue 9/20/16
	Development	**150 days**	**Sun 10/2/16**	**Thu 4/27/17**
	Model 3D assets	5 days	Mon 10/3/16	Fri 10/7/16
	Develop prototype	20 days	Mon 10/10/16	Fri 11/4/16
	Create database schema for high score system	2 days	Mon 11/7/16	Tue 11/8/16
	Polish prototype	6 days	Mon 11/7/16	Mon 11/14/1
	Expand prototype to minimum product	75 days	Tue 11/15/16	Mon 2/27/17
	Phase in stretch features	20 days	Tue 2/28/17	Mon 3/27/17
	Polish product	23 days	Tue 3/28/17	Thu 4/27/17
	Testing	130 days	Mon 10/10/1	Fri 4/7/17
	Deliverables	**32 days**	**Thu 9/29/16**	**Fri 11/11/16**
	Preliminary Gantt Chart	1 day	Sun 10/2/16	Sun 10/2/16
	Team contract	3 days	Thu 9/22/16	Mon 9/26/16
	Project Abstract	5 days	Tue 10/4/16	Mon 10/10/1
	User profile report	1 day	Mon 10/10/16	Mon 10/10/16
	Final Problem Statement	1 day	Mon 10/17/16	Mon 10/17/16
	Progress Report 2	1 day	Wed 10/19/1	Wed 10/19/1
	Use Case Diagram	2 days	Fri 10/21/16	Mon 10/24/1
	Draft Report	5 days	Tue 10/25/16	Mon 10/31/1
	Final Draft Report	5 days	Mon 11/7/16	Fri 11/11/16

	Task Mode	Task Name	Duration	Start	Finish	Predec
1		**Project Planning**	**2 days**	**Mon 1/9/17 8:00 AM**	**Tue 1/10/17 5:00 PM**	
2		Retrospective of current progress	0.5 hrs	Mon 1/9/17 8:00 AM	Mon 1/9/17 8:30 AM	
3		Reconsider and define remaining	1.5 hrs	Mon 1/9/17 8:00 AM	Mon 1/9/17 9:30 AM	
4		Define stretch	1.5 hrs	Mon 1/9/17 8:00 AM	Mon 1/9/17 9:30 AM	
5		**Development**	**76 days**	**Mon 1/9/17 8:00 AM**	**Mon 4/24/17 5:00 PM**	
6		Add global timer and logic	3 days	Mon 1/9/17 8:00 AM	Wed 1/11/17 5:00 PM	
7		Add global block count and logic	3 days	Thu 1/12/17 8:00 AM	Mon 1/16/17 5:00 PM	
8		Allow structured block placement	14 days	Fri 2/17/17 8:00 AM	Wed 3/8/17 5:00 PM	
9		Destroy filled	14 days	Thu 3/9/17 8:00 AM	Tue 3/28/17 5:00 PM	
10		Implement non-saving score	7 days	Wed 3/29/17 8:00 AM	Thu 4/6/17 5:00 PM	
11		Phase in stretch features	3 days	Fri 4/7/17 8:00 AM	Tue 4/11/17 5:00 PM	
12		Testing	67 days	Mon 1/9/17 8:00 AM	Tue 4/11/17 5:00 PM	
13		**Deliverables**	**76 days**	**Mon 1/9/17 8:00 AM**	**Mon 4/24/17 5:00 PM**	
14		Updated Gantt	1 day	Sat 1/7/17 8:00 AM	Sat 1/7/17 5:00 PM	
15		Testing Plan /	2 days	Fri 9/2/16 8:00 AM	Mon 9/5/16 5:00 PM	
16		Revised Project Abstract	2 days	Sat 2/11/17 8:00 AM	Mon 2/13/17 5:00 PM	
17		Draft Tech Expo	7 days	Thu 2/23/17 8:00 AM	Fri 3/3/17 5:00 PM	
18		Draft Report	5 days	Mon 2/27/17 8:00 AM	Fri 3/3/17 5:00 PM	
19		Final Poster	3 days	Wed 3/15/17 8:00 AM	Fri 3/17/17 5:00 PM	
20		Final Report	5 days	Mon 4/3/17 8:00 AM	Fri 4/7/17 5:00 PM	
21		Final Library Copy	5 days	Mon 4/17/17 8:00 AM	Fri 4/21/17 5:00 PM	

Technical Elements

HTC Vive 3.1

HTC Vive is a virtual reality system that uses lighthouse sensors to track movement in a user-defined space called a play area. The headset takes in light distributed from the two lighthouses to guide the movement of the player. Handheld controllers, also tracked by the lighthouses, give the user a new element to actually have a wide variety of control over any gaming environment.

Unity 3.2

Unity is the game engine being used to develop the game. A free tier is available for personal use. A library called “SteamVR” is being used to assist in development. We have also obtained some free assets from the Unity Asset store, along with modeling our own.

Blender 3.3

Blender is an open source modeling software that is being incorporated to model assets, such as the blocks.

Problems Encountered 3.5

The main problem that was faced throughout development was the fact that virtual reality is such a new technology. Documentation does exist for it, but it is limited. Often times we would experience issues that others had not come across yet and have no way of resolving the solution without struggling with it. A huge positive to this is that we were forced to adjust and overcome which helped us learn a lot. Aside from eventually figuring it out on our own, we reached out to our experts at University Hall and also reached out to experienced VR developers on the internet via Twitter.

Application

User Interface 4.1

Upon launch of the application, the user is prompted with a start screen that with three selections

- Start Game
- High Scores
- Settings

Start game initializes a new game starting at round 1.

High scores allow the user to view the high score of their current play session. These do not persist.

Settings allow the player to configure various in game aspect. The configurable features are as follows:

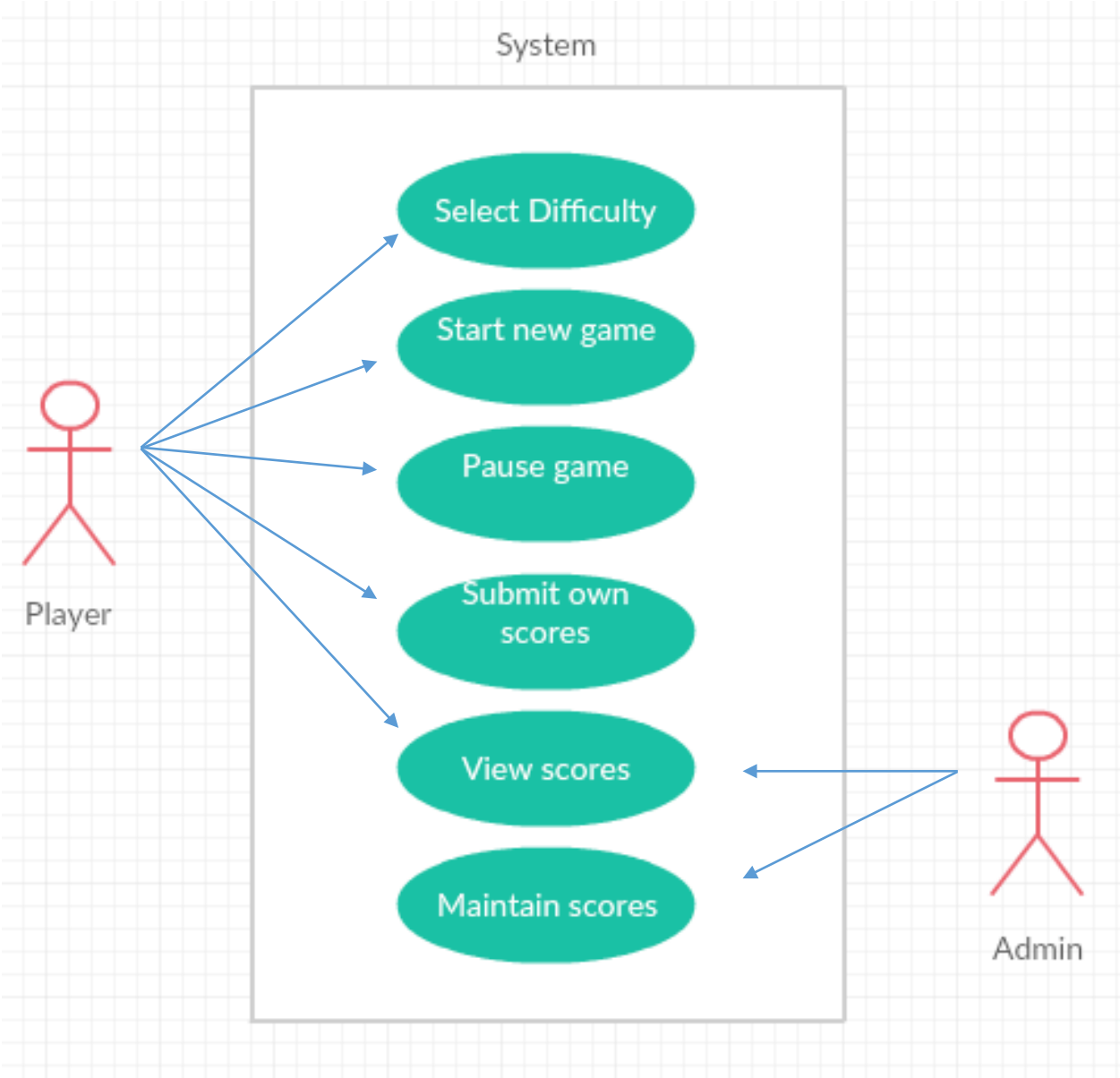
- Game sound
- Difficulty (Increase in fall speed, block count decrease, block time allowance on table)
- Background music
- Environment surroundings

Note: *As these screens are implemented, we will add screenshots to this document.*

Use Case Diagram 4.2

Figure 5 represents the Use Case Diagram for the game. There is only one main user, the player, which will be able to interact and play with the complete front end user interface. The admin user, or the developer, is able to view the application's source code for any type of bug fixes that might need to be done.

Figure 3: Use Case Diagram



Testing

Overview 5.1

When developing a game, testing is paramount. Like any software, a buggy experience will lead to unsatisfied users. In a game, where a player is likely being tested on various skills and performance, it can cause extreme frustration. This can directly affect the success of a gaming product. As developers, we will use testing to ensure that our game is as bug free as our eyes, and other trusted tester's eyes (e.g. family and friends), can see. Our goal is to be as confident as we can be in an initial release. When inevitable bugs get reported, assuming we can replicate it, they will be our top priority and we will release patches to address them ASAP.

Scope 5.2

The scope of testing Blocks VR will be to thoroughly inspect all features of the gameplay. A list of the features are as follows:

- Block Counter
- Timer
- Accurate block interaction (catching, rotating, collisions)
- Accurate block placement
- Accurate removal of completed rows
- Score
- Sound implementation

Objective 5.3

The objective of our testing is to ensure the playability of the game meets our standard of development and core features are implemented the way we intended for them to be. As features are created, the developer should go through every use case / scenario to try to "break" the code.

Testing Procedures 5.4

Procedures are straightforward, as new ideas come to fruition, we discuss in detail how these ideas / features should work and be implemented into the game. The hypothetical logic that we discuss as an idea is then translated into code. These features are noted in GitHub as issues along with the criteria for them to be a working feature.

As stated previously, the developer that implements the feature is responsible for going through every case listed in the GitHub issue until he/she feels that it is complete. At this point, other developers will be notified of the implemented feature and will then test and fix (or not fix) anything the first developer may or may not have missed. Upon review of all developers, this feature can then be merged into the main build of the game.

Finally, for further testing we request the participation of family and friends. We use these testers as a fresh set of eyes. They're on the outside looking in, which gives them the advantage

of critiquing features or implementations that we might not see. Ultimately, these types of casual users are the player-base that we are seeking. So their constructive feedback, along with finding bugs, is completely invaluable throughout the development process.

Conclusion

Fall Semester 6.1

Throughout the Fall semester of the 2016 – 2017 academic year, a functioning prototype was completed. The user could interact with blocks by picking up and placing them. The movement felt intuitive and life-like. Game logic had not yet been implemented.

Spring Semester 6.2

As the academic year winded down, our prototype has been shaped into a polished product with features as follows:

- Intuitive interaction with blocks in a VR space
- Game logic true to classic Tetris that incorporates new features
- Missed block count
- Block-on-table time allowance
- A high score system
- Setting configuration

With the completion of these key features, we have, what we consider, a complete and successful project where all goals have been met.

References

OpenVR SDK
An API and runtime that allows access to VR hardware from multiple vendors without requiring that applications have specific knowledge of the hardware they are targeting. This repository is an SDK that contains the API and samples. The runtime is under SteamVR in Tools on Steam.
https://github.com/ValveSoftware/openvr

Repo by Sean Lee
This is a repository that supports a YouTube tutorial that was created by Sean Lee. This was referenced when setting up the controllers and grabbing objects in BlocksVR.
https://github.com/b0ard/YoutubeVive

Appendix A

Assets

BlocksVR took advantage of free assets in the Unity Asset Store. The environment that exists was made possible by an asset package titled “Simple Home Stuff” by artist “Mohelm97”. The following are links to the artist’s profile and the “Simple Home Stuff” package.

Package: https://www.assetstore.unity3d.com/en/#!/content/69129

Profile:https://www.assetstore.unity3d.com/en/#!/search/page=1/sortby=popularity/query=publisher:23076

Poster

A document created to represent BlocksVR at the 2017 University of Cincinnati Tech Expo.