

# EE 267 Project Proposal

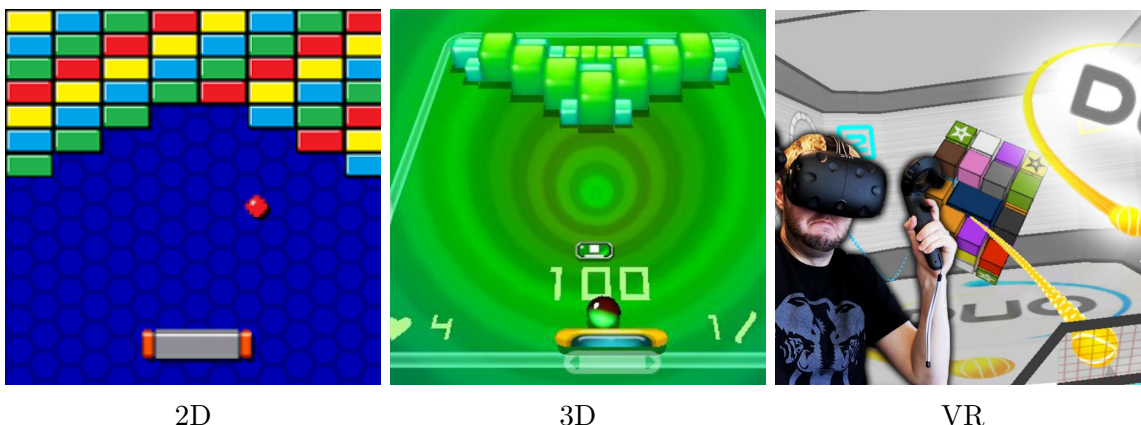
Dae Hyun Kim (dhkim16@stanford.edu)  
Marcus Pan (mpanj@stanford.edu)

## 1 Motivation

Our project's goal is to create a VR Brickbreaker game with a novel input device. In this game, there is a collection of brick objects in front of a player, and a bouncing ball. The player can control a paddle, and the objective is to bounce the ball off the paddle toward the bricks in the scene. The bricks disappear after being hit by a ball, and the player loses iff the ball flies past the player. The player wins when all the bricks have disappeared.

We propose creating the game environment for VR, with a novel input device for the paddle. The device would be a physical object with sensors that can track translation across a surface, and orientation. This device, combined with the VR scene would create an immersive and realistische gaming experience.

## 2 Previous Work



The primitive version of brickbreaker is simply a 2D scene, and a paddle that can only translate along the x-axis. 3D versions of the game have been developed as well, which are basically 2D versions of the game with 3D graphics.

Then there have been VR versions of the game such as *Brick Breaker Heaven* for the HTC Vive. These incorporate 3D motion of the ball, and use the input from the HTC Vive Controller to track the pose of the paddle. The handheld input device contains IMU sensors to track the orientation of the device. It also contains buttons and trackpad for finger navigation.

### 3 Our Method

We intend to create a "tabletop" version of the game where the paddle and objects are on a 2D surface in front of the user. This works well for VR, as an existing table can be used as the game surface. Hence, the experience will feel very realistic.

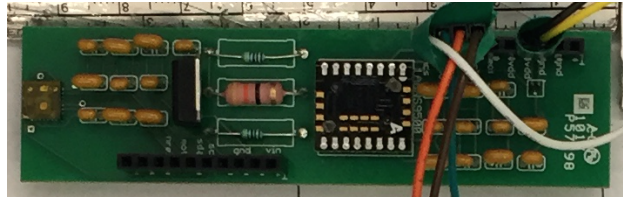


Figure 2: optical mouse chip

Additionally we have a input device that Marcus designed for a medical device prototype. This device consists of an optical mouse chip that can track translation across a surface. We plan to combine it with an IMU to give the translation and orientation of a paddle. The paddle can now move in x, and y directions, as well rotate about an axis. This would add new dimensions to the gaming experience.

### 4 Timeline & Milestones

The end goal of the project is to create a VR Brickbreaker game that functions with our "paddle", with at least one type of item (paddle-length varying item). The project will be broken into 6 steps:

1. Implement data structure (See Appendix A for more information)
2. Generate models of objects in the game (ball, brick, paddle)
3. Stationary scene rendering
4. Game mechanics implementation (with keyboard input)
5. Change input method to our 'paddle'
6. Change output to our VR device (rotation only)

The first two steps are planned to be completed by Sunday (May 28). Step 3 is planned to be completed by Wednesday (May 31). Following this, Step 4 is planned to be completed by Saturday (June 3). Step 5 will be done by the Monday of the week after (June 5). Finally, the last step will be completed by the deadline of the assignment (June 8).

## Appendix A: Data Structures

The following is a list of data structures that we will be using in the project (parenthesis means optional):

- Brick: represents the bricks in the game
  - position: position of the brick
  - size: size of the brick
  - type: the type of the brick
  - hp: remaining hit points of the brick until broken
  - item: item inside the brick
- Paddle: represents the paddle in the game
  - position: position of the paddle
  - rotation: rotation angle of the paddle
  - length: length of the paddle
  - (more depending on the items we choose to implement)
- Ball: represents the ball in the game
  - radius: radius of the ball
  - position: position of the ball
  - velocity: velocity of the ball
  - (spin)
- Item: represents item in the game
  - position: position of the item
  - velocity: velocity of the item
  - hidden: whether the item is shown on screen
  - effect: the effect that the item will have
- BoundaryLine: the boundary of the game
  - p1: one end of the boundary line
  - p2: the other end of the boundary line

(Note:  $(x - p1) \times (p2 - p1) > 0$  (z-component) will indicate 'inside the game')
- GameInfo: the game state

- position (const): the place where the game information is displayed
- score: player's score
- (life)

The "Game" will be composed of:

- Bricks[]
- Paddle
- Ball[]
- Item[]
- BoundaryLine[]
- GameInfo