ZACH BARTH, VALVE SOFTWARE

DEVELOPING ROOM-SCALE VR GAMES
Zach Barth
SteamVR
We’re going to build a VR game…

RIGHT NOW!
Vive + Unity + SteamVR
Along the way, we’re going to explore and solve some common challenges of VR game design.
Let’s get started!
PROBLEM #1

Every VR headset is essentially its own platform, and supporting lots of platforms is difficult.
SOLUTION #1

Use SteamVR; we do the integration work so that you don’t have to!
DEVELOPING ROOM-SCALE VR GAMES

STEAMVR

Game

Unity

SteamVR Plugin

OpenVR APIs

Vive Drivers  Rift Drivers  Future Drivers
PROBLEM #2

Moving the player’s camera makes most people sick, even if they’re “in control” with a gamepad or WASD.
SOLUTION #2

Embrace room-scale VR design principles, even if you’re not just targeting the Vive.
Letting the player move 1:1 in the game world enhances presence and eliminates nausea.
Instantaneous movement, like the “blink” system in *The Gallery*, enables large game environments.
PROBLEM #3

There aren’t enough buttons on these VR controllers for every action in my game!
SOLUTION #3

Map your game’s controls into the player’s virtual environment.
Tracked controllers make it easy to create intuitive interactions with the environment, where less-used controls can be made contextual.
DEVELOPING ROOM-SCALE VR GAMES CONTROLLERS

HEIGHT CONTROL

FARTING CAT
Picking up and holding objects that are held the same way as the controller is convincing and fun.
PROBLEM #4

Having a “HUD” plastered onto your face is distracting, and possibly nauseating.
SOLUTION #4

Map the user interface into the player’s virtual environment as well!
It can be difficult to get users to look at your UI when you want, as you can’t control where they look.
• Think about your environment as a life-sized 3D “visual hierarchy”.

• Attach UI widgets to controllers and use haptics to get the player’s attention.
Thanks!

zachb@valvesoftware.com