

Written description. Write a document that describes how to use your system (including, if appropriate, a clear indication of which targets you use for Vuforia or its equivalent) and why you designed it the way that you did. The grade you receive will depend in part on the rationale behind your choice of techniques, justified in terms of the material covered in class, in the assigned readings, and in any additional material that you have consulted. You should include screenshots in your description, integrated into the document. There is no minimum length; however, your document should fully explain your user interface and design choices. Your description should be submitted as a PDF file.

Angry Birds AR

1. Slingshot
2. Scene Construction
3. Damaging Effect
4. Wayfinding

Slingshot

The slingshot is made to simulate a real slingshot. In the game, the slingshot consists of two parts: two flexible belts and a pad holding the ammo.

Control:

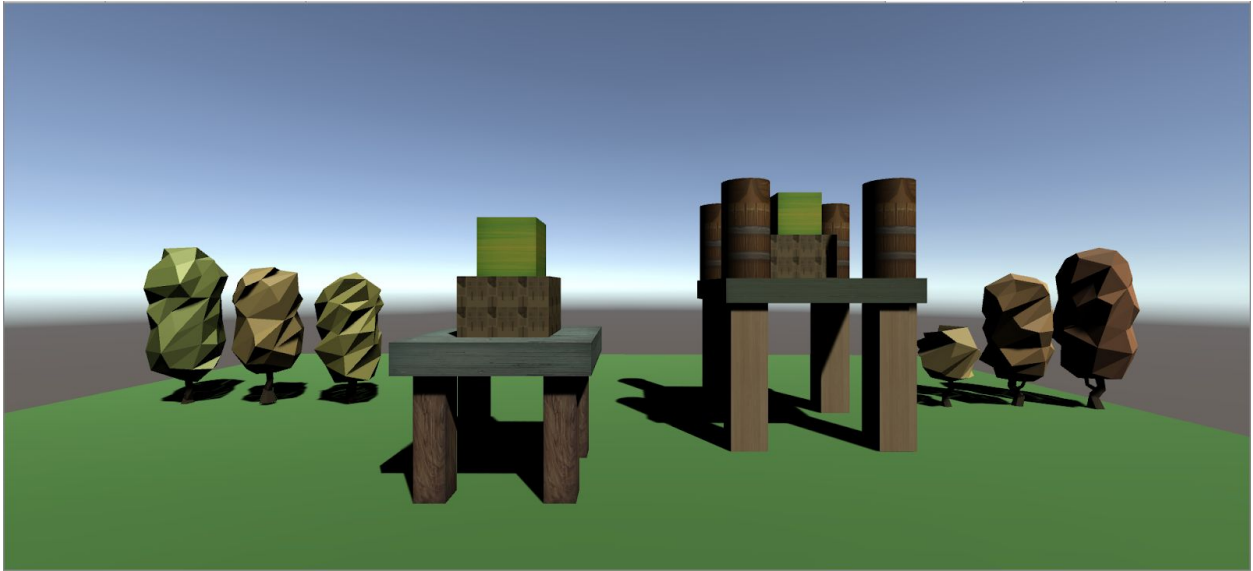
To control the slingshot, users need to take the following steps:

1. Tap on the pad to activate controlling
2. When the users' fingers move on the screen, the pad will follow the finger tip.
Therefore, the pad is dragged towards a certain direction
3. Move the finger away from the screen. The pad will be released from dragging and the ammo will be fired in the direction opposite to the direction the pad is dragged towards.

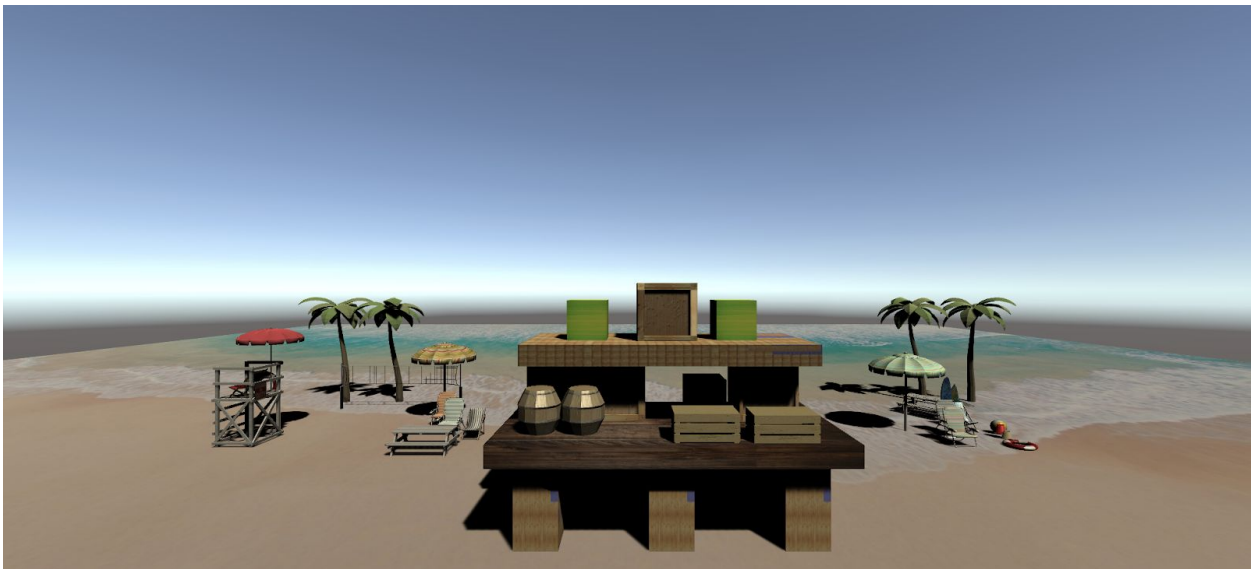
Scene Construction

Two scenes were constructed for demoing purposes. Please see screenshots below.

3D Scene 1:



3D Scene 2:



Several considerations are made regarding scene construction.

- We simulate the classic angry bird game feeling by stacking different objects to create buildings made of blocks. We also add some backgrounds such as forest and beach.
- We choose to use simple objects such as cubes and cylinders, instead of the “fancy” models from asset stores. This is because of the following two reasons.
 - First, when the scenes are displayed in AR mode, the objects tend to slip away because the physics in the AR mode works differently than in the regular 3D mode. Therefore, we calculate the center of gravity for all of the simple objects, making sure that the building blocks structure is stabilized.
 - Second, through trial and error we discover that simple objects have more realistic physical reactions when hit by the slingshot or when colliding with each other. On the contrary, the fancy models look better but feel clumsy in terms of physics when we play the game.
- Please note that the green cubes are the enemies. We choose to use cubes as enemies for the same reason as above.
- We then integrate the two scenes with Vuforia Engine to create the finalized AR scenes. We put both scene “together” in the sense that if one scene is out of camera view, we could use wayfinding to find another. This would be explained in more detail in the wayfinding section.

Damage Effect

To understand the damage effect, we referred to the original game Angrybird2. After playing the game to level 60 and unlocking all birds, we have fully experienced the game design from all perspectives.

Despite the rather annoying In-Game Purchasing system, and different tribe wars, daily challenges, etc. that encourages users to do In-Game Purchasing, the game provides 8 different kinds of birds with different effect, including wave attack to destroy building structure, speeding to give birds higher impact, boombing, etc. Also other special effect tools are provided, including duck rains, a blow of wind, UFO x-ray beam, etc. Some environment decorations also provide special consequences. From teleporting to shooting swallowed items, they help players to win.

Those units and effects all generate different influences onto the building itself and also the enemies, in the way that provides users different approaches to enjoy the game and conquer levels. But from the enemy side, effects are rather simple. We summarized the damages into three categories: Falling, Explosion, and Collision.

Falling is as simple as you could see from the word. When any unit drops out of the boundary (for example, into the endless valley), it disappears and gets destroyed immediately. It is an instant kill magic that wipes out all HP, so knocking enemies to fall from cliffs becomes a good way to defeat bosses with enormous HP.

Explosion has a larger effect area, that would damage things inside the circle with a shockwave. We decided not to implement this.

Collision is the most common way to destroy bricks and enemies. Although it seems that enemies are instantly killed when they are hit by anything, it is not true that different objects with different forces would have different impacts. Birds touching of course is 100% lethal, however, most enemies could survive from gentle fall of a wooden brick or any collision with balloons.

Based on the discoveries, we give enemies and brick prefabs different HP and damage. Different collisions would be reflected on the HP reducing the other collision object's damage. When HP falls to 0 or under 0, the unit is destroyed with a particle effect (maybe not used in final version).

Wayfinding

In the game we have two image targets, each with one scene built on it.

At the start of the game, the camera in our device would capture both image targets. It then “remembers” the spatial information for both scenes including location and orientation. In the meantime, it creates a proper view point for each scene which provides reference location and orientation for starting shots in a specific scene. What’s more, taking an exocentric reference form, a cognitive map that divides scenes into distinct parts and records the sequential relationship of these scenes will be built.

At this point, the user is free to move the camera around. If either scene is out of the camera view, a list of arrows as an artificial cue would appear on screen, following a proper trail that guides the player towards the proper view point of that following scene. According to the relative spatial information, it distributes these arrows properly, gradually changing location and orientation of each arrow. In this way the user would always know where and which orientation to play both games.