



Okay everyone, here's the complete design document. This covers everything we decided this yesterday, including who is responsible for what and all of the finalized details of the game. I'm also including a couple of things that occurred to me on the way home from class last night. At this point we have everything nailed down, so all we need to do is fulfill our ends of the project and we should be done on time. So here goes...

Title: "Spitball"

4 players

Controller: Wiimote "Nunchuck" attachment

Basic Description: A two on two dodgeball style video game. The goal is to hit the other team's players with the balls to score points. Rounds are three minutes long. At the end of the round the team with the highest score wins.

Controls:

Control stick = move player/aim throw

Catch = Z button

Dodge = C button

Throw = release Z button (or throw gesture if it is working)

Action descriptions:



the

nature of the bouncing.

Moving: Players control their position on-screen via the control stick. In class Scott had the arduino detecting the analog signal, so I presume this means movement velocity will be pressure sensitive based off of how far the stick is pushed. This action will be visually represented by the idle animation, which is essentially the blobs bouncing slightly in place. So, whether the character is in motion or not the animation being played will be the same, which should work well given



ni-

cate more clearly to the player if it is one frame simply held in the mouth open pose. This way the player will understand precisely how long the "catching" window is. The player can move while catching.

Catch: Players press the Z button to catch a ball. The catching action only lasts X amount of time (probably about a half second), and it can not be repeated until a specified duration passes (probably about 2 seconds). This should make the act of catching a skill, and prevent players from simply holding down the catch button or tapping it repeatedly to become invulnerable. This action be visually represented by the catch frame. This can be animated, but it might communicate more clearly to the player if it is one frame simply held in the mouth



Charging a throw: Players continue to hold the Z button to charge up a throw. While a player is charging a throw the player can not move. Charging a throw increases the velocity in which it will be propelled in order to make it more difficult to dodge. If a ball is charged too long it will kill the player charging it, so timing will be crucial.

The skill then becomes knowing how long to charge the ball before it blows up in a players mouth, and in the event that a ball is incoming while the player is charging a ball then the player must release the ball in order to move out of the way, thus creating a small amount of strategy to the play mechanic. The charge builds from the moment it is caught until X number of seconds (probably about 3 seconds) while being held. Once it reaches maximum charge it will take X amount of time (probably about 1 second) before it blows up in the players mouth. This whole sequence will be visually represented by the players cheeks beginning to puff up and the eyes widening. The first 3 seconds will be the puff-up stage, with the last 1 second being something akin to a look of desperation before the player explodes.

Note: A player can perform no other action while charging a ball. Moving and dodging are disabled. The player is temporarily committed to offense.

Throwing: Players simply release the Z button or gesture with the Nunchuck to launch a ball. Aiming is handled by tilting the control stick in the desired direction

Dodging: By pressing the C button a player is able to dodge briefly. Dodging allows balls to safely pass through a player. This move will only last X amount of time (probably 1 second) and will not be able to be done again until X amount of time after the last dodge was initiated (probably about 4 seconds). We didn't determine how this would be visually represented, but I thought an elegant solution would be to have the player disappear and only the shadow on the ground would remain. This visual representation would clearly communicate to the uninitiated player exactly what is happening. The dodge is the only counter to the uncatchable throw.

Uncatchable throw: Successful catches by a player build up to the ability to launch an uncatchable throw. An uncatchable throw is a throw that can not be caught, and will continue to be lethal for X number of bounces off of the wall (probably 2 or 3). An uncatchable throw builds up in three stages. After one successful catch a player begins to be surrounded by slight flames. On the second successful catch the flames grow higher. On the third successful catch the flames reach the maximum intensity and the player is now able to launch an uncatchable throw.

Note: An uncatchable throw and a charged up throw are not mutually exclusive, meaning that a player can release an uncatchable throw at any velocity. A player's uncatchable status is reset once they launch the throw or if they are hit.



Death: Death comes in two flavors. The first is suicide, which was already covered under charging a throw. The second is homicide, which is being hit by a player's ball. Once a player is struck by a ball that player is disabled for X amount of time (probably 3 seconds) and then comes back to life where they stand. It would probably be a good idea to have a grace period of invulnerability (about 1 second) after the player comes back to life to prevent re-spawn deaths. It is possible to die by friendly fire. If this happens, a point will simply be awarded to the opposing team.

Game flow:

The game starts with both players on each team evenly placed on the playing field. There is one ball sitting in the dead center of the field, and one ball on each team's side placed evenly between the players (see illustration for starting position). The players then race for the balls, with the middle ball being a point of contest. It would be nice if the balls were not weapons until after they were thrown, so the players don't accidentally hurt themselves. Once chaos ensues it continues until the timer runs out.

Duties:

Scott - Programming/controller implementation

Paul - Animation

June - Animation

Galen - Arduino case fabrication/gofer/"Team Beagle Claw" logo design

Jeff - Sound effects and music/this damn document

Deadlines:

This thing is due on Thursday the 12th. Everyone should be done with their end of the bargain by Tuesday the 10th. This means that Scott will have a working version of the game minus the visuals and sound effects. Paul and June will be finished with the animation. Galen will have fabricated the arduino casing. And I will be done with all of the sound and music. This way on Tuesday we can supply Scott with all of the assets and do some QA and begin assembling the final version of the game, leaving one day before the due date for troubleshooting. In addition, I may need select animations early so I can create specific sound effects.



Detailed breakdowns:

Ball size is 40 x 40

Player size collision ellipse is 75 x 75

Our total game window is 800 x 600

Our play field area is 760 x 480

There is a 20 pixel boarder around the left, right and bottom of the field, with a 100 pixel high top area for score and time limit.

Character sprites should be animated within a 100 x 100 frame, and rendered as .PNG to preserve the transparency.

The characters themselves should be 75 x 75 pixels in dimension, and the animation should always center around the 75 x 75 circle in the center of the 100 x 100 frame to

ensure that all action animation will relate with no skipping and no collision detection discrepancy. Animate over the template I created to simplify the process.

Asset table

Event	Duration	Animation	Controls	Sound
Player movement	Variable	Idle (bouncing in place)	Control stick	none
Catch	.5 seconds (can't be activated again for 2 seconds after action is completed)	Static "catch" frame	Z button	none
Charge-up	4 seconds total (3 seconds to max. velocity. 1 additional second until suicide)	first 3 seconds = cheek buldge/eyes widen. last 1 second = panicked look/possibly change player color to red	hold Z button	yes
Throw	action instant. Duration for animation 1 second	Spitting	Release Z or gesture	yes
Suicide	action instantaneous. Duration for animation 1 second	Exploding head	Hold Z too long	yes

Event	Duration	Animation	Controls	Sound
Homicide	player stays dead for 3 seconds	dead head (x'd eyes and tongue sticking out) will work well as static frame (dead things don't move).	n/a	yes
Dodge	1 second (can't be activated again for 3 seconds after the action is completed)	invisible (just a shadow)	C button	yes
Charge-up flames	variable	three iterations of flames. Goes from just visible to ON FIRE!	n/a	maybe. Depends on how annoying potentially having four players "on fire" would be.
Background music	3 minutes (round length)	n/a	n/a	yes
Uncatchable throw	2 bounces then returns to normal	no animation per say, but while uncatchable the ball should flash instead of being solid white.	n/a	yes

Model Sheet and Color Palette:

*note to animators: I presume you worked out how to split the workload, so we'll leave it at that. Don't forget the shadows. Also, we talked about having a striped and solid team so people wouldn't confuse what team they're on. I think the fact that the players are constrained horizontally and that they are facing different directions should be sufficient, so I say just leave them all solid. This should cut the animation workload by almost half and minimize visual confusion. Don't forget to mirror two of the characters and color them all differently. Go team Beagle Claw!

Whew! I think I covered everything. If I missed something or you have questions please contact me: jef-farmstead@hotmail.com

