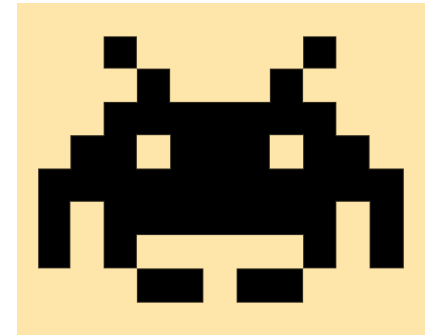




CSCI 1106

Lecture 8



Projectiles



Today's Topics

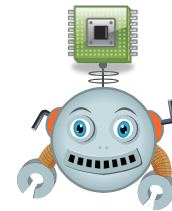
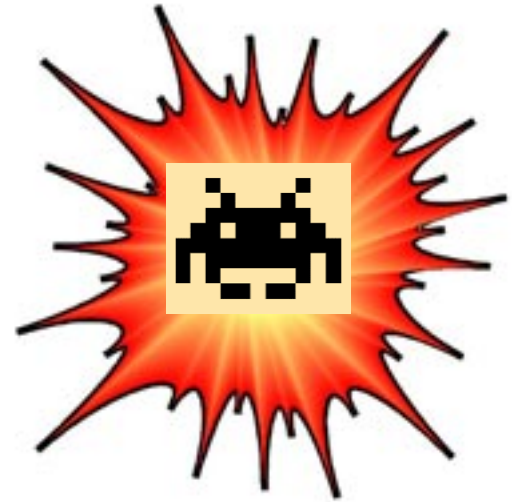
- Motivation
- The projectile life-cycle
 - Design
 - Instantiation
 - Creation
 - Motion
 - Elimination

Projectiles

- One of the most common interaction mechanisms in games are projectiles
 - Bullets, lasers, asteroids, ships, boomerangs, etc.
- Both the players (good guys) and the game opponents (bad guys) may use projectiles that are launched at the other side
- How do we implement projectiles?

A Projectile ...

- Appears on the stage when the player/opponent does something
- Appears initially at the player's/opponent's location
- Moves away from the player/opponent in a set direction
- Disappears when it hits something
- Causes opponent/player to react in some way



The Projectile Life-Cycle

- Design (during game development)
- Initiation
- Creation (Cloning)
- Motion and Collision
- Elimination

Projectile Design

- Design projectiles to support the game's unifying theme
- Create projectile sprites
 - Draw projectile objects
 - Import projectile costumes, if drawn using tools other than Scratch
- Mark sprites as hidden
- Clone the projectiles when needed
 - Move clone to initial starting location
 - Make clone visible

Projectile Initiation

- Idea: A projectile is initiated as a result of an event
- Player events:
 - Mouse click or key press
 - Collision with another object
- Game (opponent) events:
 - In a random or regular time intervals
 - Collision of objects within the game
 - Start of game or level (e.g., the ball in BrickBreaker)
- Idea:
 - Broadcast `NEW_PROJECTILE` when a projectile is needed
 - The projectile sprite will receive the event and create the projectile

Frequency of Projectiles

Player Options

- Unlimited load and speed
 - As fast as possible
- Limited load
 - As fast as possible for a fixed number of projectiles
 - Require a recharge period to continue firing
- Limited speed
 - Allow player to fire one projectile per time period
 - Many players find this annoying
- Limited load and speed

Opponent (Game) Options

- Regular frequency
 - Create new projectiles on a regular basis
 - Not too fast or too slow
- Random frequency
 - Randomly decide in each time interval
 - Total number of projectiles per unit time should be limited
- Frequency increases as levels increase

Projectile Creation

- Idea: Projectiles are created by an event listener
- To create a projectile,
 - Projectile sprite
 - Receives NEW_PROJECTILE
 - If sprite is not a clone, a projectile can be created
 - Set position
 - Set speed
 - Set direction
 - Clone self
 - Projectile clone
 - Marks itself as a clone
 - Set itself as visible



Projectile Position and Velocity

Player's Projectiles

- Position
 - In front of the player's avatar
- Direction
 - Same as the player's avatar
- Speed
 - Depends on game itself
 - Cannon ball vs laser beam

Opponent's (Game) Projectiles

- Position
 - Front of the opponent's avatar or
 - Random position from edge of stage
- Direction
 - Away from the opponent
 - Towards the player's avatar
 - Parallel to the stage
- Speed
 - Sufficient to give the player a challenge

Projectile Movement

- Idea: Projectiles move just like all other objects
- On each FRAME event
 - If projectile is a clone
 - move projectile
 - check for collisions with other sprites
 - check for collisions with stage edge
- Projectile must be removed if there is a collision
- Note: The original projectile sprite should never move and always remain hidden

Projectile Collisions

- Purpose of projectiles is to collide!
- On each FRAME event
 - Check if projectile has collided with
 - Avatar (player or enemy)
 - Other game objects (terrain, walls, bricks, etc)
 - If collision occurs
 - Broadcast COLLISION event to the sprite
 - Delete projectile
- On COLLISION event, the sprites receiving the event
 - Check if they have collided with a projectile, if so
 - Create some special effects (optional)
 - Adjust state of hit object (health, etc)

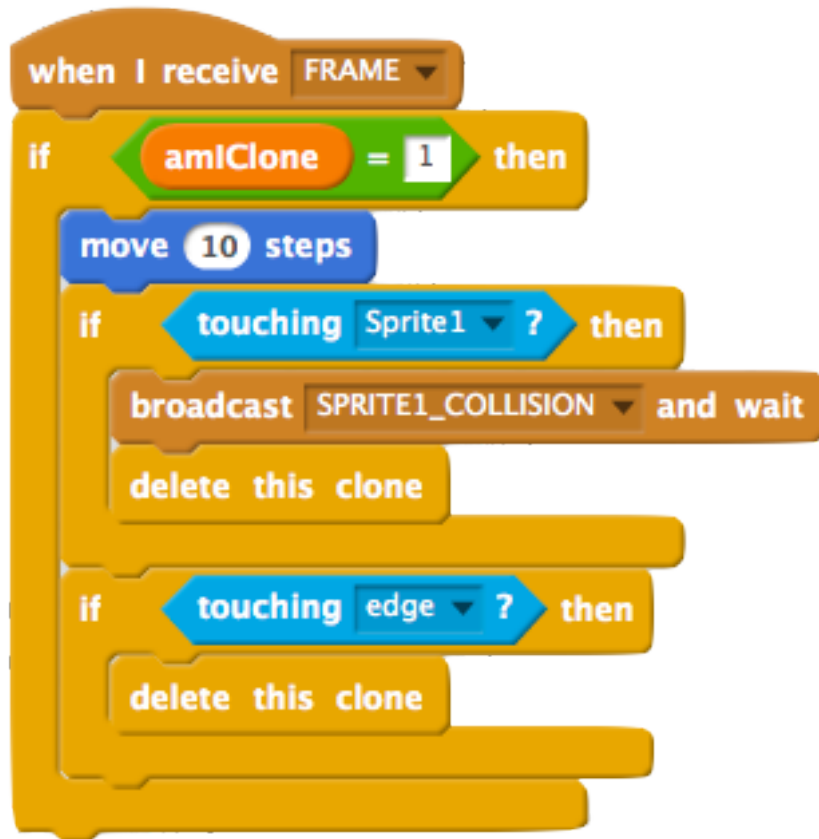
Projectiles Moving Off-Stage

- Projectiles moving off the stage are removed
- Idea: On each FRAME event
 - Check if projectile has moved off-stage
 - If projectile is off-stage, delete projectile

Projectile Elimination

- Once a projectile moves off-stage or has collided, remove it!
- Your game will slow down if you do not!
 - Why?

Example of Projectile FRAME Script



- On each FRAME event
- If Projectile is a clone
 - move
 - If collision with Sprite1
 - Inform all Sprite1s
 - Delete projectile
 - If at edge of stage
 - Delete projectile

Some hints

- <http://wiki.scratch.mit.edu/>
- Search :How to (do something) in scratch
- Documentation:
 - Description of Concept :
 - Game's genre, mechanics, and story.
 - Description of Sprites:
 - Name, purpose, behaviour, variables, messages it receives, and interaction with other sprites.
 - Description of the Stage:
 - The behaviours, messages, and actions performed by the stage
 - High-level Description :
 - High-level interaction between the sprites and stage, all the variables and how they are used.
 - Description of Important Scripts :
 - Describe the scripts associated with the sprites and the stage