Comp 388/488 - Introduction to Game Design and Development

Spring Semester 2017 - Week 12

Dr Nick Hayward

Contents

- Game Designers
- Games and dramatic elements
- Pygame - Game Example 1
  - *fun game extras*
    - repetitive firing sequence
    - explosions for sprite objects
- References

Game designers

Designer example - Will Wright

- Wright is a veteran American game designer
  - *best known for his work on The Sims*
- *The Sims* was originally released in 2000
  - *led to countless versions, spin-offs &c.*
  - *driven a genre more interested in participation than a definitive win*
- as a co-founder of *Maxis*, and then later part of EA
  - *Wright also developed the game Spore*
- he's often referred to as a designer of *software toys* instead of traditional games
  - *a consideration of the non-traditional structure employed for many of his games*
- he's also been a passionate developer of, and advocate for, emergent and adaptive systems
- Wright has continued to develop this concept for many of his games
  - *his legacy is evident in games such as Spore, The Sims 3 and The Sims 4*
- Wright has tried to use these systems with their simple rules and definitions
  - *to provide the possibility for the development of complex, detailed outcomes*

Resources

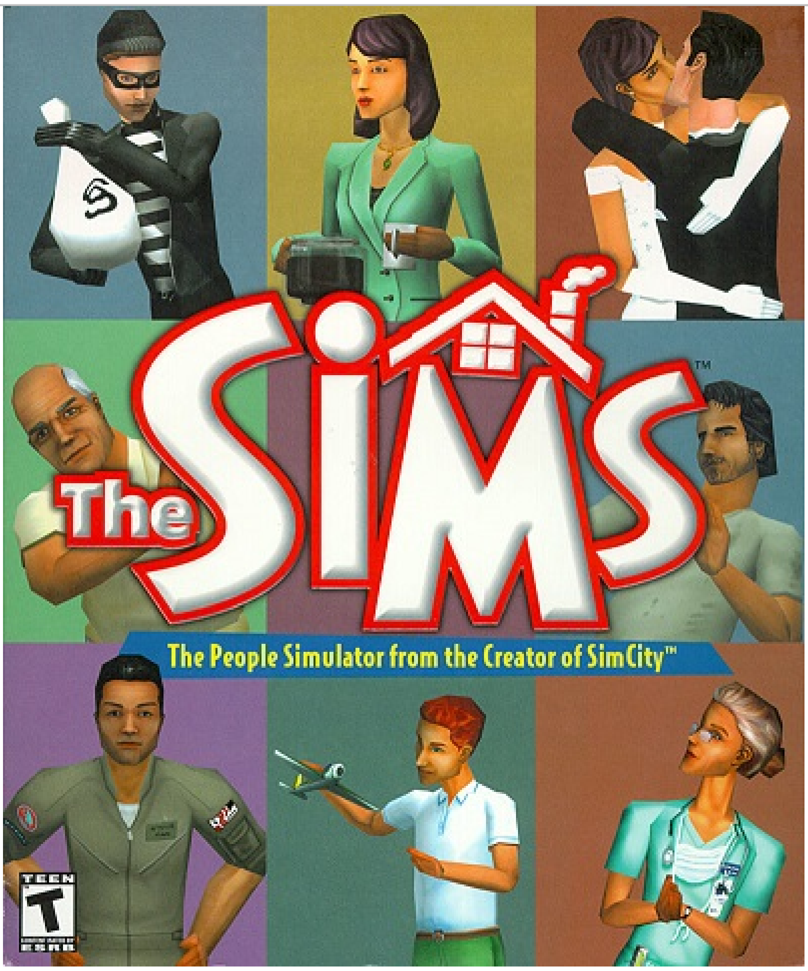
- Maxis
- The Sims
- Spore
- Will Wright

Image - Will Wright

Will Wright

Image - Will Wright

The Sims and Spore



The Sims

Games and dramatic elements

intro

- may consider *dramatic elements* as we continue to design and develop our games
- already considered many underlying elements and concepts that create a game we recognise
- also need to consider those elements that create...
  - *a sense of emotion,*
  - *engagement*
  - *and challenge for our players*
- aspects of our game that encourage an emotional connection
  - *simple desire to invest time and effort in gameplay*
- **dramatic elements** help create a sense of context to a player's experience with our game
- **dramatic elements** provide a backdrop/overlay for our game
  - *combines many disparate formal elements of our game logic and development*
  - *creates a conceptually meaningful experience for the player*
- may start with universal concepts for such dramatic elements
  - *including challenge and play*
- then branch out into more complicated considerations of elements, e.g.
  - *characters, premise, story...*
  - *used by most games we design, develop, and play*
- used to form core for explaining many of more abstract elements of a game's formal system
- help create a deeper sense of connection between the game and its player

Games and dramatic elements

gaming challenge

- *challenge* and an associated sense of accomplishment
  - *fundamental definition of gaming for many players*
  - *perception of worthwhile gaming experience*
- challenge alone is often no different from work, daily issues...
- designers need to find a happy balance to challenge and reward
- need to consider tasks that are satisfying to complete and provide a balance between work and fun
- designers are inherently limited by the abilities and skills of an individual player
- challenge may also become an individual perception and characteristic of a player
  - *consider difference between age groups, skill levels, experience...*
- challenge may also be considered *dynamic*
  - *a player's ability will adapt and improve*
  - *hopefully as they learn and progress through a game*
- a challenging early task may become considerably easier
  - *i.e. as a player progresses to subsequent levels and areas within a game*
- as a player learns these new skills
  - *enjoys opportunity to test and demonstrate these skills elsewhere in the game*
- incremental modifications and updates to earlier, completed challenges
  - *provides a quick and easy option for the player to balance challenge with reward*
- designers and developers need to consider challenge carefully
  - *challenge that is not necessarily defined by individual experience*

Games and dramatic elements

a sense of flow

- carefully consider how to design our games to effectively consider *challenge*
  - *as defined and restricted by individual experience, &c.*
- each experience can, therefore, take advantage of an appropriate level of challenge
- a well-known example of this was developed by the psychologist **Mihaly Csikszentmihalyi**
- he wanted to identify concepts and elements that might help define enjoyment for a given task
  - *he studied experiences and similarities of various tasks for different people*
  - *trying to discern similarities of experience for these tasks, players...*
- his research noted a distinct lack of traditionally perceived bias
  - *for what we consider fun and meaningful tasks*
  - *lack of bias in results for age, social standing, gender...*
- people simply described their perception of enjoyable activities in a similar manner
- regardless of the activity itself
  - *often included disparate pursuits such as music, painting, and playing games...*
  - *the words and concepts people used to articulate this sense of fun was largely the same*
- for each of these tasks
  - *certain conditions became recurrent and popular for describing pleasurable activities*
  - *each user and player was entering into a state of **flow***
  - *allowed for this heightened sense of achievement, and associated fun*

Games and dramatic elements

perceptions of flow

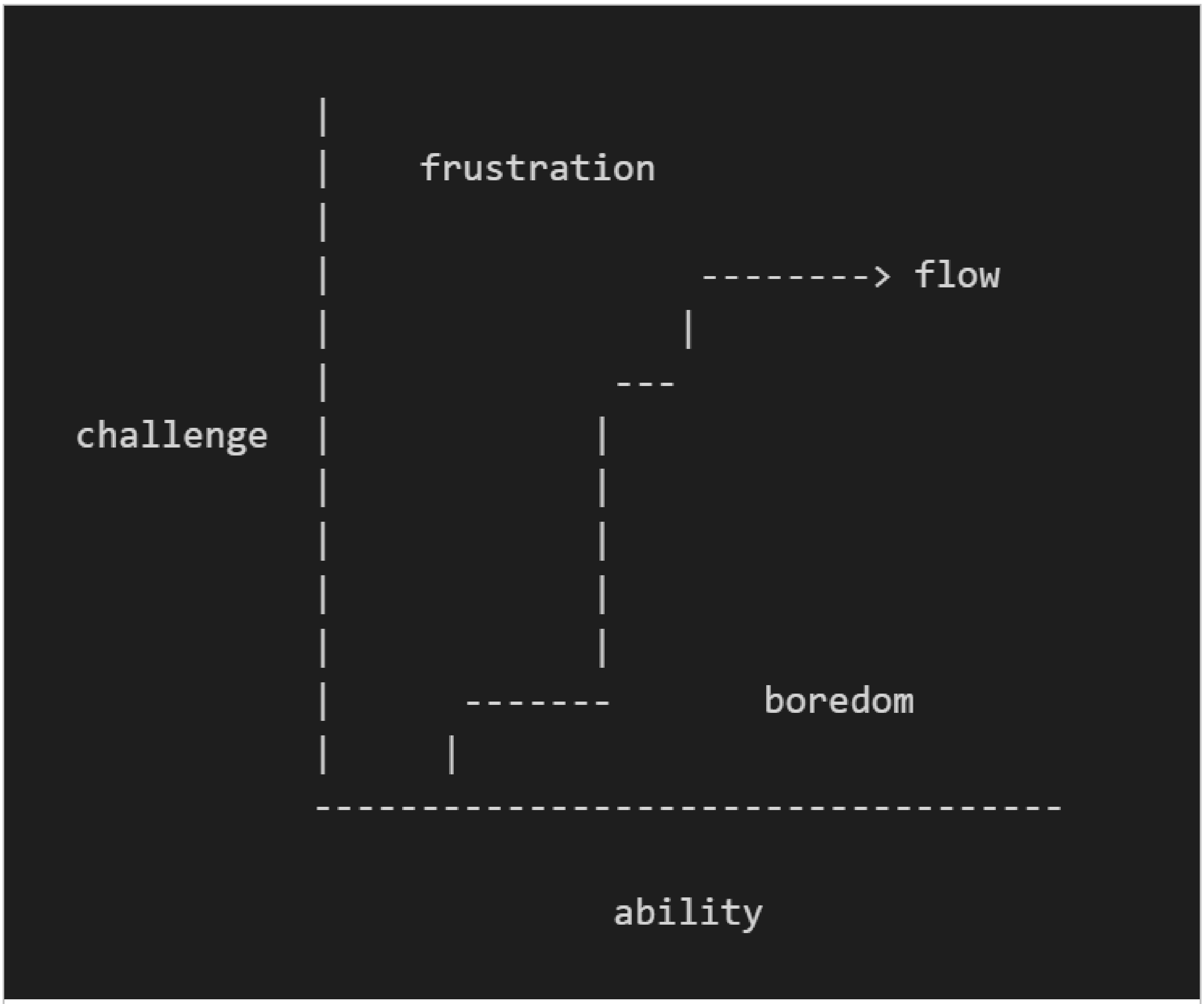
Flow by **Mihaly Csikszentmihalyi**

- player's creativity, ability, and general awareness are high
  - *performance of activity occurs naturally and unconsciously*
- player experiences deep concentration and immersion in their current activity
  - *player is effectively both alert and relatively relaxed*
- living in the moment
  - *a sensation of being so engrossed in an activity a player is unaware of the passage of time*
- balancing interest and challenge
- player is confident and exhibits a sense of control over their current situation
- player is working progressively towards achieving a specific goal, e.g.
  - *getting to the next level in a game*
  - *completing a mini-challenge*
  - *or mastering a particular mechanic for their current character*
  - *Luigi's Mansion and the vacuum cleaner...*

TED 2004 - Flow, the secret to happiness

Image - Games and dramatic elements

a state of flow

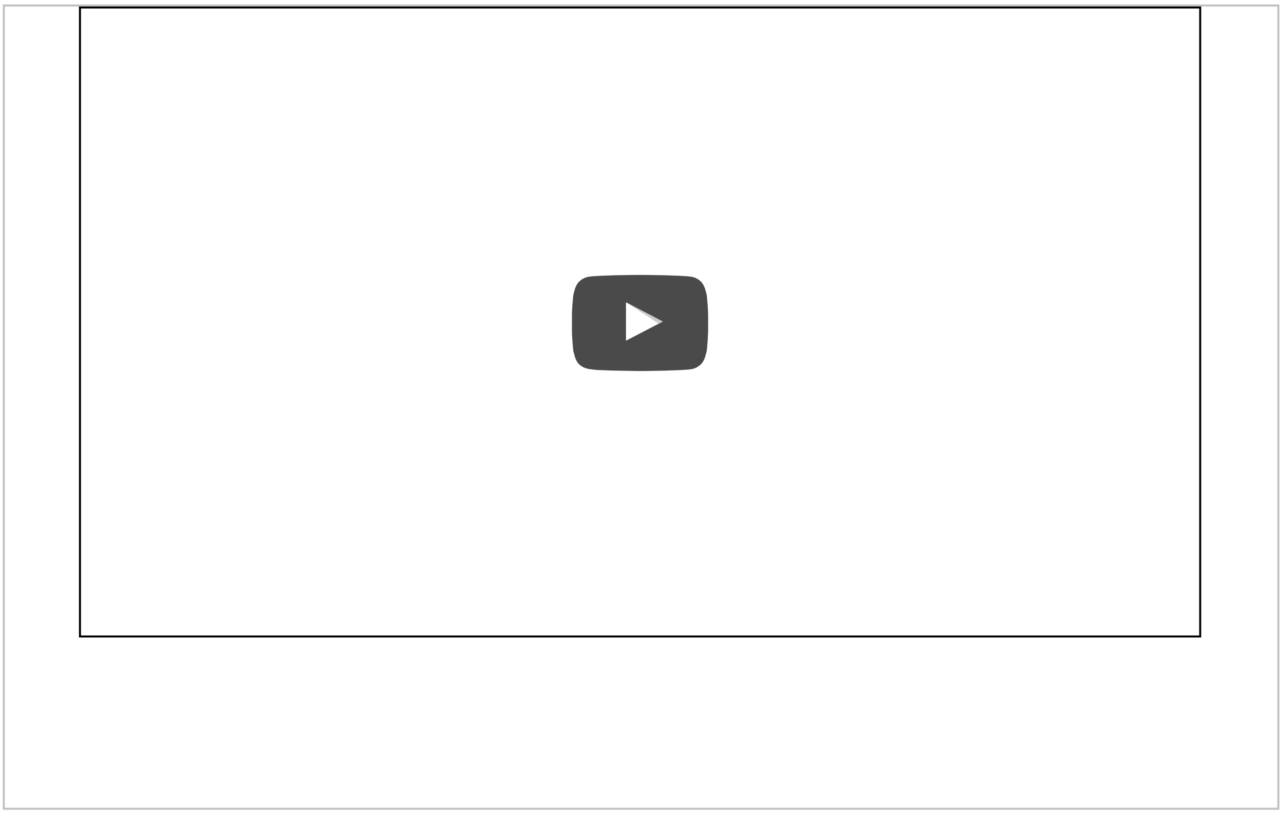


```
           |
           |    frustration
           |
           |                  --------> flow
           |                 |
           |              ---
challenge  |             |
           |             |
           |             |
           |             |
           |             |
           |      -------        boredom
           |     |
           -----------------------------------

                         ability
```

A state of flow

Video - Colin McRae Rally



Source - Colin McRae Rally, YouTube

Python and Pygame - Game Example I

fun game extras - repetitive firing sequence - intro

- add a repetitive firing sequence for the player's sprite object
- in our current game logic
  - *as a player presses down on the space bar a laser beam will be fired from the top of the player's ship*
  - *one press is equal to one firing sequence...*
- to add a repetitive firing sequence
  - *need to still check that the spacebar has been pressed down*
  - *but now continue to fire a laser beam until the key is released*
- in our `Player` class we can add some new variables, e.g.
  - *specify the delay in milliseconds between each firing of the laser beam*
  - *check the time, the number of ticks, since the last beam was fired*
  - *e.g. update `Player` class as follows,*

```
...
# firing delay between laser beams
self.firing_delay = 200
# time in ms since last fired
self.last_fired = pygame.time.get_ticks()
```

Python and Pygame - Game Example I

fun game extras - repetitive firing sequence - fire - part I

- add a listener for the space bar event to the `update()` method in the `Player` class

```
# check space bar for firing projectile
if key_state[pygame.K_SPACE]:
    # fire laser beam
    self.fire()
```

- update our `fire()` method to reflect this repetitive firring sequence, e.g.

```
...
# get current time
time_now = pygame.time.get_ticks()
if time_now - self.last_fired > self.firing_delay:
    self.last_fired = time_now
    ...
```

Python and Pygame - Game Example 1

fun game extras - repetitive firing sequence - fire - part 2

- our `fire()` method has now been updated as follows,

```
# fire projectile from top of player sprite object
def fire(self):
    # get current time
    time_now = pygame.time.get_ticks()
    if time_now - self.last_fired > self.firing_delay:
        self.last_fired = time_now
        # set position of projectile relative to player's object rect for centerx and top
        projectile = Projectile(self.rect.centerx, self.rect.top)
        # add projectile to game sprites group
        game_sprites.add(projectile)
        # add each projectile to sprite group for all projectiles
        projectiles.add(projectile)
        # play laser beam sound effect
        laser_effect.play()
```

- remove listener for a space bar event in the events section of the game loop

resources

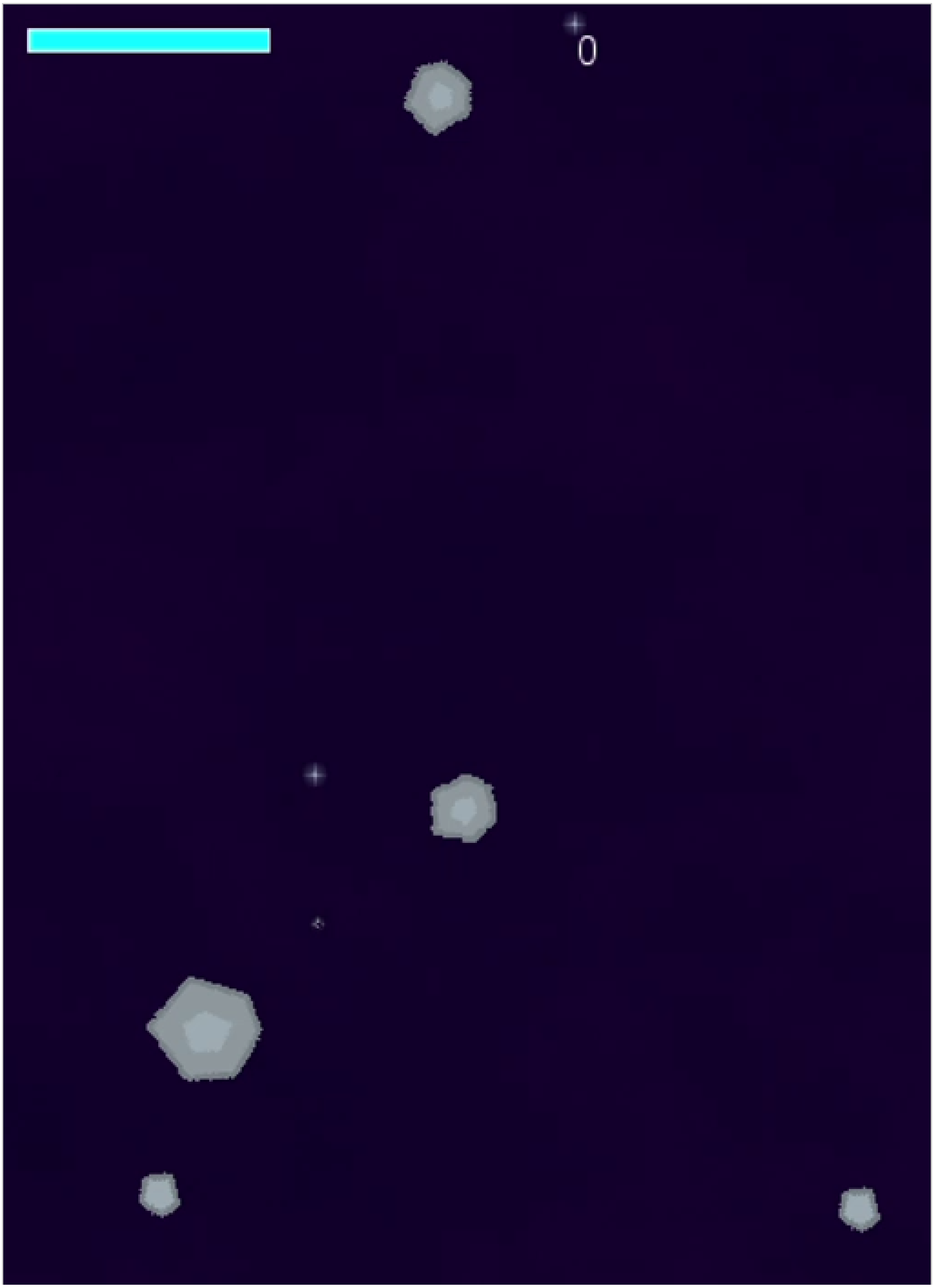
- notes = extras-part1-firing.pdf
- code = repetitivefiring.py

game example

- shooter1.1.py
- add repetitive firing sequence for player's laser beam
    - *move keypress check for space bar to player class*
    - *fire laser beam whilst space pressed down*
    - *set interval in ms for firing sequence*
    - *check time between now and last firing*

Video - Shooter 1.1

add repetitive firing sequence...



0:00 / 0:37

Games and dramatic elements

consider skills

- start introducing challenges and associated activities into our games that require definable skills
- may be a mixture of assumed or learnt skills, applicable to the current game
- for *flow*, **Csikszentmihalyi** describes it relative to activities that are considered,

> *goal-directed and bounded by rules...*

- Csikszentmihalyi, M. *Flow: The Psychology of Optimal Experience*. Harper & Row. New York. 1990. P.49.
- such activities not customarily achieved or completed without proper requisite skills
- skills may include various examples, including
  - *standard motor skills for controls and interaction*
  - *problem solving*
  - *social interaction with other players...*
- challenges, and the development of skills, need not necessarily be limited
  - *e.g. by simple clicking of buttons, and the resultant moving of pixels...*
- a common trick to manipulate such skills is the introduction of doubt or variance
- imagine a challenge or task where the ending is not known or guaranteed
  - *e.g. a player's character walking along a ledge*
  - *may be wet underfoot*
  - *perception of wind blowing from any direction*
  - *random mob objects falling*
  - *varying time due to health status...*

- underlying motor skills, for example, are the same for the player's character
  - *but the end result has now been challenged and thrown into doubt*

Games and dramatic elements

a story and premise

- a **premise** becomes a wrapper or container for our game
  - *we may use to create a sense of context for such challenges, skills, and fun*
  - *a sense of story...*
- each game we design and develop will include such a *premise*
  - *might be a single concept or a detailed dramatic backdrop*
- our games will often leverage a few well-known dramatic elements
  - *help create a player's connection and interest in a game's formal elements*
- use *premise* to help identify the game's formal elements within a setting or a metaphor
- without a sense of context and setting
  - *we may abstract mechanics, gameplay, and skills too far*
  - *reducing sense of fun for our player*
- consider difference between an outline of initial game logic and the wrapper a *premise* provides

Games and development

quick exercise

Consider the following metaphors,

> *The skies of his future began to darken*

> *Her voice is music to his ears*

> *The ballerina was a swan, gliding across the stage*

> *A heart of stone*

Choose two of the above metaphors, and consider the following:

- how might your chosen metaphors shape the premise and story of a game?
- how might the premise of this game influence mechanics and skills for characters?
- how may you use such skills to create challenges in the game?
- how do your chosen metaphors, and the inferred premise, wrap this game's formal elements?

Games and dramatic elements

examples of premise in games

Space Invaders

- classic example of a shoot-em up game
- simple premise for this game
  - *easy to extrapolate and apply to game's mechanics, gameplay, and challenge*
- game is set on a planet currently being attacked by advancing aliens
- game's protagonist is responsible for fighting off these aliens and saving the planet
- game will start as the aliens start advancing down the screen
  - *and the player starts firing their weapon...*

Diablo

- first released in 1996 by Blizzard Entertainment
  - *Diablo III available for latest consoles &c.*
- more detailed premise for this game
- allows the player to act out the role of a wandering warrior
- located in a town called *Tristram*
- the town has been attacked and ravaged by Diablo
- player is acting in reponse to a call of help from the people of this town
  - *who need the player to defeat Diablo and his army of the undead*
- army is located in the dungeon beneath the town's church
- game will start as the player accepts the town's proposal

- the game leads to a final confrontation with Diablo in Hell

- Diablo III - console

Games and dramatic elements

characters

- as we define our game's story, and the premise for its structure, gameplay, &c.
    - *a core consideration is the nature of our game's characters*
- characters form the route, conduit, or agent for a player
    - *a player may experience the game through these characters*
- this identification becomes an important consideration for our design
    - *helps promote a sense of immersion and internalisation*
- a player will often start to empathise with a character
    - *their role in the game*
    - *their inherent need to often resolve the game's story*
- from a pyschological standpoint, a dramatic character is often perceived
    - *an extension of fears and desires often projected by a player &c.*
- such characters will often embody certain characteristics - good and bad
    - *may be associated with a greater goal or need of the player*
- a character may also be influenced by a game's type or genre
    - *often why we encounter stereotypes &c. in certain game genres, series...*
- may help lessen the need to deconstruct the game's story
    - *effectively making it easier to accept the premise of the game...*
- the *protagonist*
    - *a game's main character*
    - *often helps drive a sense of conflict and challenge*
    - *by engaging with a defined problem or series of related problems*
    - *this sense of conflict will help drive the story*
- the *antagonist*
    - *a game's counterpoint to the main character*
    - *may be another character or a feature of the game's logic*
    - *the antagonist may be used to push back against our game's protagonist*

- without this conflict and contrast
  - *a game will often lack the necessary dramatic counterpoint*
  - *any semblance of depth to the gameplay will often be lacking...*

Games and development

quick exercise

Consider the following questions relative to perceived characters in the game you outlined for the previous exercise.

- for the game's protagonist, what do they want?
- what does the antagonist need?
- what are the hopes of the player for the protagonist?
- what are the fears of the player for the protagonist?

Python and Pygame - Game Example I

fun game extras - load explosion images

- need to be able to define and load our images for the explosions
  - *use a list for these images*
  - *then cycle through these explosions as required...*
- our first exanple will use a list to simply load these explosion images
  - *initially use a `for` loop to iterate over this directory and load our images, e.g.*

```python
# explosions
explosion_imgs = []

# iterate over explosion images in directory
for i in range(9):
    file = 'explosion{}.png'.format(i)
    expl_img = pygame.image.load(os.path.join(img_dir, file)).convert()
    expl_img.set_colorkey(BLACK)
    explosion_imgs.append(expl_img)
```

- use built-in function, `format()`, to specify abstracted value for iterator index
  - *in this example abstracted for the required filename*
- create our image for the Pygame window
  - *set colour key to black to create our transparency for the containing shape's background*
- then append these images to our list for explosions

Python and Pygame - Game Example 1

fun game extras - create explosion sprite object - part 1

- create a new class to help us represent and organise our sprite object for *explosions*
- add a new class for this object
  - *then start by initialising this sprite, e.g.*

```
# create a generic explosion sprite - use for asteroids, player explosions &c.
class Explosion(pygame.sprite.Sprite):
    # initialise sprite
    def __init__(self, center):
        pygame.sprite.Sprite.__init__(self)
        ...
```

- after initialising this new sprite object
  - *set starting image for our explosions*
  - *set to first index position of our list for explosion images*
- need to add the rectangle for this image
  - *set its centre to the specified value of the passed parameter*
- also set initial frame for our animation
  - *we can set it to a starting default of 0*

Python and Pygame - Game Example 1

fun game extras - create explosion sprite object - part 2

- animation needs to be steady and constant
  - *may create a steady framerate for the animation itself*
  - *now check the time in ticks for the last update*
- then set a default framerate for this animation
  - *modify framerate of animation to suit game requirements*

```python
# create a generic explosion sprite - use for asteroids, player explosions &c.
class Explosion(pygame.sprite.Sprite):
    # initialise sprite
    def __init__(self, center):
        pygame.sprite.Sprite.__init__(self)
        # specify image for explosion sprite
        self.image = explosion_imgs[0]
        # set rect for image
        self.rect = self.image.get_rect()
        self.rect.center = center
        # set initial frame for animation
        self.frame = 0
        # check last update to animation
        self.last_update = pygame.time.get_ticks()
        # set framerate delay between animation frames - sets speed for explosion
        self.frame_rate = 50
```

Python and Pygame - Game Example 1

fun game extras - create explosion sprite object - part 3

- need to add an `update` function to our class
  - *updates image of explosion for this sprite object as time progresses*
  - *i.e. as the framerate advances, switch explosion images to create animation*

```python
...
# change image as time progresses for explosion sprite
def update(self):
    # get current time
    now = pygame.time.get_ticks()
    # check if enough time has passed between animations
    if now - self.last_update > self.frame_rate:
        self.last_update = now
        # if enough time passed - add 1 to frame
        self.frame += 1
        # check if end of explosion images reached
        if self.frame == len(explosion_imgs):
            # kill if end of image reached
            self.kill()
        else:
            center = self.rect.center
            self.image = explosion_imgs[self.frame]
            # update rect for image
            self.rect = self.image.get_rect()
            self.rect.center = center
```

- need to check the current time in the game
  - *check if enough time has passed between each animation*
- if enough time has elapsed
  - *update the value for the* `last_update` *time record*
  - *advance our animation frame by an increment of 1*
- then `kill()` animation at the end of the explosion images...

Python and Pygame - Game Example 1

fun game extras - add explosions to collisions

- our sprite object for explosions has now been created
- now call this explosion whenever we record a collision between
    - *a projectile and a mob object*
    - *a mob object and player object...*
- in our game loop `update` section
    - *check for collisions we can now add an animation for the explosions*

```
...
# add more mobs for those hit and deleted by projectiles
for collision in collisions:
    # calculate points relative to size of mob object
    game_score += 40 - collision.radius
    # play explosion sound effect for collision
    explosion_effect.play()
    # add animation for explosion images if collision
    explosion = Explosion(collision.rect.center)
    # add explosion sprite to game sprites group
    game_sprites.add(explosion)
    # create a new mob object
    createMob()
...
```

- as we're checking for collisions, we can now
    - *update game score*
    - *play a sound effect for an explosions/collision*
    - *create the animation for the explosion effect*
    - ...

resources

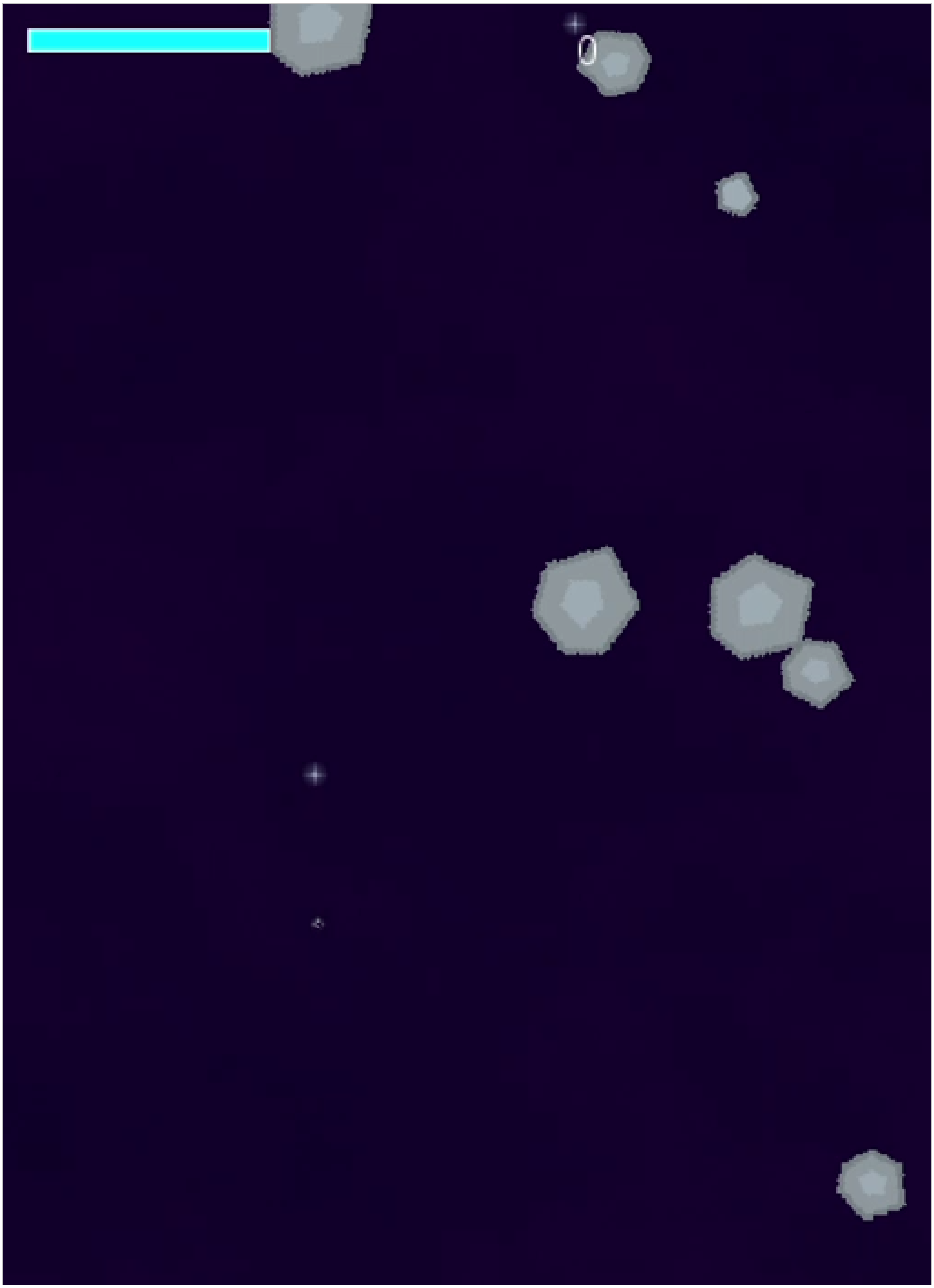
- notes = extras-part1-explosions.pdf
- code = objectexplosions.py

game example

- shooter1.2.py
- add some fun explosions
  - *create sprite object for explosion*
  - *cycle through images to create explosion animation*
  - *add explosion for each collision*

Video - Shooter 1.2 - Part 1

add some fun explosions - mob objects



0:00 / 0:20

Demos - Pygame

fun game extras

- repetitivefiring.py
- objectexplosions1.py

Demos - Pygame - Game 1 Example

- shooter1.1.py
- shooter1.2.py

Games

- Colin McRae Rally
- Diablo - Wikipedia
- Diablo III - console

References

- Bogost, I. *Persuasive Games: The Expressive Power of Videogames*. MIT Press. Cambridge, MA. 2007.
- Bogost, I, *The Rhetoric of Video Games*. in *The Ecology of Games*... Salen, E. MIT Press. Cambridge, MA. 2008.
- Bogost, I. *Unit Operations: An Approach to Videogame Criticism*. MIT Press. Cambridge, MA. 2006.
- Csikszentmihalyi, M. *Flow: The Psychology of Optimal Experience*. Harper & Row. New York. 1990.
- Murray, J. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. Free Press. New York. 1997.

References - Pygame - Game Notes

- extras-part1-firing.pdf
- extras-part1-explosions.pdf

References - Various

- The Sims - Free Will

Videos

- Colin McRae Rally - YouTube
- TED 2004 - Flow, the secret to happiness