‘Gamification’ in education

Dave Parry dparry@aut.ac.nz

Agenda

- Who games ?
- What games are
- Gamification
- Examples
- Assessing gamification

Who games ?

- Market research from https://igea.net/wp-content/uploads/2019/09/DNZ20-Final-Report-2019.pdf
- Around 70% of people aged 1-17 play games regularly
- Slightly more males than females (56/44), but males play longer
- PC’s and mobiles most popular

What are games ?

- Need to be:
  - Engaging
  - Challenging
  - Reward increasing expertise
- Fun – i.e. you **want** to play the game
- Often somewhat addictive
- Very personal - https://www.similarweb.com/apps/top/google/store-rank/nz/games/top-grossing/

So what ?

- Even very popular games aren't liked by everyone.
- Differences in:
  - Age
  - Gender
  - Interests
  - Language skills
  - Hardware
  - Free time
  - Personality
  - Visual/hearing/motor skills level
- Choice in design – Multiple games to appeal to different people, more general-purpose games to appeal to everyone.

Types of game

- Single player – i.e. against the computer
  - Single player games can be played socially and often are
- Multiplayer – against other people.
  - Can be huge environments eg world of warcraft, can be tournament eg call of duty etc.
- Genres
  - Puzzles
  - First Person Shooters/Race
  - Strategy
  - World-building
  - Reaction/skills
  - Gambling
  - Computer versions of board games etc.
- Settings
  - Console
  - Mobile – including Augmented reality
  - PC
  - Browser-based

Examples

- AR – PokemonGo https://www.youtube.com/watch?v=R04u9E5INTI&ab_channel=Pok%C3%A9monGO
- Puzzles – Candy Crush https://www.youtube.com/watch?v=Ar_dbDSvPS0&ab_channel=CandyCrushSagaOfficial
- First person shooter – Call of Duty https://www.youtube.com/watch?v=OFMUvKu-bZs&ab_channel=CallofDuty
- Strategy - Railroad empire https://www.youtube.com/watch?v=s8U8qr_Fmzk&ab_channel=GameTrailers
- World –building Sims etc. https://www.youtube.com/watch?v=ICmP6HDXLKc&ab_channel=Throneful

Hardware issues

- Operating systems – especially for Mobiles
- https://gs.statcounter.com/os-market-share/mobile/new-zealand Android/ios roughly equal but OS versions not all the same
- Phone/PC power and space
- Connectivity costs – big issue for PokemonGo
- https://www.youtube.com/watch?v=R04u9E5INTI&ab_channel=Pok%C3%A9monGO
- Browsers – normally quite easy to change

Over to you

- Games on your phone
  - How many ?
  - Which do you play ?
  - Why do you like it ?
  - Do you play with others ?

Gamification

- Active area of research https://www.nzcer.org.nz/research/games-learning
- Function of the game is to do something other than just have fun (in our case learn)
- Very wide range of approaches:
  - Drill and test – eg Duolingo, Mathletics
  - Explore a space
  - Matching/graphics
  - Story – based including VR and AR
  - Manage a system
  - Puzzles

Some examples

- Explore/puzzle/programme https://spaceplace.nasa.gov/menu/play/
- Drill and test – eg https://www.digitaldialects.com/French/Fruit_words.htm
- Management - https://www.next-kraftwerke.com/virtual-power-plant-vpp-simulation/?lang=en
- Puzzles https://kids.nationalgeographic.com/explorer-academy/games/scavenger-hunt/
- Examples of scratch games https://scratch.mit.edu/studios/25871933/

Developing games AS and FOR Learning

- Benefits:
  - Real-world software development
  - Engagement in testing and design
  - Requirements gathering and understanding
  - May be useful for other learners
- Issues
  - Porting to mobile can be complex
  - Often very simple model of learning (drill and test)
  - High-quality UX needed – very obvious when this is missing
  - Repetitive steps for development

How to do it

- Identify the need ie learning outcomes, resources and where the fun is going to come from !
- Rather than code from scratch, use a game engine (either stand-alone or libraries for other languages). These reduce the complexity of standard tasks
    - Track actions
    - Graphics
    - Recording scores
    - Computer “AI”
    - Install process
- Develop ! Teams are normally far more successful than individuals – different opinions, skills. Teams allow more demonstration of achievement of the LO’s in the CS class. Teams can develop soft skills, but also in real life they are always used.
- Use and evaluation of internet resources – support critical thinking development.

Tools

- Unity - https://unity.com/solutions/game
  - Cross platform
  - Good at AR/web/Mobile
  - Relatively easy to use
  - Free student version
  - Graphics can be quite simple
  - https://armedgorilla.itch.io/math-ball
  - https://jakubjahic.itch.io/learn-object-oriented-programming
- Unreal - https://www.unrealengine.com/en-US/features
  - Very high-level professional tool
  - Free if not commercially used
  - Relatively complex, but good set of tutorials

Other languages

- Python has a library PyGame which supports game development https://wiki.python.org/moin/GameProgramming
- Examples at: http://www.grantjenks.com/docs/freegames/
- Can run on very low power hardware
- Javascript/HTML5 – not really focussed on Games but very widely used in browsers. Most ported “classic” games use JS. https://blog.logrocket.com/top-6-javascript-and-html5-game-engines/

Social aspects

- Many games allow you to post to social media, Twitch https://www.twitch.tv/ is the ultimate here where you can stream your game in progress.
- Competition between players can be good, but can also have issues if uncontrolled/moderated
- You can see social media as a game – likes and connections as prizes ;-)
- Undesirable activity can be hidden in games so be careful around in-game communication

Some evaluation ideas

- Does this game actually contribute to a learning outcome ?
- Can you cheat ? Cheating may not be bad – it might actually be research.
- Are there frustration bottlenecks ?
- Is the interface usable including for different groups ?
- What sort of time do you want students to spend on it ?
- How does it encourage metacognition – can you rewind/ record to see how you made decisions ?
- Hardware compatible ?
- Can you record usage ?

Take home messages

- MCQ’s are very popular as are matching games. These are OK for basic facts but are often very limited
- More complex games can become very difficult to control – UX is often a major stumbling block. Consistency of interface is very rare !
- You already know this – games to tech particular concepts need to fit into the curriculum and add value
- Getting students to develop serious/educational games is an interesting strategy and may be a good way to keep students from seeing CS as “loner in a basement”.